

*Clefts, Relatives, and Language Dynamics:
The Case of Japanese*



Tohru Seraku

St. Catherine's College
University of Oxford

A thesis submitted to the University of Oxford
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
IN COMPARATIVE PHILOLOGY AND GENERAL LINGUISTICS

Trinity Term 2013

Clefts, Relatives, and Language Dynamics: The Case of Japanese

Tohru Seraku
St. Catherine's College

A thesis submitted to the University of Oxford in Trinity Term 2013
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY IN COMPARATIVE PHILOLOGY AND GENERAL LINGUISTICS

Abstract

The goal of this thesis is to develop a grammar model of Japanese within the framework of Dynamic Syntax (Cann et al. 2005, Kempson et al. 2001), with special reference to constructions that involve the nominaliser *no*: clefts and certain kinds of relatives. The more general theoretical position which it aims to defend is that an account of these constructions in terms of 'language dynamics' is preferable to other 'static' approaches currently available. What is here meant by 'language dynamics,' in a nutshell, is the time-linear processing of a string and attendant growth of an interpretation.

First, I shall motivate, and articulate, an integrated account of the two types of *no*-nominalisation. These two classes are uniformly modelled as an outcome of incremental semantic-tree growth. The analysis is corroborated by naturally-occurring data extracted from the Corpus of Spontaneous Japanese (CSJ). Moreover, novel data with regard to coordination are accounted for without losing uniformity.

Second, the composite entry of *no* and the topic marker *wa* handles the two types of clefts uniformly. This account fits well with the CSJ findings. New data concerning case-marking of foci are explained in terms of whether an unfixed relation in a semantic tree is resolvable in incremental processing. The account also solves the island-puzzle without abandoning uniformity. As a further confirmation, the analysis is extendable to stripping/slucing, making some novel predictions on case-marking patterns.

Third, the entry of *no* characterises free relatives and change relatives in a unitary manner. Furthermore, the composite entry of *no* and a case particle predicts a vast range of properties of head-internal relatives, including new data (e.g., negation in the relative clause, locality restriction on the Relevancy Condition).

In sum, the thesis presents a realistic, integrated, and empirically preferable model of Japanese. Some consequences stand out. The various new data reported are beneficial theory-neutrally. Formal aspects of Dynamic Syntax are advanced. The insights brought by a language dynamics account challenge the standard, static conception of grammar.

My family, relatives, and their love

Contents

<i>Prologue</i>	vii
<i>Notes to the Reader</i>	x
CHAPTER 1 SETTING THE SCENE	
1. Ambitions	1
2. Guidelines	2
2.1 Leading Principles	2
2.2 Why Japanese?	3
2.3 Why Clefts and Relatives?	4
2.4 Why Language Dynamics?	7
3. Previews	9
CHAPTER 2 EMPIRICAL SCAFFOLDING	
1. Structural Properties of Japanese	18
2. Methodological Preliminaries	22
2.1 Corpus of Spontaneous Japanese	22
2.2 Search Procedures	23
3. Overall Results	26
4. <i>No</i>-Nominalisation	29
4.1 Introduction	29
4.2 Definiteness	31
4.3 Grammatical Function of the <i>No</i> -Part	32
4.4 Grammatical Function of the Gap	37
4.5 Long-Distance Dependencies and Islands	40
4.6 Summary	41
5. Cleft Constructions	42
5.1 Introduction	42
5.2 Previous Corpus Work	43
5.3 Particle Dropping	47
5.4 Category of Foci	50
5.5 Grammatical Function of Foci	52

5.6 Information Status of Foci	55
5.7 Long-Distance Dependencies and Islands	56
5.8 Summary	58
6. Theoretical Significances	59
 CHAPTER 3 THEORETICAL SCAFFOLDING	
1. Dynamic Syntax (DS)	61
2. The Declarative Part	63
2.1 Trees and Tree Descriptions	63
2.2 Epsilon Calculus	67
2.3 Summary	71
3. The Procedural Part	72
3.1 Actions for Tree Update	72
3.2 Verbal Encoding	76
3.3 Structural Underspecification	78
3.4 Summary	83
4. Sample Tree Transitions	84
4.1 Simple Sentence	84
4.2 Clause Embedding	88
4.3 Summary	91
5. Paired Trees in LINK	92
5.1 LINK Relations	92
5.2 Head-External Relatives	94
5.2.1 The Analysis	94
5.2.2 Case Particles Revisited	98
5.2.3 Tree-Node Identification	101
5.2.4 Novelties	104
5.3 Topic Constructions	104
5.4 Summary	109
6. Dynamic Construal of Context	110
7. Coda: Cross-Theory Comparisons	111
 CHAPTER 4 THE DYNAMICS OF <i>No</i>-NOMINALISATION	
1. Overview	115
2. Motivating a Unitary Analysis	118
2.1 Introduction	118
2.2 Methodological Considerations	119
2.3 Functional Considerations	120
2.4 Diachronic Considerations	122
2.5 Cross-Linguistic Considerations	123
2.6 Cross-Dialectal Considerations	126
2.7 Summary	128

3. A Dynamic Account	129
3.1 Introduction	129
3.2 The Proposal	129
3.3 The Analysis	131
3.3.1 Participant Nominalisation	131
3.3.2 Situation Nominalisation	136
3.3.3 Ambiguity Cases	139
3.4 Consequences	140
3.4.1 <i>No</i> as a Dependent Item	140
3.4.2 Expressive Meaning	141
3.4.3 Nature of Denotation	144
3.4.4 Definiteness	147
3.4.5 Grammatical Function of the <i>No</i> -Part	148
3.4.6 Grammatical Function of the Gap	150
3.4.7 Long-Distance Dependencies and Islands	152
3.4.8 Loose Ends	154
3.5 Summary	158
4. Nominalisation and Coordination	158
4.1 Introduction	158
4.2 The Asymmetry	159
4.3 Participant Nominalisation	162
4.3.1 Clausal Coordination	162
4.3.2 The Analysis	164
4.3.3 Further Predictions	167
4.4 Situation Nominalisation	169
4.4.1 Representing Tense	169
4.4.2 The Analysis	170
4.4.3 Residual Issues	175
4.5 Summary	176
5. Conclusion	176
 CHAPTER 5 THE DYNAMICS OF CLEFT CONSTRUCTIONS	
1. Overview	178
2. Previous Studies	181
2.1 Introduction	181
2.2 Hoji (1990)	182
2.3 Kizu (2005)	184
2.4 Hiraiwa & Ishihara (2002, 2012)	187
2.5 Summary	188
3. A Dynamic Account	189
3.1 Introduction	189
3.2 The Proposal	189

3.3 The Analysis	193
3.3.1 Clefts _{+C}	193
3.3.2 Clefts _{-C}	199
3.3.3 A Realistic Model of Clefts	201
3.4 Consequences	202
3.4.1 Category of Foci	202
3.4.2 Grammatical Function of Foci	204
3.4.3 Information Status of Foci	205
3.4.4 Topic and Focus	206
3.5 Summary	207
4. Multiple Foci	208
4.1 Introduction	208
4.2 A Formal Basis for Explanation	209
4.3 The Analysis	211
4.4 Further Predictions	214
4.5 Summary	217
5. Long-Distance Dependencies and Islands	218
5.1 Introduction	218
5.2 Long-Distance Clefts	218
5.3 Island-Involving Clefts	225
5.3.1 Clefts _{-C}	227
5.3.2 Clefts _{+C}	230
5.3.3 The Copula Revisited	232
5.4 Summary	234
6. Implications for Ellipsis	234
6.1 Introduction	234
6.2 The Analysis	235
6.2.1 Stripping	235
6.2.2 Sluicing	239
6.3 Multiple Foci	242
6.3.1 Multiple Stripping	242
6.3.2 Multiple Sluicing	247
6.4 Long-Distance Dependencies and Islands	251
6.4.1 Long-Distance Stripping	252
6.4.2 Long-Distance Sluicing	254
6.4.3 Island-Involving Stripping	256
6.4.4 Island-Involving Sluicing	259
6.5 Summary	261
7. Conclusion	261

CHAPTER 6 THE DYNAMICS OF RELATIVE CONSTRUCTIONS

1. Overview	264
2. Properties of HIRs	266
2.1 Introduction	266
2.2 Indeterminacies	267
2.3 Quantification	269
2.4 Islands	274
2.5 Ambivalence	275
2.5.1 Non-Nominality of HIRs	275
2.5.2 Nominality of HIRs	279
2.5.3 Consequences	282
2.6 Summary	283
3. Previous Studies on HIRs	283
3.1 Introduction	283
3.2 A Brief History	284
3.3 Kim (2007, 2008a, 2008b, 2009)	286
3.4 Grosu (2010) and Grosu & Landman (2012)	295
3.5 Summary	300
4. A Dynamic Account of HIRs	301
4.1 Introduction	301
4.2 The Proposal	302
4.3 The Analysis	305
4.4 Consequences	308
4.4.1 Indeterminacies	308
4.4.2 Quantification	314
4.4.3 Islands	318
4.4.4 Ambivalence	319
4.5 Summary	325
5. Change Relatives	326
5.1 Introduction	326
5.2 Nominality of CRs	326
5.3 Inertness of the Relevancy Condition in CRs	332
5.4 The Analysis	334
5.5 Summary	338
6. Conclusion	338

CHAPTER 7 SYNOPSIS AND BEYOND

1. Overall Summaries	340
2. Future Prospects	346
2.1 Construction in Linguistic Theories	346
2.2 Towards Dynamic Construction Grammar	351

APPENDIX THE DYNAMIC SYNTAX FORMALISM	
1. Introduction	358
2. Supplements to Chapter 3	358
3. A Mini Grammar of Japanese	363
3.1 An Axiom	363
3.2 General Actions	363
3.3 Lexical Actions	366
3.4 Pragmatic Actions	369
REFERENCES	370

Prologue

Mais, la semaine dernière, j'ai reçu avis d'une communication téléphonique surprise par un de mes correspondants et dont l'importance ne vous échappera pas. De son appartement, sis à Paris, une dame communiquait avec un monsieur de passage dans un hôtel d'une grande ville des environs. Le nom de la ville, le nom du monsieur, le nom de la dame, mystères. Le monsieur et la dame causaient espagnol, mais en utilisant cet argot que nous appelons le javanais, et, même, en supprimant beaucoup de syllabes. Quoi qu'il en soit des difficultés accumulées par eux, si toute leur conversation ne fut pas notée, on réussit cependant à saisir l'essentiel des choses très graves qu'ils se disaient et qu'ils mettaient tant de soin à cacher!

Maurice Leblanc
'Thérèse et Germaine'
Les huit coups de l'horloge, 1923

If my memory serves me correctly, the first time I became interested in language was when I was reading (the Japanese translation of) the above French excerpt in a living room of my piano teacher, Aiko Yamaguchi. Ever since, little by little, this experience has been shaped into my research interests and love of language.

My supervisors at Oxford, Jieun Kiaer and David Cram, have guided me through this exciting and challenging programme. Jieun pointed out a number of fundamental issues relating to my research. In fact, it is partly due to Jieun's suggestions that I chose Japanese as a target language and decided to use a corpus in my thesis. David always stimulated my work in his gentle manner, giving me lots of valuable advice from the perspective of pragmatics, as well as advice on non-academic matters; it was David who taught me about, for instance, dress codes required for Oxford's social events.

The present thesis has immensely benefitted from numerous comments made at my viva from my examiners, Mary Dalrymple and Lutz Marten. I greatly appreciate

their constructive feedback, which shaped my thesis in terms of both content and style. I was fortunate to have been able to learn Lexical Functional Grammar from Mary at Oxford and to discuss issues in Dynamic Syntax with Lutz occasionally in London.

My very special thanks should go to Ruth Kempson, an ideal researcher for me. Throughout my D.Phil. period, my work has evolved ‘incrementally’ with countless discussions with Ruth. She reminds me of how to be excited about language, cognition, and all facets of our lives, as well as how to be kind to others. This way of saying it may sound banal, but this is how it was.

Along with the road that took me to Oxford were my wonderful former teachers in London and Tokyo: Robyn Carston, Yukio Otsu, Yuji Nishiyama, Masafumi Torii, Tetsuichi Nishimura, and Misato Tokunaga. My interests in language, communication, and cognitive science were founded in the guidance and insights which I have gained from these teachers.

This thesis would never have come into being without the support and advice that I received from the following people, to whom I am really grateful: Márta Abrusán, Nick Allott, Masatake Arimoto, Ash Asudeh, Emmon Bach, Ylva Berglund, Miriam Bouzouita, Ronnie Cann, Stergios Chatzikyriakidis, France Dhorne, Kate Dobson, Arash Eshghi, Bjarke Frellesvig, Mayumi Fujita, Teruhiko Fukaya, Geoffrey Gosby, Eleni Gregoromichelaki, Alison Hall, Sayaka Hasegawa, Tomonori Hatano, Ken Hiraiwa, Stephen Wright Horn, Naoko Hosokawa, Chris Howes, Mika Igarashi, Satoshi Imamura, Noriko Imanishi, Toru Ishii, Miwa Isobe, Barış Kahraman, Anna Kibort, Satoshi Kinsui, Hisatsugu Kitahara, Kazuma Kitamura, Hiroaki Kitano, Yoolim Kim, Mika Kizu, Nathan Klinedinst, Naomi Kodama, Masayuki Komachi, Masatoshi Konno, Akiko Kurosawa, Aimi Kuya, Chungmin Lee, Laurence Mann, Tomoko Matsui, Wilfried Meyer-Viol, Koji Mineshima, Kaori Miura, Edson Miyamoto, Tomoko Monou, Iris Mulders, Louise Mycock, Atsushi Nagai, Seiichi Nakada, Hiroki Narita, Ad Neeleman, Ken’ya Nishikawa, Mayumi Nishikawa, Masashi Nomura, Toshio Ohori, Akira Ohtani, Etsuko Oishi, Sven Osterkamp, Koichi Otaki, Takahiro Otsu, Sandra Paoli, Jong Park, Rob Peel, Philomen Probert, Matthew Purver, Daniel Rothschild, Kerri Russell, Shigeru Sakahara, Kumiko Sakoda, Tetsuya Sano, Ryoko Sasamoto, Yo Sato, Jun Sawada, Osamu Sawada, Makoto Sekita, Peter Sells, Ethan Sherr-Ziarko, Ryo Shibagaki, Masayoshi Shibatani, John Charles Smith, Dan Sperber, Maki Sudo, Yasutada Sudo, Yoko Sugioka, Koji Sugisaki, Peter Sutton, Takeru Suzuki, Axel Svahn, Tomokazu Takehisa, Michiko Takeuchi, Kensuke Takita, Masami Tanabe, Hiroaki Tanaka, Ye Tian, Dan Trott, Asako Uchibori, Hiro Uchida, Satoshi Umeda, Reiko Vermeulen, Akira Watanabe, John Whitman, Deirdre Wilson, Martin Wynne, Katsuhiko Yabushita, Aiko Yamanaka, Masaaki Yamanashi, Carryn Yong, Etsuko Yoshida, Tomomi Yoshihara, Zixi You, Ichiro Yuhara, and Beata Zacharska.

I also hugely appreciate the generous financial support I have received, over the years, to carry out my D.Phil. research: the Oxford-Kobe Scholarship (St. Catherine’s College), the Clarendon Fund Scholarship (the University of Oxford), the Sasakawa

Fund Scholarship (the University of Oxford), and the Light Senior Scholarship (St. Catherine's College). My conference presentations, all of which are relevant to the present thesis, are supported by the Siddiki Travel Bursary and various travel grants offered by the Faculty of Linguistics, Philology, and Phonetics at Oxford. At the very moment of writing the thesis, the writing-up grant, also provided by this Faculty, was quite helpful.

It is more than four years ago that I came to Oxford to study, listening to *Love and Laughter* (Hiromi Uehara, 2005, included in *Spiral*), which became a sort of theme tune to Oxford for me. My beloved friends made my Oxford life full of wonderful moments – *Staircase 2*, *high tables*, *college balls*, *the water boy*, *Rewley Road*, *Port Meadow*, and so on and so on. Londoner friends, and in fact, friends around the world, provided me with the occasional moments of jocular distraction I needed to make it through to the end! And, of course, every time I was back in Japan, I was delighted to see my dear friends there, too. I would like to say a big thank-you to these friendships.

Last but not least, I cannot express my gratitude enough to my parents, Takeshi Seraku and Hiroko Seraku, who believe in me and relieve me with constant care and love. Writing this prologue has caused me to consider just how fortunate I am: I have a great family. Sincerely, I dedicate this thesis to them.

Notes to the Reader

Examples

Three types of example are used in this thesis: (i) examples from previous works, (ii) examples collected from the Corpus of Spontaneous Japanese (CSJ), and (iii) examples invented by myself. When examples are cited from previous studies, on occasion, the formatting has had to be adapted. This is not always explicitly mentioned in the text. When examples from the CSJ are presented, they are translated into English by myself, often with minor changes (e.g., omission of fillers); these modifications are not stated in the text. My example sentences are based on the fantasy worlds depicted by Studio Ghibli. Characters in my beloved Ghibli films appear in each chapter and the appendix.

- Chapter 1. 耳をすませば (*Whisper of the Heart*, 1995)
- Chapter 2. 紅の豚 (*Porco Rosso*, 1992)
- Chapter 3. 天空の城ラピュタ (*Laputa: Castle in the Sky*, 1986)
- Chapter 4. とんりのととろ (*My Neighbor Totoro*, 1988)
- Chapter 5. 千と千尋の神隠し (*Spirited Away*, 2001)
- Chapter 6. 魔女の宅急便 (*Kiki's Delivery Service*, 1989)
- Chapter 7. 風立ちぬ (*The Wind Rises*, 2013)
- Appendix. もののけ姫 (*Princess Mononoke*, 1997)

Each example consists of three lines. The first line represents a natural language string, though some symbols (e.g., brackets) are used. The second line provides glosses. A list of abbreviations used for glosses is presented on the next page. The third line provides a free translation. Translations are sometimes preceded by the symbols 'Lit.' (= literal translation) or 'Int.' (= intended interpretation).

A List of Abbreviations

Abbreviation	Description
2SG	second person singular
ACC	accusative-case
CAUSE	causative
CL	classifier
COMP	complementiser
COP	copula
DAT	dative-case
DECL	declarative
DEF	definite
GEN	genitive-case
GENP	general preposition
IMPRF	imperfective
KES	<i>kes</i> -nominaliser in Korean
KOTO	<i>koto</i> -nominaliser in Japanese
LINK	clause-linking
NEG	negation
NMNZ	nominaliser
NO	particle <i>no</i> in Japanese
NOM	nominative-case
PASS	passive
PAST	past tense
POLITE	politeness
PRE	pre-nominal
PRES	present tense
PRF	perfect
PROG	progressive
Q	question marker
QUOT	quotation marker
REL	relative-clause marker
SE	particle <i>se</i> in Haitian
SFP	sentence final particle
TOIU	compound particle <i>toiu</i> in Japanese
TOIUNO	compound particle <i>toiuno</i> in Japanese
TOIUNOWA	compound particle <i>toiunowa</i> in Japanese
TOP	topic

A List of Figures

Description	Page
1.1. Form-function correspondences	12
1.2. Form-function correspondences	16
4.1. Form-function correspondences	119
6.1. Form-function correspondences	266
6.2. Form-function correspondences	339
7.1. Form-function correspondences	345
7.2. Multiple inheritance hierarchy	348
7.3. Template hierarchy	350

A List of Tables

Description	Page
2.1. Overall result of the corpus search	26
2.2. <i>No</i> -nominalisation in the CSJ	30
2.3. Grammatical function of the <i>no</i> -part in participant nominalisation	33, 148
2.4. Grammatical function of the <i>no</i> -part in situation nominalisation	35, 148
2.5. Grammatical function of the gap in participant nominalisation	37, 150
2.6. Type of the focus in clefts	45
2.7. Particles attached to the focus	45
2.8. Particle dropping in clefts in the CSJ	47
2.9. Category of a focus in clefts in the CSJ	50, 202
2.10. Type of <i>wh</i> -word in clefts in the CSJ	51
2.11. Grammatical function of a focus in clefts in the CSJ	52, 204
2.12. Procedures to detect the information status of a focus in the CSJ	56
2.13. Long-distance and island-involving cases of clefts in the CSJ	58
6.1. Classification of <i>no</i> -nominalising constructions (preliminary)	337
6.2. Classification of <i>no</i> -nominalising constructions (final version)	337
7.1. Classification of <i>no</i> -nominalising constructions	345
A.1. Repertoire of logical types	360
A.2. Repertoire of LOFT operators	361
A.3. Repertoire of atomic actions	362

Chapter 1

Setting the Scene

1 Ambitions

What does it mean to know a language? In contemporary linguistics, especially what Culicover & Jackendoff (2005) dub ‘Mainstream Generative Grammar,’ a language has often been viewed as a static system of principles and rules, which are abstracted from the way they are put to use in context. A fundamental distinction in this regard is the one between ‘competence’ and ‘performance,’ or put differently ‘knowledge of language’ and ‘use of language’ (Chomsky 1965, 1986). The present thesis also assumes this distinction, agreeing that we possess tacit knowledge of language as an internal system in the mind. It departs from the standard view, however, in viewing this system in the light of how we communicate with each other in natural settings. As we pursue this research direction, we shall be returning to our naïve conception of language: ‘language is a tool for communication.’

A property of language use is that we produce and comprehend a string of words¹ left-to-right in real time. To focus on the comprehension side, we accumulate a representation of what a speaker/writer intends to convey as we process an incoming linguistic stimulus in a piecemeal manner. A challenge is how to capture in formal terms this dynamic nature of incremental processing. This task took on an interesting

¹ What is meant by ‘word’ here (and throughout the thesis) includes affixes.

twist in the 1990's, when a radical claim was put forward by Ruth Kempson and colleagues that modelling language dynamics serves as a basis for explaining linguistic phenomena that have been addressed in the study of the syntax and semantics of natural languages (Kempson & Gabbay 1998, Kempson et al. 1999). This insight has been shaped and crystallised into a formalism known as 'Dynamic Syntax' (Cann et al. 2005, Kempson et al. 2001, Kempson et al. 2011).

This thesis aims at contributing to this provocative enterprise by defending the theme: 'language dynamics is a significant concept in linguistic theorising.' In order to make a case for this fundamental claim, I shall demonstrate that language dynamics is vital for developing internally-consistent, satisfactory² grammatical descriptions (and, where possible, explanations) of the two constructions involving the nominaliser *no* in Japanese: clefts and certain types of relatives (i.e., change relatives, free relatives, head-internal relatives). These constructions will be illustrated shortly below. Through these case studies, I hope to show that our tacit knowledge of language is best illuminated by formalising how we build up a context-dependent interpretation in a progressive manner as we process a string of words in the order it is produced.

2 Guidelines

2.1 Leading Principles

Let us first clarify three principal assumptions regarding the architectural design of a grammar to be proposed: (i) generative theory of competence, (ii) competence reflecting language dynamics, and (iii) syntax as a set of procedures for constructing semantic representation.

² An account to be developed is 'satisfactory' in that it is 'explanatory,' 'realistic,' 'integrated,' and 'empirically favourable' (see Sect. 2.4).

This thesis models a mental grammar of Japanese as used in interpreting a string that instantiates so-called cleft constructions and relative constructions. This grammatical model pairs strings and structures ‘explicitly’ in the sense of Chomsky (1965), hence a version of generative grammar. In short, the grammatical model to be presented is a **generative** theory of **competence**, not performance.

Unlike other linguistic frameworks, this competence model directly reflects **language dynamics**: incremental processing and attendant progressive update of an interpretation in context. In this construal, competence is transparently dependent upon performance, especially how we process a string of words left-to-right under a specific context. Thus, competence intimately interacts with pragmatic inference. A model of competence, however, does **not** constitute a pragmatics system by itself; rather, it provides a possible range of interpretations out of which the pragmatics system chooses at each step of incremental processing.

In this conception of competence, ‘syntax’ is regarded as no more than a set of instructions which the parser employs to construct a semantic structure. There is **no** independent level of syntactic representation, and the string is **directly** mapped onto semantic representation. Structural puzzles of natural languages that have been tackled in the syntax literature are explained as an outcome of how a structured interpretation is progressively built up as a string is parsed left-to-right online.³

2.2 Why Japanese?

I shall articulate a grammar of Japanese to defend the theme: ‘language dynamics is a significant concept in linguistic theorising.’ But why Japanese? This is because the language poses several challenges to the modelling of incremental processing (Kiaer in press). First, there is an issue of linearity. In Japanese, a verb is processed at the end of a string (so-called ‘verb-final’), but it is the verb *per se* that builds up a structure of the

³ Despite this new conception of grammar, standard terminology in the syntax literature such as ‘islands’ and ‘long-distance’ will be used purely for the sake of illustration.

string. This is complicated by the fact that as long as a verb comes at the end of a clause, arguments and adjuncts may be freely scrambled, possibly across a clausal boundary. Second, there is an issue of the grammar-pragmatics interface. Japanese is a fully pro-drop language, and all arguments may be dropped if they are retrievable contextually. Even if arguments are expressed overtly, their quantificational force and plurality must be contextually inferred because there is no such morphological marking. (For these structural properties, see Sect. 1, Chap. 2.) Therefore, Japanese is suitable for testing the adequacy of a dynamic grammar to be proposed.

2.3 Why Clefts and Relatives?

Why, then, is this thesis particularly concerned with cleft constructions and certain types of relative clause constructions? There are two interconnected reasons.

First, both clefts and these kinds of relatives vividly illustrate the dynamic flow of language understanding in context. These constructions involve the particle *no*. In terms of linear parsing, the pre-*no* clause is seen to establish a context for the post-*no* part to be subsequently processed. In Chap. 4, the particle *no* is viewed as a nominaliser that takes a preceding clause and outputs a nominal which reflects the proposition of the clause. To take ‘participant nominalisation’ (see Sect. 3 for details) as an example, the pre-*no* clause *toshokan-kara karita* denotes an (open) proposition, and this is turned into a nominal expression that denotes an entity reflecting the propositional content, namely an entity which Shizuku borrowed from a library.

(1.1) Participant Nominalisation

Shizuku-wa [toshokan-kara karita no]-o yonda.

S-TOP [library-from borrowed NO]-ACC read

‘Shizuku read what she borrowed from a library.’

In Chap. 5, the composite unit of the nominaliser *no* and the topic particle *wa* takes a preceding clause and outputs a propositional schema to be enriched with the content of

a focus expression. This characterises cleft constructions. In (1.2), the pre-*no-wa* clause *Shizuku-ga karita* denotes an (open) proposition, and this proposition is mapped onto a new proposition into which the content of the focus expression *hon* (= ‘book’) will be subsequently incorporated.

(1.2) Clefts

[Shizuku-ga karita no]-wa hon da.
 [S-NOM borrowed NO]-TOP book COP
 ‘It is a book that Shizuku borrowed.’

In Chap. 6, the composite unit of the nominaliser *no* and a case particle takes a preceding clause and outputs a ‘rich’ propositional schema to be fleshed out by the embedding clause. This parsing-strategy leads to an account of head-internal relatives. In (1.3), the head *udon* is located inside the relative clause. The sequence *no-o* maps the propositional content of the relative clause onto a new propositional structure. This structure is ‘rich’ in that it is partially articulated when it is induced; in particular, there has already been a node decorated with the content of the head. This emergent structure will be subsequently fleshed out by the parse of the embedding verb *tabe-* (= ‘eat’).

(1.3) Head-Internal Relatives (HIRs)

Shizuku-wa [Shiro-ga udon-o tsukuttekureta no]-o tabe-ta.
 S-TOP [S-NOM udon-ACC served NO]-ACC eat-PAST
 ‘Shiro served udon to Shizuku, and she ate it.’

It is also shown in Chap. 6 that the nominaliser *no* models the time-linear flow of interpreting free relatives and change relatives. ‘Free relatives’ is another name for ‘participant nominalisation’ such as (1.1). Change relatives are a construction which is quite similar to head-internal relatives in that the head is found within the relative clause and that the relative clause ends with the particle *no*. Unlike head-internal relatives, change relatives denote the state of the individual as a result of the change described in

the relative clause. (See Sect. 5, Chap. 6 for structural/interpretive differences between head-internal relatives and change relatives.) In (1.4), what is denoted is the individual that underwent a change from a doll into a gentleman. In this construction, the pre-*no* clause represents a (closed) proposition, and it is turned into a nominal expression that denotes an individual reflecting the propositional content, namely an entity that became a gentleman from a doll.

(1.4) Change Relatives (CRs)

[Ningyou-ga shinshi-ni natta no]-ga Shizuku-to

[doll-NOM gentleman-COP become NO]-NOM S-with

rapisurazuri-no-komyaku-e mukatta.

Lapis.Lazuli-GEN-vein-to went

‘A gentleman who is the result of changing from a doll went to the vein of Lapis Lazuli with Shizuku.’

In this way, these *no*-involving constructions nicely manifest the time-linear process of understanding a string of words left-to-right; what has been parsed before the particle *no* represents a propositional structure relative to which the rest of the string is parsed. Thus, these constructions constitute good test cases of articulating a dynamic, linear-parsing-sensitive grammar model.

Second, despite the parallelism concerning the flow of language understanding, these constructions raise distinct syntactic and semantic problems. To mention a few, Japanese clefts may license more than one focus item, but at the same time, there are tight constraints on these multiple foci, which, I shall claim, is a reflection of language dynamics (see Sect. 4, Chap. 5). Further, to take head-internal relatives as an example of relative constructions, the head appears to be located inside of the relative clause, but this internal head is associated with the embedding verb. This association is vastly context-sensitive (e.g., the so-called ‘split head’) and still subject to tight interpretive constraints (i.e., the so-called ‘Relevancy Condition’) (see Sect. 2, Chap. 6).

These two reasons are interrelated because the heterogeneous structural and interpretive properties of these constructions, it will be argued, are accounted for as an outcome of modelling the dynamic flow of Japanese incremental processing.

2.4 Why Language Dynamics?

Our central theme is that ‘language dynamics,’ or the incremental nature of language understanding in context, is fruitful for modelling knowledge of language. At this initial juncture, it would be useful to delineate advantages of taking a dynamic approach.

First, the heart of this approach is the way we understand a string in real time as it is presented to us bit by bit. Linguistic phenomena are explained as an outcome of modelling gradual update of an interpretation in context. Thus, an analysis offered in the dynamic approach is **explanatory** in that it is deep-seated in capturing a foundational property of language use (Kiaer in press).

Second, the dynamic approach enables us to articulate a **realistic** grammar, ‘realistic’ in that the competence model is transparently embedded within a model of human linguistic performance (Sag & Wasow 2011). One might object that this is not a counterargument against a static model of grammar since the aim of the static approach is to develop a grammar which itself is independent of language use, in the first place. Still, the dynamic approach is at least methodologically preferred to the static approach in that the former is more heavily constrained by performance data. That is, when we propose a grammatical model in the dynamic approach, it is not satisfactory if it just accounts for syntactic puzzles; it must account for them so as to be consistent with our actual use of language in context, hence a ‘realistic’ grammar.⁴ For instance, Japanese being strictly verb-final, we process a verb at the very end of a string, but it is the verb that gives us the basic structure of the string. One might presume that the parser delays

⁴ Another equally important aspect of a ‘realistic’ grammar is that it must be induced from the sequences the language-learning child is exposed to (Clark & Lappin 2011). But the issue of learnability is beyond the scope of the present thesis (see Eshghi et al. 2013).

structure building until it processes a verb; but there is a growing body of evidence that suggests the opposite conclusion: the parser starts to build a structure without waiting for a verb to come (Kamide 2006, Kiaer in press). A realistic grammar of Japanese must reconcile this putatively ambivalent state of affairs while solving a number of structural and interpretive problems posed by the language. This direct modelling of time-linear process by a core grammar is absent from MGG (Mainstream Generative Grammar), where ‘a generative system involves no temporal dimension’ (Chomsky 2007: 6). Within MGG, a parsing device may be directly falsified by the incremental processing data mentioned above, but not a core grammar itself.⁵

Third, the dynamic approach **integrates** seemingly distinct constructions under the same umbrella in terms of how a string is processed left-to-right online. This is illustrated in Chaps. 4, 5, and 6, where apparently distinct sets of data are shown to be amenable to a uniform treatment. In Chap. 4, the two types of *no*-nominalisation phenomena are reduced to a parser’s choice of what type-*e* term it copies in processing the particle *no*. In Chap. 5, the two types of clefts boil down to different modes of resolving the unfixed node of a focus. In Chap. 6, change relatives are handled by the same mechanism used in *no*-nominalisation; further, if this mechanism is enriched with the insight of the analysis of clefts, it predicts a vast array of structural and interpretive aspects of head-internal relatives.

Finally, the dynamic approach is **empirically** favourable in that it accounts for phenomena that pose a challenge to the static approach. To give a concrete example in Chap. 5, previous studies have assigned radically different syntactic structures to cleft strings depending on whether a focus is case-marked. One set of data that motivates this bifurcated view is that multiple foci are possible only if the foci are case-marked. This thesis, however, observes that in multiple foci, the final focus (and only the final focus)

⁵ Previous studies have investigated the nature of the parser that is governed by core-grammar principles (see, e.g., Pritchett 1992). In these works, a core grammar itself is silent to the time-linear process in incremental parsing. An exception is Phillips (2003), who develops (but does not fully formalise) an incremental grammar within MGG (see Sect. 7, Chap. 3).

may be case-less. These data are recalcitrant for previous studies since partially case-marked foci instantiate the two types of clefts at the same time, as it were. It will be argued that the partially case-marked foci reflect the way the parser resolves the unfixed node of a focus during the course of updating semantic representation.

The dynamic approach is thus justifiable from methodological, theoretical, and empirical points of view.⁶ These justifications will be substantiated through the analysis of cleft constructions and relative constructions in Japanese, as presented in snapshot form in the next section.

3 Previews

The present thesis consists of two main parts: Chaps. 2-3 set out empirical and formal backgrounds, against which Chaps. 4-6 propose an in-depth account of various *no*-involving constructions. Chap. 7 draws overarching conclusions and posits prospects for future study. The appendix formally defines the theoretical underpinnings of the thesis, Dynamic Syntax.

Chap. 2 lays out empirical foundations. First, basic structural properties of the Japanese language are illustrated. These include argument dropping, scrambling, head-finality, subordination, and so on. Second, a large part of this chapter is devoted to the presentation and discussion of naturally-occurring data collected from the Corpus of Spontaneous Japanese (2nd edition, The National Institute for Japanese Language and Linguistics, 2008). The main target of this corpus study is clefts; still, since clefts involve the particle *no*, the data of other *no*-involving constructions are also discussed. These corpus data will be formally accounted for in later chapters (see Chap. 4 on *no*-nominalisation and Chap. 5 on clefts).

⁶ As will be suggested in Chap. 5 (Sect. 4.3), another benefit of the dynamic approach is that it may predict not only the ungrammaticality of a string but also the **timing** where the string becomes ungrammatical during the course of incremental left-to-right processing.

Chap. 3 introduces a theory of language dynamics, as modelled in Dynamic Syntax. Though the thesis largely follows the formalism provided in Cann et al. (2005), Kempson et al. (2001), and Kempson et al. (2011), it also advances some formal aspects of the framework based on Seraku (2013b, in press a). Most importantly, it will be claimed that the initial node in structure building is **underspecified** for a node address. That is, the initial node may, but does not have to, turn out to be a root node of the whole semantic tree. Based on this new assumption, I shall refine the recent account of head-external relatives (Kempson & Kurosawa 2009). The entry of case particles is enriched with the bottom-up structure building process, and the tree-node identification process provides formal explication of how globally unfixed relations may be resolved.

Against these empirical and formal backgrounds, the subsequent three chapters confront several constructions in Japanese. Chap. 4 concerns the particle *no*, a key item that appears in clefts (see Chap. 5) and relatives (see Chap. 6). The particle *no* presents a number of intriguing issues and merits its own chapter. The particle *no* enables two types of nominalisation (e.g., Kitagawa & Ross 1982): ‘participant nominalisation’ as in (1.5) and ‘situation nominalisation’ as in (1.6) and (1.7).

(1.5) Participant Nominalisation

Shizuku-wa [toshokan-kara karita no]-o yonda.
 S-TOP [library-from borrowed NO]-ACC read
 ‘Shizuku read what she borrowed from a library.’

(1.6) Situation Nominalisation

Shizuku-wa [Seiji-ga baiairin-o hiku no]-o mita.
 S-TOP [S-NOM violin-ACC play NO]-ACC saw
 ‘Shizuku saw that Seiji played the violin.’

(1.7) Situation Nominalisation

Shizuku-wa [Yuko-ga Sugimura-o sukina no]-o shitteiru.
 S-TOP [Y-NOM S-ACC like NO]-ACC know
 ‘Shizuku knows that Yuko likes Sugimura.’

I shall offer functional, diachronic, cross-linguistic, and cross-dialectal motivations for analysing participant and situation nominalisation in a **unitary** way (Seraku in press c). The two types of *no*-nominalisation are uniformly characterised by the nominaliser *no* (Cann et al. 2005): *no* maps a proposition of the preceding clause onto a term that denotes an entity reflecting the proposition. My contention is that the two types of *no*-nominalisation are **reducible** to a parser's choice of what term (i.e., an individual term or a situation term) it copies in processing *no* (Seraku 2012b).

This uniform analysis is extended to novel data: a certain asymmetry emerges between the two types of *no*-nominalisation when **coordinate** clauses are nominalised by the particle *no*. It is suggested that this asymmetry would be difficult for previous studies to handle without giving up uniformity in explanation. By contrast, the dynamic account captures the contrast naturally **without** sacrificing uniformity. For this purpose, a Dynamic Syntax treatment of clausal coordination is presented (cf., Cann et al. 2005) and the situation term becomes more fine-grained so as to represent tense as the restrictor of the situation term (Cann 2011).

Chap. 5 turns to clefts, a paradigmatic construction involving the particle *no*. In the literature (Hoji 1990), it has been held that there are two types of clefts depending on whether a focus (i.e., a pre-copula item) has a case particle or not. Clefts with a case-marked focus are called ‘clefts_{+C},’ whereas clefts with a case-less focus ‘clefts_{-C}.’ The examples below form a minimal pair, differing solely in terms of case-marking of a focus expression, as emphasised in bold.

(1.8) Clefts_{+C}

[Shizuku-ga karita no]-wa **hon-o** san-satsu da.

[S-NOM borrowed NO]-TOP book-ACC 3-CL COP

‘It is three books that Shizuku borrowed.’

(1.9) Clefts_{-C}

[Shizuku-ga karita no]-wa **hon** san-satsu da.
 [S-NOM borrowed NO]-TOP book 3-CL COP
 ‘It is three books that Shizuku borrowed.’

I shall argue that the entries of the nominaliser *no* and the topic particle *wa* have formed a unit through routinisation of procedures for minimising costs required to construct a semantic structure for a cleft string. The unit *no-wa* lexically encodes the instruction to map the content of the pre-*no-wa* clause onto a propositional schema into which the content of the focus is subsequently incorporated. The form-function correspondences of the nominaliser *no* and the unit *no-wa* are depicted in Figure 1.1.

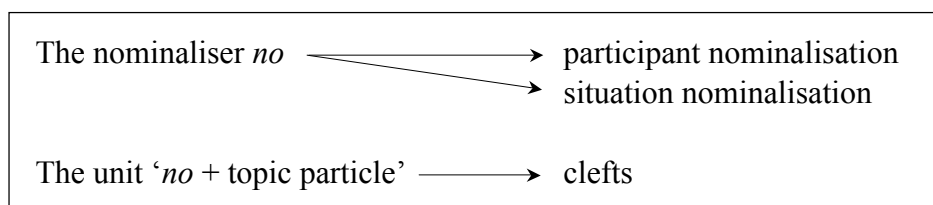


Figure 1.1. Form-function correspondences

The two types of clefts are uniformly characterised; they do not differ structurally but they reflect two distinct modes of resolving **structural underspecification** of the focus (Seraku 2011). That is, the node position of a focus in a semantic structure is initially underspecified and it is subsequently resolved by the parse of a case particle if the focus is case-marked (clefts_{+C}) or by the execution of the general action called SUBSTITUTION if the focus is case-less (clefts_{-C}). This unitary treatment is in harmony with the corpus finding that the presence of a case particle at a focus position is affected by pragmatic factors rather than structural factors (Chap. 2; see also Sunakawa 2005).

Given this unitary treatment, the question is how to account for differences between clefts_{+C} and clefts_{-C} with respect to (i) multiple foci and (ii) island-sensitivity. First, it has been held that multiple foci are possible only in clefts_{+C}. But this thesis observes that the **final** focus (and only the final focus) may lack a case particle as in

(1.10): the particle *o*, attached to the final focus *udon*, may be dropped, whereas the particle *ni*, attached to the non-final focus *Shizuku*, may not.

(1.10) Clefts with multiple foci

[Shiro-ga tsukutteageta no]-wa Shizuku*(-ni) udon(-o) da.
 [S-NOM served NO]-TOP S(-DAT) udon(-ACC) COP
 Lit. ‘It is udon_i to Shizuku_j that Shiro served e_i e_j.’

These data are problematic for any previous studies which stipulate different structures depending on whether a focus is case-marked (e.g., Hiraiwa & Ishihara 2012, Hoji 1990, Kizu 2005). This case-marking pattern follows from the dynamics of incremental parsing (Seraku 2012a, 2013b). Structural underspecification of each focus must be fixed **immediately** by parsing a case particle or the copula *da*-. If the case particle *ni* is not scanned in (1.10), the unfixed node of *Shizuku* remains unresolved and the nodes of *Shizuku* and *udon* lead to inconsistency in the node description.

As for the second difference, it has been widely assumed that clefts_{+C} are sensitive to syntactic islands, whereas clefts_{-C} are not (Hoji 1990). This discrepancy follows from the way the focus is processed. It was argued above that the focus was parsed by means of structural underspecification, but there is another tree update for processing the focus: LINK transitions (i.e., the mechanism of pairing discrete trees in virtue of a shared term). The LINK-based parsing of the focus is available to **both** types of clefts, but the tree will lead to a well-formed state **only** when the focus is case-less. This correctly predicts that island-involving clefts are grammatical (hence, not sensitive to islands) only when the focus lacks a case particle.

The analysis of clefts is applied to ellipsis constructions such as stripping (1.11) and sluicing (1.12). It is argued that the entry of the copula *da*- in clefts, together with the mechanism of ‘structural underspecification and resolution,’ unifies these ellipsis data. The analysis predicts (i) the same case-marking pattern of sluicing and stripping as with clefts and (ii) the same island-sensitivity pattern of sluicing and

stripping as with clefts. In particular, partial case-marking of sluicing and stripping is a novel observation of the thesis, and it constitutes problematic data for recent accounts (e.g., Fukaya 2007).

(1.11) Stripping

- A: Seiji-ga hon-o karita.
 S-NOM book-ACC borrowed
 ‘Seiji borrowed a book.’
- B: Shizuku-mo da.
 S-too COP
 ‘Shizuku, too.’ (= ‘Shizuku also borrowed a book.’)

(1.12) Sluicing

- Sugimura-ga dareka-o suki-rashii ga,
 S-NOM someone-ACC like-seem but
 Shizuku-wa dare(-o) (da) ka wakara-nai.
 S-TOP who(-ACC) (COP) Q know-NEG
 ‘It seems Sugimura likes someone, but Shizuku does not know who.’

The analyses of *no*-nominalisation and clefts are intertwined in Chap. 6, where we will address several relative constructions involving *no*: (i) Free Relatives (FRs), (ii) Head-Internal Relatives (HIRs), and (iii) Change Relatives (CRs).

(1.13) Free Relatives (FRs)

- Shizuku-wa [Shiro-ga tsukuttekureta no]-o tabeta.
 S-TOP [S-NOM served NO]-ACC ate
 ‘Shizuku ate what Shiro served to her.’

(1.14) Head-Internal Relatives (HIRs)

- Shizuku-wa [Shiro-ga udon-o tsukuttekureta no]-o tabeta.
 S-TOP [S-NOM udon-ACC served NO]-ACC ate
 ‘Shiro served udon to Shizuku, and she ate it.’

(1.15) Change Relatives (CRs)

[Ningyou-ga shinshi-ni natta no]-ga Shizuku-to

[doll-NOM gentleman-COP become NO]-NOM S-with

rapisurazuri-no-komyaku-e mukatta.

Lapis.Lazuli-GEN-vein-to went

‘A gentleman who is the result of changing from a doll went to the vein of Lapis Lazuli with Shizuku.’

All of these relative constructions involve the nominaliser *no*, and they are accounted for based on the analyses of *no*-nominalisation and clefts.

The FR is a construction where the head is missing and the relative clause part is followed by the nominaliser *no* (Hoshi 1995). The FR (1.13) does look quite similar to participant nominalisation (1.5). In fact, they are **identical**; ‘free relatives’ is another term for what I dub ‘participant nominalisation.’ Thus, the analysis of the FR is already in our hands, but the present chapter reveals further characteristics of the FR by comparing it with the HIR.

The HIR is a construction in which the head (e.g., *udon* in (1.14)) appears to be inside the relative clause (Kuroda 1992). Despite surface similarities between the FR (1.13) and the HIR (1.14), they display a number of non-surface differences, which indicates that the analysis of the FR is not directly applicable to the HIR. I shall propose that the HIR clause and the case particle jointly determine a ‘rich’ environment (i.e., partially articulated tree) for the immediately embedding clause which follows them in the sequence (Seraku in press b). These tree transitions are driven by the unit ‘*no* + case particle.’ I shall define the entry of the unit by blending the type-e node creation process (i.e., part of the entry of the nominaliser *no*) with the pairing of propositional trees (i.e., part of the entry of the cleft marker *no-wa*). This lexical encoding maps the tree of the HIR clause onto another propositional tree which has been **partially articulated** when it is introduced. This partial tree contains two nodes. First, a situation node is decorated with the requirement that the situation term will stand in a certain relation to the

situation term of the embedding tree. This models the Relevancy Condition on the HIR construal (Kuroda 1992). Second, the partial tree contains a node for an internal head. This ensures that the HIR head, though internal to the relative clause, is selected by the embedding predicate. This dynamic analysis accounts for numerous data, including those problematic for recent analyses of HIRs (e.g., Grosu 2010, Grosu & Landman 2012, Kim 2007). For instance, it has been widely assumed that negation cannot be licensed within HIR clauses (Grosu & Landman 2012, Hoshi 1995), but it is licensed if a context is accessible which allows the quantified head to out-scope the negator.

Finally, the CR is a relative clause construction where the ‘change of state’ is denoted as in (1.15). Expanding on Tonosaki (1998), I shall argue that the CR and the FR are to be analysed **uniformly**. That is to say, the CR is modelled by the mechanism of *no*-nominalisation.

The overall conclusion is drawn in Chap. 7. As a summary of the analyses sketched, the form-function correspondences of the key items are clarified below:

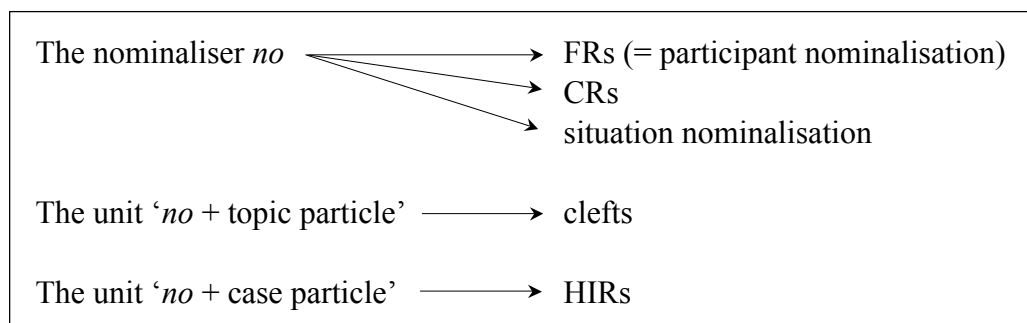


Figure 1.2. Form-function correspondences

The entry of the cleft marker *no-wa* is defined by combining the entry of *no* (Sect. 3, Chap 4) and the entry of the topic marker *wa* (Sect. 5.3, Chap. 3). The entry of the HIR marker ‘*no* + case particle’ is defined based on part of the entry of *no* (i.e., the type-e node building process), part of the entry of *no-wa* (i.e., the pairing of two propositional nodes), and part of the entry of case particles (i.e., the identification of a node position within a local tree). The procedures encoded in these items are spelt out below.

- The nominaliser *no* maps a proposition of the preceding clause onto a **term** that reflects the proposition.
- The cleft marker ‘*no* + topic particle’ maps a proposition of the preceding clause onto an **emergent propositional tree** where the focus will be parsed.
- The HIR marker ‘*no* + case particle’ maps a proposition of the preceding clause onto a **partial propositional tree** where the immediately embedding clause will be parsed.

As a future research avenue, Chap. 7 speculates that the dynamic account of the *no*-involving phenomenon might be systematised under the general theory of what may be dubbed ‘Dynamic Construction Grammar.’

Finally, the appendix is a supplement to Chap. 3, and it formally defines the Dynamic Syntax architecture by constructing a mini grammar of Japanese.

Chapter 2

Empirical Scaffolding

1 Structural Properties of Japanese

This chapter sets out empirical foundations for the present thesis by drawing data from a corpus. As a preliminary, this opening section illustrates basic structural characteristics of Japanese with artificial examples.

Japanese lacks determiners, and the quantificational force and plurality of bare nouns must be contextually determined. Thus, as in (2.1), the bare noun *ringo* may be interpreted in four ways: indefinite/definite and singular/plural. Since our concern is with structure, only one of these interpretations per example is presented (typically, the indefinite singular interpretation (e.g., ‘an apple’)).

- (2.1) *ringo*
 apple
 ‘an apple’/‘apples’/‘the apple’/‘the apples’

The basic word order is Subject-Object-Verb (SOV), as shown in (2.2). Each argument is case-marked by a particle. The subject *Porco* is marked by the nominative-case particle *ga*, and the object *ringo* is marked by the accusative-case particle *o*. The

verb *tabe-* is suffixed by the past tense marker *ta*. Although a subject is canonically marked by *ga*, a matrix subject is often marked by the topic particle *wa*, as in (2.3).¹

- (2.2) Porco-ga ringo-o tabe-ta.
 P-NOM apple-ACC eat-PAST
 ‘Porco ate an apple.’

- (2.3) Porco-wa ringo-o tabe-ta.
 P-TOP apple-ACC eat-PAST
 ‘Porco ate an apple.’

Though the basic word order is SOV, Japanese is known to be a relatively free word-order language. That is, argument NPs are freely permutable as long as V comes at the end, so-called ‘scrambling.’ So, the following string is grammatical with the same truth conditional content as (2.2).

- (2.4) Ringo-o Porco-ga tabe-ta.
 apple-ACC P-NOM eat-PAST
 ‘Porco ate an apple.’

The verb-finality is an instance of the general feature of the language, the head-finality. Japanese employs postpositions (rather than prepositions); in (2.5), the NP *teiburu* (= ‘table’) is marked by the postposition *ni* (= ‘at’). A caveat, however, is that what are called ‘postpositions’ here are bound morphemes, and they might be thought of as a special type of case particle assigning oblique case to the item.

- (2.5) Ringo-ga teiburu-ni oitea-tta.
 apple-NOM table-on be.placed-PAST
 ‘An apple was placed on a table.’

¹ There are complicated issues regarding the *ga/wa* divide. See Heycock (2008), Kuno (1973a), and Noda (1996), among many others.

A noun is preceded (but not followed) by modifiers such as the demonstrative *sono* (= ‘that’) (2.6) and the adjective *midorino* (= ‘green’) (2.7).

- (2.6) Porco-ga [sono ringo]-o tabe-ta.
 P-NOM [that apple]-ACC eat-PAST
 ‘Porco ate that apple.’

- (2.7) Porco-ga [midorino ringo]-o tabe-ta.
 P-NOM [green apple]-ACC eat-PAST
 ‘Porco ate a green apple.’

The head-finality is also true of relative clause constructions: in (2.8), the head noun *ringo* is preceded by the relative clause *Porco-ga kaji-tta*. In connection with this, it should be noted that Japanese lacks a relative clause marker; a head noun comes after a relative clause without any morphological indications of subordination.

- (2.8) [Porco-ga kaji-tta ringo]-ga teiburu-ni oitea-tta.
 [P-NOM bite-PAST apple]-NOM table-on be.placed-PAST
 ‘An apple which Porco bit was placed on a table.’

However, a morphological marker is used for certain types of subordination. For instance, clause embedding is marked by the complementiser *to*, though unlike the English complementiser *that*, *to* cannot be used to mark a clause in a subject position; in this case, the clause must be marked by other particles such as *no* (Chap. 4).

- (2.9) Gina-wa [Porco-ga buta-ni na-tta to] i-tta.
 G-TOP [P-NOM pig-COP become-PAST COMP] say-PAST
 ‘Gina said that Porco became a pig.’

- (2.10) [Porco-ga buta-ni na-tta *to/no]-wa hontouda.
 [P-NOM pig-COP become-PAST COMP/NO]-TOP true
 ‘It is true that Porco became a pig.’

Japanese is fully pro-drop. Any arguments may be omitted as long as they are contextually retrievable. Thus, in an appropriate context, the truth-conditional content of the string (2.4) could equally be expressed by any of the following strings, depending on the additional information needed being available in context:

(2.11) Porco-ga tabe-ta.
P-NOM eat-PAST
'Porco ate an apple.'

(2.12) Ringo-o tabe-ta.
apple-ACC eat-PAST
'Porco ate an apple.'

(2.13) Tabe-ta.
eat-PAST
'Porco ate an apple.'

Recall that there is no morphological indication of relative clauses. Thus, together with the fact that Japanese is fully pro-drop, the relative clause part in (2.8), repeated here as (2.14), has exactly the same surface string as the simple sentence (2.15).

(2.14) [Porco-ga kaji-tta ringo]-ga teiburu-ni oitea-tta.
[P-NOM bite-PAST apple]-NOM table-at be.placed-PAST
'An apple which Porco bit was placed at a table.'

(2.15) Porco-ga kaji-tta.
P-NOM bite-PAST
'Porco bit something (or a contextually salient object).'

These illustrations are intended to be useful as a reference point for basic properties of Japanese. For a large-scale description of the language, see, e.g., Frellesvig (2010), Kuno (1973a), Martin (1975), Shibatani (1990), and Tsujimura (2007).

2 Methodological Preliminaries

Continuing our illustration of Japanese, the rest of this chapter delves into more specific issues by presenting spontaneous data collected from a corpus. Though the main target of this corpus work is cleft constructions, non-cleft data were also extracted and are reported here. Before presenting the corpus data, however, this section introduces the corpus employed and the search methodology.

2.1 Corpus of Spontaneous Japanese

The present work utilised the Corpus of Spontaneous Japanese (CSJ; 2nd edition), developed by the National Institute for Japanese Language and Linguistics in 2005, the 2nd edition being distributed since 2008. The CSJ is a digitalised collection of spoken data in contemporary Japanese (largely, Tokyo dialect), storing about 7.52 million words (amounting to about 660 hours). They are fully transcribed and annotated (e.g., morphological information) in Japanese in the XML format. Out of the 7.52 million words, 0.5 million words are selected as ‘core’ and these ‘core’ data come with other sets of annotations (e.g., agreement information, intonational labels).

The data in the CSJ are of three types: (i) monologue, (ii) recitation, and (iii) dialogue. ‘Monologue’ accounts for more than 90% of the whole data, and the rest is occupied by ‘recitation’ or ‘dialogue.’ Inevitably, the degree of spontaneousness varies across categories. As a general indication, the least spontaneous is ‘recitation’ and the most spontaneous is ‘dialogue,’ with ‘monologue’ being somewhere in the middle.²

The category ‘monologue’ consists of ‘Academic Presentation Speech’ (APS) and ‘Simulated Public Speaking’ (SPS). In APS, talks in academic conferences were recorded. Most participants are male graduate students. In SPS, people were asked to

² The CSJ data have been evaluated for the degree of spontaneousness. For each recording, one person chooses out of 5-scale ranking for spontaneousness and formality of speech. This is supplemented by a more rigorous measure, called ‘set evaluation.’

deliver a 10~15 minute public speech about three general themes (e.g., ‘hometown’). Unlike APS, the participants in SPS are controlled for gender and age.

The category ‘recitation’ is divided into ‘Recitation’ and ‘Re-recitation.’ In ‘Recitation,’ some people were selected from SPS, and they were asked to read written texts (e.g., essays, novels). In ‘Re-recitation,’ some of the APS and SPS participants were asked to re-read the transcribed texts of their presentations.

Under the category ‘dialogue’ are there four sub-types. In ‘Interview at APS’ and ‘Interview at SPS,’ a woman interviewed some participants of APS and SPS. In ‘Task-oriented Dialogue,’ each interviewer-interviewee pair was assigned a list of 9~10 celebrities, and asked to guess how much they should pay for each celebrity if they were to ask him or her to give a speech. In ‘Free Dialogue,’ each interviewer-interviewee pair talked about any topic for about 10 minutes.

Since the CSJ itself does not have a concordancer, I used the search engine Himawari (version 1.3), developed by the National Institute for Japanese Language and Linguistics, the version 1.3 being distributed since 2011. Himawari searches the XML files in the CSJ for a specified string, and outputs a list of the target strings in the Key-Word-In-Context (KWIC) style. The detail of the search procedures is summarised in the next sub-section.

2.2 Search Procedures

The main target of my corpus work is cleft constructions. Given that clefts involve the nominaliser *no*, the obtained data contain examples of *no*-nominalisation and other related constructions.

A simple, artificial example of clefts is (2.16). There is no morphological marker that is tailored for clefts, but the succession of *no-wa* (i.e., のは in Japanese characters) serves as a clue. Thus, the CSJ was searched for the sequence *no-wa*.

(2.16) Clefts

[Porco-ga kettou-o shita no]-wa Curtis da.
 [P-NOM duel-ACC did NO]-ACC C COP
 ‘It is with Curtis that Porco duelled.’

The reader may wonder whether it is possible to specify the cleft schema by taking into consideration the copula *da-* in addition to *no-wa*, as in ‘X-*no-wa* Y-*da*’ with X and Y being underspecified for the number of items and the category of each item. This schematic characterisation, however, does not work. This is partly because the copula *da-* may be sometimes dropped in spontaneous speech. In (2.17), where the speaker talks about how to give a good speech, the copula *da-* is missing at the end of the string.

(2.17) Clefts, extracted from the CSJ (S00F0208)³

[kunren-ga hitsuyouna no]-wa ittemireba subeteno-hito...
 [training-NOM is.required NO]-TOP as.it.were every-person
 Lit. ‘It is, as it were, every person who needs training...’

Another problem is that the number of strings in the Y part may be quite large. In (2.18), where the speaker talks about her favourite Japanese TV drama, the *no-wa* part is immediately followed by the adverbial clause, and then by the focus part, which is underlined. This focus part itself is a complex NP, where the head noun *dorama* (= ‘drama’) is modified by the preceding relative clause. (In the relative clause, the name of an actor or an actress is abbreviated as A, B, and C.) In order to accommodate this type of example, the Y part in the schema ‘X-*no-wa* Y-*da*’ needs to be underspecified vacuously weakly for the number of strings.

³ When examples are drawn from the CSJ, I cite in the top line the ID number assigned to the file of the example in question. Inessential annotations (e.g., meta-linguistic annotations, fillers) are omitted in the cited version, and glosses and free translations are provided by the author.

(2.18) Clefts, extracted from the CSJ (S00F0178)

[miteita no]-wa [korewa kioku-ni-mo atarashii ndesu]-keredomo
 [watched NO]-TOP [this memory-at-too new COP]-though
 [ninen-mae A-to-B-soshite-C-ga deteita sweet-season toiu drama]
 [two.year-ago A-and-B-and-C-NOM played sweet-season TOIU drama]
 nandesu...
 COP

‘It is (though this is a recent memory) a drama called ‘sweet season’ that I watched, in which A, B, and C played a role two years ago.’⁴

To make matters worse, the adverbial clause in (2.18) itself contains the copula *da-*. Thus, if the schema ‘X-*no-wa* Y-*da*’ were applied to this example, the whole string could not be extracted due to the non-final occurrence of the copula *da-*.

Once the *no-wa* sequences in the CSJ had been extracted, they were manually classified into clefts and other constructions. This process is essential because the obtained set contains a number of non-cleft strings. In (2.19), the *no-wa* sequence is part of the *wa*-marked NP *mono-wa*, where *mono* is a lexical noun meaning ‘thing.’

(2.19) *Mono-wa*, extracted from the CSJ (S00F0090)

[watashi to haha]-wa mono-wa zenzen sutete...
 [I and mother]-TOP thing-TOP absolutely throw.away
 ‘I and my mother throw away things absolutely...’

The exclusion of this type of irrelevant example is still not enough since there are a number of cases which may be related to clefts but which themselves are not clefts. In (2.20), where the speaker talks about her writing of haiku poems, the string is not an instance of clefts but what is called ‘participant nominalisation’ in Chap. 4 (or what is re-named as ‘free relatives’ in Chap. 6).

⁴ To preserve the comprehensibility of the free translation, the relative modifier of the focussed expression is extraposed to the right of the presupposition clause.

(2.20) Participant nominalisation, extracted from the CSJ (S04F1495)

[sunaoni tsuku-reta no]-wa angai home-rareru...

[naïvely make-could NO]-TOP unexpectedly praise-PASS

‘As for what I could write naïvely, it is unexpectedly praised...’

In this way, two steps are important for extracting cleft strings exhaustively: (i) to search the CSJ for the sequence *no-wa* (i.e., のは in Japanese characters) and (ii) to classify the obtained data manually.

3 Overall Results

Following the search procedures in the last section, 2,733 strings with the sequence *no-wa* were extracted from the CSJ, and they were manually classified into several constructions. The result is in Table 2.1, where *toiuno* and *toiunowa* are ‘compound particles,’ as will be illustrated shortly; see (2.21) and (2.22) below. Each percentage is rounded to the nearest whole number; percentages may not add up to 100%, therefore the total percentage cell is blank in this and subsequent tables.

Construction Type	Occurrences	Percentage
<i>no</i> -nominalisation	257	9%
clefts	322	12%
<i>toiuno</i>	1,662	61%
<i>toiunowa</i>	44	2%
pronoun	50	2%
<i>mono</i> (= ‘thing’)	245	9%
other data	24	1%
unclassifiable	129	5%
Total	2,733	

Table 2.1. Overall result of the corpus search

No-nominalisation is discussed in Sect. 4 of this chapter, and a dynamic account of this construction will be presented in Chap. 4. Clefts are examined in Sect. 5 of this chapter, and a dynamic account of clefts will be developed in Chap. 5.

Turning to the other constructions in Table 2.1, what is striking is that a large proportion is occupied by *toiuno* and *toiunowa*. These are called ‘compound particles’ (Fujita & Yamazaki 2006). *Toiuno* literally consists of the sentential complementiser *to*, the verb *iu-* (= ‘say’), and the nominaliser *no*, but the complex as a whole functions as a particle. In (2.21), *Nerimaku*, a name of a region in Tokyo, is followed by *tteiuno* (a phonological variant of *toiuno*), which gives a ‘hearsay’ flavour; note the use of ‘called’ in the free translation. *Toiunowa* literally consists of the compound particle *toiuno* and the topic marker *wa*, but the complex serves as a discourse marker. In (2.22), *tteiunowa* (a phonological variant of *toiunowa*) occurs sentence-initially, unlike *tteiuno* in (2.21). *Tteiunowa* in this example may be translated into ‘that is to say.’

(2.21) *Toiuno*, extracted from the CSJ (S03M0089)

[Nerimaku tteiuno]-wa hoka-no-tokoro-ni kuraberuto hijouni
 [Nerimaku TOIUNO]-TOP other-GEN-place-at being.compared very
 chian-ga ii...
 public.security-NOM good

‘As for the region called Nerimaku, its public security is very good, compared with other regions in Tokyo...’

(2.22) *Toiunowa*, extracted from the CSJ (S01F0217)

tteiunowa watashi-wa shougakkou-jidai-ni chichioya-no
 TOIUNOWA I-TOP primary.school-period-at father-GEN
 tenkin-de firipin-ni sundemashita...
 job.relocation-at Philippine-in lived

‘That is to say, when I was a primary student, I lived in Philippine due to my father’s job relocation...’

These particles *toiuno* and *toiunowa* are not our direct concern, but some of the data are briefly discussed in connection with *no*-nominalisation in Chap. 4.⁵

⁵ Seraku (2013a) observes parallelisms between the compound particle *toiuno* and the construct-state genitive *ešet* in Hebrew (Falk 2001b) and proposes a Lexical Functional Grammar account of *toiuno*. For details, see Seraku (2013a).

There are 50 cases in which *no* serves as a pronoun. In (2.23), the speaker says of his son that his score in the entrance examination for high school was high but he subsequently became the student with the poorest record in the school. Here, the particle *no* is not a nominaliser because the preceding item *sono-toki* (= ‘that time’) itself is a nominal. Rather, the particle *no* in (2.23) is a pronoun, denoting the academic record of the speaker’s son.

(2.23) Pronoun, extracted from the CSJ (S02F0049)

[sono-toki no]-wa ichiban-shita de...

[that-time NO]-TOP most-low COP

‘His academic record at that time was the lowest in school...’

The view that *no* in (2.23) is a pronominal but not a nominaliser is diachronically plausible. Wrona (2012) reports that there is a time lag of at least 500 years between the first attestation of data like (2.23) and that of *no*-nominalisation (see Chap. 4).

There are 245 cases where *no* is part of the lexical noun *mono* (= ‘thing’), which is followed by the topic marker *wa*. This type of example has been presented in (2.19), as reproduced here as (2.24). Here, *no* is not a particle but just part of a lexical noun, hence it is not relevant to our inquiry into *no*-involving constructions.

(2.24) *Mono*, extracted from the CSJ (S00F0090)

[watashi to haha]-wa mono-wa zenzen sutete...

[I and mother]-TOP thing-TOP absolutely throw.away

‘I and my mother throw away things absolutely...’

There are even more irrelevant examples, where both *no* and *wa* are part of a lexical item. The CSJ contains 24 such cases. In (2.25), the sequence *no-wa* is present due to the combination of the final character of *ittokino* (= ‘temporary’) and the initial character of *hayari* (= ‘fashion’). As in the last type of example shown in (2.24), these cases are not pertinent to our study.

(2.25) Other data, extracted from the CSJ (A05F0039)

ittokino hayari⁶
 temporary fashion
 ‘temporary fashion’

Finally, there are 129 cases which are hard to classify. For instance, in (2.26), it seems that the speaker plans to produce a string that instantiates clefts or situation nominalisation but after the filler *uun*, this initial plan appears to have been changed and the string starts to take on another construction.

(2.26) Unclassifiable, extracted from the CSJ (D01F0023)

[Oubei-no hito-ga ita no]-wa,
 [Europe.and.American-GEN person-NOM existed NO]-TOP
 uun, mada ira-ssharu-ndesune...
 well, still exist-POLITE-COP
 ‘European and American people existed, well, they still exist...’

Setting aside such examples as (2.21)-(2.26), the rest of this chapter focuses on *no*-nominalisation (Sect. 4) and cleft constructions (Sect. 5).

4 *No*-Nominalisation

4.1 Introduction

As will be mentioned in Chap. 4, *no*-nominalisation is divided into two sub-types. In participant nominalisation, *no* takes the preceding clause and denotes an individual (e.g., person, object), or a participant of a situation denoted by the clause. In (2.27), what is denoted by the *no*-part is an object (e.g., a bottle of juice) in a situation described by the clause *Porco-ga katte-kureta*. In situation nominalisation, *no* takes the

⁶ In Japanese characters, *no-wa* is written as のは. In (2.25), the initial mora of the NP *hayari* (= ‘fashion’) is notated as *ha*, but it is written as は in Japanese characters.

preceding clause and denotes a situation described by the clause. In (2.28), the *no*-part denotes an event in which Gina is singing a chanson.

(2.27) Participant nominalisation

Fio-wa [Porco-ga katte-kureta no]-o nonda.
 F-TOP [P-NOM buy-gave NO]-ACC drank
 ‘Fio drank what Porco bought for her.’

(2.28) Situation nominalisation

Porco-wa [Gina-ga shanson-o utau no]-o kiita.
 P-TOP [G-NOM chanson-ACC sing NO]-ACC heard
 ‘Porco heard Gina singing a chanson.’

The proportion of these sub-types in the total of 257 instances of *no*-nominalisation is shown in Table 2.2.

Type	Occurrences	Percentage
participant nominalisation	95	37%
situation nominalisation	162	63%
Total	257	

Table 2.2. *No*-nominalisation in the CSJ

Pending a detailed account to be given in Chap. 4, the present section illustrates the two types of *no*-nominalisation with spontaneous data. It should be noted, however, that this corpus work has a practical limitation. In (2.27)-(2.28), the *no*-part is marked by the accusative-case particle *o*; in fact, the *no*-part may be marked by a wide range of case particles as well as the topic particle *wa*. In my data, the *no*-part is solely marked by the topic particle *wa*, given the search procedure to specify the *no-wa* sequence.

4.2 Definiteness

As stated in Sect. 1, Japanese lacks determiners, and definiteness of bare nouns must be pragmatically determined. This is true of *no*-nominalisation. That is to say, it must be contextually resolved whether the *no*-part denotes a definite or indefinite entity.

In participant nominalisation (2.29), the *no*-part denotes indefinite objects. The speaker talks about tombs in the Yayoi period, and states that big tombs are often found in the western part of the Izumo district. It seems that the *no*-part denotes big tombs without referring to specific ones.

(2.29) Participant nominalisation, extracted from the CSJ (A01M0077)

[ookii no]-wa Izumo-no-nishi-nohouni desune shuuchuushi...

[big NO]-TOP I-GEN-west-around well concentrated

‘As for what are big, they are concentrated around the west of Izumo...’

In (2.30), the speaker talks about an aiding system for those people who have hearing impairments. The *no*-part denotes particular images or graphs that are placed within the parentheses, as signalled by the use of the demonstrative *kono* (= ‘this’).

(2.30) Participant nominalisation, extracted from the CSJ (A01M0152)

kono [kakko-ni aru no]-wa kore-ga kidou de

this [parenthesis-in exist NO]-TOP this-NOM air.conduction COP

kochira-ga kotsudou-no-chouryoku...

this-NOM bone.conduction-GEN-hearing

‘As for these in the parentheses, this is an air conduction and this is the hearing ability of a bone conduction...’

Turning to situation nominalisation, in (2.31), the speaker talks about her cat, saying that it was unfortunate to let the cat sleep outside since it was getting cold. The speaker describes an event where the cat sleeps outside in general, without having in mind any particular events.

- (2.31) Situation nominalisation, extracted from the CSJ (S01F1522)

[yoru sotode ne-sen no]-wa kawai sou...
 [night outdoor sleep-CAUSE NO]-TOP pity
 ‘As for letting the cat sleep outside, it is a pity...’

In (2.32), the speaker talks about a windmill that was built in front of the station in his hometown. The matrix predicate *ii-* (= ‘fine’) applies to a particular event, the event of the windmill’s having been constructed.

- (2.32) Situation nominalisation, extracted from the CSJ (S00M0228)

[sono-suisha-ga n suisha tsukutta no]-wa ii-ndesu-kedo...
 [that-windmill-NOMwell windmill created NO]-TOP fine-COP-though
 ‘As for the event of having created the windmill, it is fine, but...’

In this way, both participant and situation nominalisation may denote either definite entities or indefinite entities, the distinction being resolved pragmatically. An account of *no*-nominalisation must reflect the flexibility that allows this grammar-pragmatics interaction.

4.3 Grammatical Function of the *No*-Part

In the CSJ data, the *no*-part is marked by the topic particle *wa*. Since a topicalised element may bear a grammatical function (e.g., subject, object), it would be worth investigating the type of grammatical function of the *no*-part.

Let us start with participant nominalisation. As shown in Table 2.3, the *no*-part has a subject function in 39 cases, while it has an object function only in 13 cases. This contrast is not unexpected because a matrix subject in Japanese is often marked by the topic marker *wa* (see Sect. 1). Furthermore, there are 43 cases where the *no*-part purely presents a topic of the string and does not bear any grammatical function that may be clearly recognised.

Type	Occurrences	Percentage
subject	39	41%
object	13	14%
pure topic	43	45%
Total	95	

Table 2.3. Grammatical function of the *no*-part in participant nominalisation

In (2.33), where the speaker talks about Japanese dictionaries, the *no*-part functions as a subject of the matrix predicate *waruku-wa-nai*- (= ‘not bad’).

- (2.33) Participant nominalisation, extracted from the CSJ (M02M0001)
 [kokoe dashiteru no]-wa zenbu waruku-wa-nai...
 [here present NO]-TOP all bad-TOP-NEG
 ‘As for what I present here (= dictionaries), they are all not bad...’

In (2.34), the speaker talks about her first experience of performing a play in her life. In this case, the *no*-part is a nominal denoting a pleasant feeling that the speaker had in her performance, and it is an object of the matrix verb *e*- (= ‘gain’).

- (2.34) Participant nominalisation, extracted from the CSJ (S01F0137)
 [hito-no-mae-ni tatte nanika-o suru toiu koto]-no kaikan
 [person-GEN-front-at stand something-ACC do TOIU thing]-GEN pleasure
 mitai-na no]-wa moshikashitara sokode saishoni
 like-COP NO]-TOP perhaps there for.the.first.time
 e-ta-nokamoshiremasen...
 gain-PAST-may
 ‘As for what may be said to be the pleasure of doing something in front of people, I may have gained it there...’

Compared to the examples above, there are a number of cases where the grammatical function of the *no*-part cannot be clearly determined. In these cases, the *no*-part seems to purely present a topic of the string. In (2.35), the speaker talks about information hiding in cryptography. The *no*-part raises a topic of famous fields in which information

hiding techniques are used, relative to which the rest of the string mentions mainstream areas such as images and sounds.

(2.35) Participant nominalisation, extracted from the CSJ (A03M0111)

[yuumeina no]-wa gazou-toka-onsei-no-bunya-ga shuryuu de...

[famous NO]-TOP image-and-sound-GEN-area-NOM mainstream COP

‘As for what are famous, the areas of images and sounds are the mainstream...’

A similar example is (2.36), where the speaker talks about her experience of teaching Japanese to those who came from China in a volunteer centre. The *no*-part mentions the audience of Japanese lessons as a topic of the string, and the rest of the string says of it that the majority is adults (rather than, say, students).

(2.36) Participant nominalisation, extracted from the CSJ (S03F0214)

[doyoubi-ni kiteru no]-wa shakaijin-ga ooi...

[Saturday-on come NO]-TOP adults-NOM abundant

‘As for those who come to the centre on Saturday, most of them are adults...’

A more complex example is (2.37), where the speaker talks about a bar graph in a speech recognition study. In this case, the association between the topic and the rest of the string is indirect. The *no*-part refers to some long bars in the graph, and the rest of the string states that the speaker collected the materials that are represented as these bars; that is, what the speaker gathered are materials, not the bars in the graph.

(2.37) Participant nominalisation, extracted from the CSJ (A11M0469)

[nagai no]-wa kore-wa webu-jou-kara atsumete-kimashita...

[long NO]-TOP this-TOP web-on-from collect-came

‘As for what are long in this bar graph, they were collected from the web.’

Turning to situation nominalisation, Table 2.4 shows the result. The proportion of subjects is relatively high; there are 113 such cases, which accounts for about 70% of the whole data. This is clearly comparable with 39 cases of participant nominalisation,

which accounts for 41% of the whole. The number of cases where the *no*-part functions as an object is quite small; there are only 9 examples, which occupies 6% of the whole. Finally, there are 40 cases, in which the *no*-part seems to present a pure topic.

Type	Occurrences	Percentage
subject	113	70%
object	9	6%
pure topic	40	25%
Total	162	

Table 2.4. Grammatical function of the *no*-part in situation nominalisation

In (2.38), the speaker talks about natural language processing, and the demonstrative *kore* refers to a syntactic analysis of ‘bunsetsu’ structure. The *no*-part functions as a subject of the matrix predicate *muzukashii*- (= ‘difficult’).

- (2.38) Situation nominalisation, extracted from the CSJ (A03M0111)
 [nakanaka kore-o kore-o yaru no]-wa muzukashii...
 [highly this-ACC this-ACC do NO]-TOP difficult
 ‘As for doing a syntactic analysis of “bunsetsu” structure, it is a bit difficult...’

In (2.39), where the speaker talks about the piano, the *no*-part serves as an object of the verb *akirame*- (= ‘give up’).

- (2.39) Situation nominalisation, extracted from the CSJ (S00F0209)
 [[purei-suru no]-wa mou akirame-you to] omotte...
 [[play-do NO]-TOP yet give.up-let’s COMP] think
 ‘I’m thinking that as for playing the piano, I give it up...’

Finally, there are a number of examples where the grammatical function of the *no*-part is not easily determined; in these examples, the *no*-part seems to present a pure topic. In (2.40), where the speaker talks about his hometown, the *no*-part introduces the fact that the speaker has lived for a long time at a certain place, and the rest of the string specifies the place as the city of Osaka.

(2.40) Situation nominalisation, extracted from the CSJ (S00M0009)

[koko-ga nagaku jikantekini sunden no]-wa Osaka-ni

[here-NOM for.long.time temporally live NO]-TOP O-in

zutto sunde-ta-ndesu-keredomo...

for.long.time live-PAST-COP-though

Lit. ‘As for the fact that here is where I have lived for a long time, I have lived in Osaka for a long time, but...’

In (2.41), where the speaker talks about horses, the *no*-part supplies the information to understand the following nouns *rouryoku* (= ‘effort’) and *shikin* (= ‘funds’). That is, the *no*-part presents a topic of fostering a horse, and those subsequent nouns are interpreted as denoting effort and funds which are required to foster a horse.

(2.41) Situation nominalisation, extracted from the CSJ (S00F0131)

[uma i-ttou yashinau no]-wa tashikani taihenna

[horse 1-CL foster NO]-TOP definitely much

rouryoku-to-shikin-ga hitsuyou desu...

effort-and-fund-NOM be.required COP

‘As for fostering even a single horse, a lot of effort and funds are definitely required...’

In both types of *no*-nominalisation, the *no*-part may have core grammatical functions such as subject and object. This confirms our initial characterisation of *no*-nominalisation that the *no*-part is a nominal denoting some entity.⁷ As for the pure topic, it would be a reflection of the topic marker *wa*, which associates the *wa*-part with the rest of the string by an ‘aboutness’ relation (Kuno 1973a). Another noticeable point is that the type of entities denoted in situation nominalisation is quite diverse, including an action (2.38), a perceived situation (2.40), and some type of statement (2.41).

⁷ This is not a trivial conclusion because it will be pointed out in Chap. 6 that the *no*-part in the head-internal relative clause construction cannot be construed in this way.

4.4 Grammatical Function of the Gap

The last sub-section showed the distribution of grammatical function of the *no*-part. It would also be informative to consider the grammatical function of a gap in the pre-*no* clause. Given that a gap is not involved in situation nominalisation, Table 2.5 concerns only participant nominalisation. In addition to core grammatical functions, there are two types of adjuncts whose functions I call *function*_{TIME} and *function*_{PLACE} respectively.

Type	Occurrences	Percentage
subject	56	59%
object	32	34%
<i>function</i> _{TIME}	3	3%
<i>function</i> _{PLACE}	2	2%
unclassifiable	2	2%
Total	95	

Table 2.5. Grammatical function of the gap in participant nominalisation

In (2.42), the speaker talks about so-called ‘doutaku’ (i.e., bronze bell-shaped vessels, created during the Yayoi period). In this example, the gap is the subject of the embedded predicate *furui*- (= ‘old’).

- (2.42) Participant nominalisation, extracted from the CSJ (A01M0077)
- kono-guruupu-wa, [ichiban furui no]-wa arima-sen-keredomo...
- this-group-TOP [most old NO]-TOP exist-NEG-though
- ‘As for this group, what is the oldest is not here, but...’

In (2.43), the speaker talks about phonetic experiments. The gap, namely the object of the embedded verb *bunsekishi*- (= ‘analyse’), has the object function.

- (2.43) Participant nominalisation, extracted from the CSJ (A05M0118)
- [jissaini bunsekishi-ta no]-wa doumo sukoshi takame
- [actually analyse-PAST NO]-TOP seemingly a.bit high
- noyou-nandesu...
- seem-COP
- ‘As for what we actually analysed, their values seem to be a bit high...’

There are 5 cases in which the gap is an adjunct. In 3 cases, the gap has the function_{TIME}. In (2.44), the speaker talks about preliminary rounds of the Tokyo area for All Japan High School Soccer Tournament. The pre-*no* part *yosen-ga hajimaru* is a complete clause in that the argument of the intransitive verb *hajimaru*- (= ‘start’) is supplied by the noun *yosen* (= ‘preliminary round’). Thus, there is no argument gap. Instead, what could be identified as a gap is the time adjunct gap, concerning the time at which the preliminary rounds are kicked off.

(2.44) Participant nominalisation, extracted from the CSJ (S00M0059)

[yosen-ga hajimaru no]-wa natsu-no-owari-kara
 [preliminary.round-NOM start NO]-TOP summer-GEN-end-since
 hajima-tte fuyu-no 12-gatsu-no-mae-gurai-made tsuzukimasu.
 start-and winter-GEN 12-month-GEN-before-about-until continue
 ‘As for the time at which the preliminary rounds start, they start at the end of summer and continue until the winter time, or about the end of November.’

In the other 2 cases, the gap has the function_{PLACE}. In (2.45), where the speaker talks about tennis, the *no*-part seems to denote a place for playing tennis.

(2.45) Participant nominalisation, extracted from the CSJ (S04F1636)

[renshuu-suru no]-wa kashi-kouto-de ya-ttari koukyou-no-kouto-de
 [practice-do NO]-TOP rental-court-at do-or public-GEN-court-at
 ya-ttari...
 do-or
 ‘As for the place where we play tennis, we do it at a rental court or we do it at a public court...’

Finally, there are 2 unclear examples. In (2.46), the speaker talks about sound effects in concert halls. The embedded verb *ohanashi-suru*- (= ‘talk’) seems to be an intransitive verb since if the content of a talk is mentioned, it must be marked by the

postposition *nitsuite* (= ‘about’) as in (2.47). If this assumption is on the right track, the gap in (2.46) will be an adjunct, possibly bearing the function_{THEME}.

- (2.46) Participant nominalisation, extracted from the CSJ (A05M0058)

[korekara ohanashi-suru no]-wa keisan-wa subete
[from.now.on talk-do NO]-TOP calculation-TOP all
nijigen desu.
two.dimensions COP

‘As for what I am going to talk about, calculations are made on the two-dimension basis.’

- (2.47) Porco-wa kumo-no-heigen-nitsuite ohanashi-suru.

P-TOP cloud-GEN-field-about talk-do

‘Porco talks about the field of clouds.’

Another unclear case is (2.48). The speaker talks about her dog which had an official commendation for living long, and complains humorously about the certificate of merit given to the dog. In the pre-*no* clause, *monku* (= ‘complaint’) is an object of the verb *i-* (= ‘say’). This is illustrated in (2.49), where *monku* may be marked by the accusative-case particle *o*. Thus, it may appear that this pre-*no* clause lacks a gap, but (2.49) also shows that the complex *monku i-tta* may take a quoted element. My supposition is that the gap in (2.48) may correspond to the quoted element.

- (2.48) Participant nominalisation, extracted from the CSJ (S02M1698)

[sontoki monku i-tta no]-wa ‘nande kamikire-o’
[then complaintsay-PAST NO]-TOP why paper-ACC
‘inu-wa yo-me-nai yo’ to
dog-TOP read-can-NEG SFP QUOT

Lit. ‘As for what I complained about then, I said “Why did they give the dog a piece of paper?” and “The dog cannot read it!”.’

- (2.49) Watashi-wa ‘nande kamikire-o’ to monku(-o) i-tta.
 I-TOP ‘why paper-ACC’ QUOT complaint(-ACC) say-PAST
 ‘I complained by saying “Why did they give the dog a piece of paper?”.’

To sum up, the gap in participant nominalisation has a subject function or an object function in the majority of the CSJ examples, but it may also serve as an adjunct which bears various functions (e.g., function_{TIME}, function_{PLACE}). An adequate account of participant nominalisation must model that the denotation of the *no*-part may be based on an adjunct gap as well as an argument gap.

4.5 Long-Distance Dependencies and Islands

Since a gap is involved in participant nominalisation, it is essential to consider whether it exhibits a long-distance dependency. In the CSJ cases of participant nominalisation, there are no cases that manifest a long-distance dependency. Still, one cannot conclude that participant nominalisation never displays a long-distance dependency. For instance, consider the artificial example (2.50). The gap is the object of the most embedded verb *katte-ki-*, which is associated with *no* across a clause-boundary.

- (2.50) [Porco-ga [Fio-ni katte-ki-ta to] itta no]-wa oishii.
 [P-NOM [F-for buy-come-PAST COMP] said NO]-TOP tasty
 ‘As for what Porco said that he bought for Fio, it is tasty.’

Another issue is whether participant nominalisation is sensitive to syntactic ‘islands’ (Ross 1967, 1986). In the CSJ data, there are no island-involving examples. Yet, it is possible to create an artificial example involving islands, as in (2.51), where the speaker talks about Porco’s flying boat. Here, the gap is located within the complex NP headed by the noun *uwasa* (= ‘rumour’).

- (2.51) [[Porco-ga shuuri-shita toiu usawa]-ga atta no]-wa
 [[P-NOM repair-did TOIU rumour]-NOM existed NO]-TOP
 Folgore-o tousaishiteiru.
 F-ACC have.on.board
 ‘As for the flying boat x such that there was a rumour that Porco repaired x, it
 has the Folgore engine on board.’

The upshot is that though the 95 CSJ examples of participant nominalisation do not manifest long-distance dependencies or island-insensitivity, their behaviour with respect to these constructions is illustratable by artificial examples. Thus, they must not be structurally blocked when one proposes an account of *no*-nominalisation.

4.6 Summary

This section has reported corpus findings regarding *no*-nominalisation. The main results are condensed into the following bullet points:

- In both participant and situation nominalisation, the *no*-part is a nominal that denotes a definite or indefinite entity. In situation nominalisation, the type of entities is diverse, ranging from actions and events to propositions.
- In both types of *no*-nominalisation, the *no*-part may bear a core grammatical function or induce a pure topic.
- The gap in participant nominalisation may have a wide range of grammatical functions (including $\text{function}_{\text{TIME}}$ and $\text{function}_{\text{PLACE}}$).
- Participant nominalisation shows both long-distance dependencies and island-insensitivity, though these cases are not attested in the CSJ itself.

In Chap. 4, a dynamic account of *no*-nominalisation will be proposed. Each of the above generalisations will be accounted for in the following ways:

- The particle *no* takes the preceding clause as an input and outputs a content which denotes a definite or indefinite entity. This content may be a situation term, which may denote actions, events, propositions, etc.
- The particle *no* creates a type-e node, and this is an appropriate node position for parsing any case particles and the topic particle.
- The particle *no* encodes an instruction to copy a type-e element regardless of the grammatical function of the gap.
- The particle *no* encodes an instruction to copy a type-e element regardless of the structural position in which a gap occurs.

5 Cleft Constructions

5.1 Introduction

In the literature, Japanese clefts are classified into two sub-types depending on whether a case particle is attached to a focus item (Hoji 1990). In this thesis, clefts with a case-marked focus as in (2.52) are called clefts_{+C}, while clefts with a case-less focus as in (2.53) are called clefts_{-C}. In both cases, the particle *no* turns a preceding clause (often called ‘presuppositional clause’ (Kizu 2005: 3)) into a nominal, and this nominalised clause is topicalised by the particle *wa*. With this topic as context, the focus item *Curtis* is processed, and the string ends with the copula *da*-.⁸

(2.52) Clefts_{+C}

[Porco-ga kettou-o shita no]-wa Curtis-to da.
 [P-NOM duel-ACC did NO]-ACC C-with COP
 ‘It is with Curtis that Porco duelled.’

⁸ It seems that the copula *da*- does not express the tense of the string. So, even if it is in a past-tense form *da-tta*, the truth-conditional content of the string remains unchanged. This is true of the copula *be* in English clefts, as in *It is/was with Curtis that Porco duelled*. For relevant discussion, see Declerck (1988: 80-5) on English clefts and Tamura (2009) on Japanese clefts.

(2.53) Clefts_{-C}

[Porco-ga kettou-o shita no]-wa Curtis da.

[P-NOM duel-ACC did NO]-ACC C COP

‘It is Curtis that Porco duelled with.’

The aim of this section is to present spontaneous data on clefts and to make some generalisations that may be useful for a theoretical account of clefts. After previous studies are surveyed in Sect. 5.2, my corpus findings are reported through Sect. 5.3 to Sect. 5.7. Finally, theoretical implications of these data are pointed out in Sect. 5.8.

5.2 Previous Corpus Work

There are only a few previous studies of Japanese clefts that made use of a corpus. In general, a corpus is regarded as a digitalised, balanced, and annotated collection of naturally-occurring strings. But this sub-section uses the term ‘corpus’ simply to mean a collection of spontaneous language data. This is because not all previous studies to be reviewed fit well with the aforementioned definition of corpora; for instance, Sunakawa (2005) examines cleft strings that were manually gathered from written materials such as newspaper and magazines.

Hasebe (2007) extracted from Wikipedia the strings which conform to the ‘X-*no-wa* Y-*da*’ schema⁹ by using a morphological analyser, and then selected 500 cleft strings manually. Based on these data, Hasebe argues that the semantic structure of clefts in English and Japanese is uniformly characterised within Cognitive Grammar (Langacker 1987, 1991). Hasebe reveals interesting discourse functions of clefts, but his selection procedure ends up counting in non-cleft strings such as (2.54).

⁹ In Sect. 2.2, I stated that the schema ‘X-*no-wa* Y-*da*’ does not fully characterise clefts since (i) the copula *da-* may be omitted in spontaneous production and (ii) the copula *da-* may appear in the focus part Y if the focus part itself involves a clause. But this drawback does not apply to Hasebe’s (2007) work because it does not aim at extracting all cleft strings from the database.

- (2.54) [Shizengengo-ga suugakutekini teigisa-reru fuhen
 [natural.language-NOM mathematically define-PASS universal
 ruibetsu daisuukei de-nai no]-wa mochiron dearu.
 sorted algebra cop-NEG NO]-TOP natural COP
 ‘It is taken for granted that natural languages are not a universal sorted algebra
 to be mathematically defined.’ (Hasebe 2007: 183)

Although Hasebe (2007: 190-1) states that these data have similarities with clefts, their fundamental differences have also been pointed out in the literature (Declerck 1988). For instance, unlike clefts, data such as (2.54) do not have a ‘specificational’ function. In my corpus work, these data are labelled as ‘situation nominalisation’ and explicitly differentiated from clefts. Thus, it is difficult to compare Hasebe’s study with my work.

Kahraman et al. (2011b) present a complementary corpus work to Kahraman et al. (2011a), who provide experimental data that object-clefts (i.e., clefts with an object gap) are easier to process than subject-clefts (i.e., clefts with a subject gap). Kahraman et al. (2011b) seek a factor affecting this processing asymmetry by considering (i) the frequencies of non-clefts, subject-clefts, object-clefts, and other types of clefts and (ii) ‘transitional probabilities’ (i.e., probabilities of an expression Y given an expression X) of the sequence ‘[argument verb *no*]-*wa*.’ They utilised part (3 million words) of the KOTONOAHA corpus, a written corpus of 10 million Japanese words (developed by the National Institute for Japanese Language and Linguistics). Strings involving the sequence *no-wa* were extracted by ChaKi.NET (developed by the Nara Institute of Science and Technology), and they were manually classified into 656 subject-clefts, 170 object-clefts, 930 cases of other types of clefts, and 329 non-clefts. It is observed that (i) the frequency of subject-clefts is higher than that of object-clefts and that (ii) the transitional probability of subject-clefts is lower than that of object-clefts. These results indicate that the processing difficulty of subject-clefts is not a matter of frequency but due to a parser’s expectation for upcoming structure. This corpus work is illuminating in conjunction with Kahraman et al. (2011a), but the corpus finding itself is limited.

Sunakawa (1995, 2005) manually gathered data from written materials such as novels and magazines. Sunakawa's (2005) results are presented in Table 2.6, which is translated and adapted from Sunakawa (2005: 219).

Type	Occurrences	Percentage
noun	185	84%
noun + case	11	5%
embedded clause	21	10%
adverb	3	1%
Total	220	

Table 2.6. Type of the focus in clefts¹⁰

Sunakawa observes that there is a strong tendency that the argument case particles (i.e., nominative, accusative, dative) are absent from the focus. In all of 11 examples where the focus is case-marked, the particle is not an argument case particle, as shown in Table 2.7 (Sunakawa 2005: 221, my translation and inconsequential modifications).

Particle	Occurrences
<i>kara</i> (inception)	3
<i>made</i> (duration)	2
<i>kara</i> (reason)	3
<i>de</i> (location)	1
<i>nikakete</i> (temporal boundary)	1
<i>nioite</i> (situation)	1
Total	11

Table 2.7. Particles attached to the focus

In these examples, the use of case particles is obligatory since they are essential for retrieving the intended meaning of the string. For instance, consider (2.55), adapted and translated from Sunakawa (2005: 221).

¹⁰ In my classification, the category 'noun' and 'adverb' correspond to clefts_{-C}, and the other two categories correspond to clefts_{+C}.

- (2.55) [Koushita [chuuoushuukanka-sareta kokka]-ga shutsugenshi-ta
 [like.this [centralise-PASS state]-NOM emerge-PAST
 no]-wa 1870-nen-goro-kara da-tta.
 NO]-TOP 1870-year-around-since COP-PAST
 ‘It was since around 1870 that the centralised states like these emerged.’

If the particle *kara* is dropped, the resulting string means: ‘It was in around 1870 that the centralised states like these emerged.’ That is, the omission of the particle affects the propositional content of the string. Sunakawa claims that case particles are produced in clefts where without them the intended meaning would vanish.

On the other hand, Sunakawa notices the examples where argument case particles are attached to foci. At first sight, they seem to be incompatible with her claim. Sunakawa (2005: 225) analyses an example in Noda (1996: 72), cited here as (2.56) with my translation and modifications. The writer of this example is surprised at the price of a plane ticket from Rome to Tokyo, which is much lower than the price of a ticket from Tokyo to Rome.

- (2.56) [Odoroi-ta no]-wa sono-nedan-no-yasusa-ni desu.
 [be.surprised-PAST NO]-TOP that-price-GEN-cheapness-at COP
 Lit. ‘It is at that cheapness of the price that I was surprised.’

In (2.56), the particle *ni* is attached to the focus, though the intended meaning could be retrieved without the particle. Sunakawa (2005: 227) points out that the topic of the discourse following this sentence is about the cheapness of plane tickets bought in Europe, and claims that the use of the particle *ni* in (2.56) emphasises the content of the focus item as a topic of the following discourse. Her generalisation, then, is that the particle is attached to a focus item if the speaker intends a special discourse function such as emphasis and contrastiveness.

On this evidence, Sunakawa (2005: 229) puts forward (2.57). According to this conjecture, particle dropping is governed by discourse (but not structural) factors. This

indicates that a grammar model of clefts must be neutral to clefts_{+C} and clefts_{-C} and that it should not distinguish between them structurally.

(2.57) Sunakawa's (2005) conjecture

Particle dropping is motivated by discourse factors. A particle is dropped off a focus unless (i) it is required for retrieving the intended meaning or (ii) the writer adds a special meaning to the proposition expressed.

While Sunakawa's contribution to the study of clefts is a useful stepping stone, her conjecture remains vacuous in the present form since she does not address any syntactic issues that have motivated the clefts_{+C}/clefts_{-C} divide (e.g., island-sensitivity). Without considering these syntactic issues, her claim that the distinction is not a syntactic one remains virtually untestable. In Chap. 5, I shall develop an account of clefts which is neutral to clefts_{+C} and clefts_{-C} but which predicts their structural differences as a result of how the content of a focus item is incorporated into the propositional structure during the course of incremental processing.

5.3 Particle Dropping

From this sub-section on, I shall present various CSJ findings of clefts, starting with particle dropping. The proportion of clefts_{+C} and clefts_{-C} is summarised in Table 2.8. The result is strikingly similar to the one in Sunakawa (2005). That is, there is a strong tendency to drop case particles off the focus expressions in a spontaneous setting.

Type	Occurrences	Percentage
clefts _{+C}	25	8%
clefts _{-C}	297	92%
Total	322	

Table 2.8. Particle dropping in clefts in the CSJ

There are 297 instances of clefts_{-C}. For instance, consider (2.58), where the speaker talks about a polyline in a graph in his presentation. The focus *michigo-ritsu* (=

‘the rate of unknown words’) lacks a case particle, which could potentially have the accusative-case particle *o*.

(2.58) Clefts, extracted from the CSJ (R00M0134)

[oresen-de shimeshita no]-wa michigo-ritsu desu...
 [polyline-with showed NO]-TOP unknown.word-rate COP
 ‘It is the rate of unknown words that I showed with the polyline...’

It should be noted, however, that out of 297 instances, there are 10 examples where the absence of a particle is compulsory. In (2.59), the focus item *mikkakan* (= ‘for three days’) cannot accommodate any particle in the first place.

(2.59) Clefts, extracted from the CSJ (S01F0217)

[Nishiomotejima-ni i-ta no]-wa jissai mikkakan nandesu...
 [N-in stay-PAST NO]-TOP actually for.three.days COP
 ‘It is actually for three days that I was in Nishiomotejima...’

Apart from these obligatorily particle-less examples, a particle tends to be dropped off a focus in spontaneous speech.

Next, there are only 25 instances of clefts_{+C}. In these cases, a particle is present because (i) particle dropping is not possible in the first place or (ii) the use of the particle provides additional meaning. As for the point (i), there are 20 examples where a particle is compulsorily present. In all of these 20 examples, the particle attached to the focus is *kara*, which may serve either as the connective ‘because’ (11 examples) or the temporal connective ‘since’ (9 examples). An example where *kara* is translated into ‘because’ is (2.60). The particle *kara* cannot be omitted because it has a nominalising function, turning the preceding clause into a nominal. This issue is not discussed in Sunakawa (2005), but her conjecture (2.57) remains intact if we assume a structural condition that the focus must be a nominal or a nominalised element.

(2.60) Clefts, extracted from the CSJ (S03F0133)

[Nishinipori-ga hattenshi-ta no]-wa [kawa-no-soba
[Nishinipori-NOM develop-PAST NO]-TOP [river-GEN-side
ja-naka-tta]-kara da...
COP-NEG-PAST]-because COP

‘It is [because it was not close to a river] that Nishinipori developed...’

An example where *kara* is translated into ‘since’ is (2.61), where the speaker tells when she got interested in delivering a speech. As in the last example, the particle *kara* has a nominalising function.

(2.61) Clefts, extracted from the CSJ (S00F0208)

[supiichi-ni kyoumi-o motta no]-wa [tsuuyaku-no-benkyou-o
[speech-at interst-ACC had NO]-TOP [translation-GEN-study-ACC
hajimeru-youni-natte]-kara desu...
start-like-become]-since COP

‘It is since I started to study about translation that I got interested in speech...’

As for the point (ii) mentioned at the start of this paragraph, there are 5 cases where a particle is optionally present. In these cases, it seems the speaker tries to achieve an additional meaning by explicitly producing a case particle. For instance, consider (2.62), where the nominative-case particle *ga* is attached to the focus *hitobito-ippa*n (= ‘people in general’). The speaker talks about a language processing study, stating that his processing model correctly identifies the case-marking of a zero pronoun. In (2.62), he says that in the string ‘X must be able to do communication properly’ with a zero pronoun X, the processing model outputs the case-marked NP *hitobito-ippa*n-*ga*. In this context, the speaker stresses that the model identifies the *ga*-marking accurately, and this emphasis is effectively achieved by producing the particle *ga*. (The same speaker produces the same type of cleft strings with the *ga*-marked focus once again in his talk.)

(2.62) Clefts, extracted from the CSJ (A03M0032)

[kichinto deki-nakerebanaranai no]-wa hitobito-ippa-da...
 [properly can-must NO]-TOP people-general-NOM COP
 ‘It is “people in general” that must be able to do communication properly...’

These speech data lend support to Sunakawa’s hypothesis (2.57) which is based on the discourse data. The hypothesis (2.57) may thus be generalised as (2.63) by replacing ‘discourse factors’ with ‘pragmatic factors’ (including both discourse and dialogue factors) and by replacing ‘the writer’ with ‘the producer.’ As stated around (2.60), this pragmatic constraint is supplemented by the structural condition that the focus must be a nominal or a nominalised element.

(2.63) Pragmatic constraint on particle dropping (Refining Sunakawa (2005: 229))

Particle dropping is motivated by pragmatic factors. A particle is dropped off a focus item unless (i) it is required for retrieving the intended meaning or (ii) the producer adds a special meaning to the proposition expressed.

5.4 Category of Foci

The last sub-section showed that foci must be a nominal. As long as this requirement is satisfied, the syntactic category of foci is diverse, as presented in Table 2.9.

Category	Occurrences (clefts _{+C})	Occurrences (clefts _{-C})
NP	8	261
Adv-NP	0	23
Clausal-Adv-NP	16	1
Dem	0	1
WhP	1	11
Total	25	297

Table 2.9. Category of a focus in clefts in the CSJ

Examples of the first three categories have been provided in the last sub-section. An example of ‘NP’ is (2.58), that of ‘Adv(erbale)-NP’ is (2.59), and that of ‘Clausal-Adv(erval)-NP’ is (2.60). An example of ‘Dem(onstrative)’ is offered here as (2.64).

The speaker talks about a river running in his living place, stating that the place is notorious for the river since it is quite stagnant. In this example, the focus position is occupied by the demonstrative *sore* (= ‘that’).

(2.64) Clefts, extracted from the CSJ (S03M0996)

[warui-hou-no yuumeina no]-wa sore desu-ga...
 [bad-side-GEN famous NO]-TOP that COP-but
 ‘It is that thing (= the river) that is notorious, but...’

An example of ‘WhP’ is given in (2.65), where the speaker talks about ruins of the Yayoi period.

(2.65) Clefts, extracted from the CSJ (A01M0077)

[iseki-ga ippai aru no]-wa doko darou...
 [ruin-NOM many exist NO]-TOP where COP
 ‘Where it is that there are many ruins...’

In this example, the *wh*-word *doko* (= ‘where’) is used, but other types of *wh*-words may also be a focus in clefts, as shown in Table 2.10.

<i>Wh</i> -words	Occurrences (clefts _{+C})	Occurrences (clefts _{-C})
<i>nani</i> (= ‘what’)	0	2
<i>dore</i> (= ‘which’)	0	1
<i>dare</i> (= ‘who’)	0	3
<i>doko</i> (= ‘where’)	0	1
<i>itsu</i> (= ‘when’)	1	1
<i>naze</i> (= ‘why’)	0	3
Total	1	11

Table 2.10. Type of *wh*-word in clefts in the CSJ

An account of clefts must predict that clefting may take place regardless of the type of syntactic category of a focus as long as the focus is a nominal or a nominalised element.

5.5 Grammatical Function of Foci

The last sub-section pointed out that the syntactic category of a focus in clefts is quite diverse. Table 2.11 shows that the diversity is also found in the type of grammatical function of a focus.

Type	Occurrences (clefts _{+C})	Occurrences (clefts _{-C})
subject	3	138
object	0	74
indirect object	0	3
function _{PLACE}	0	14
function _{TIME}	9	33
function _{REASON}	11	7
function _{THEME}	2	0
unclear	0	28
Total	25	297

Table 2.11. Grammatical function of a focus in clefts in the CSJ

An example of ‘subject’ is (2.64), an example of ‘object’ is (2.58), an example of ‘function_{PLACE}’ is (2.65), an example of ‘function_{TIME}’ is (2.61), and an example of ‘function_{REASON}’ is (2.60). A possible example of ‘indirect object’ is presented here as (2.66), where the speaker talks about his pet, a dog. Literally, *aiso* (= ‘amiability’) is a direct object of *furu-* (= ‘wave’), though the accusative-case particle *o* is omitted. What is meant by this string is that the dog gives amiability only to the grandparents of the speaker. Then, the focus could be seen as having the indirect object function.

(2.66) Clefts, extracted from the CSJ (S02M0198)

[Kare-ga aiso furu no]-wa sodesune [esa-o ageru
[he-NOM amiabilitywave NO]-TOP well [food-ACC give
ojiichan-obaachan] nomi...
grandpa-grandma] only
‘It is, well, only my grandparents that he is tamed for...’

An example of what may be viewed as ‘function_{THEME}’ is (2.67), where the focus is marked by the particle *nitsuite* (= ‘about’). The speaker talks about his comparative

lexicon research, and stresses that his research is about not the lexicon of a language but the lexicon of a specific literary work. (The speaker calls the lexicon of a specific literary work *kobetsu-goi* (= ‘individual lexicon’).)

(2.67) Clefts, extracted from the CSJ (A02M0117)

[sono-shiyō-ritsu-o atsukau no]-wa shutoshite

[that-use-rate-ACC handle NO]-TOP mainly

kono-kobetsu-goi-nitsuite dearimasu...

this-individual-lexicon-about COP

Lit. ‘It is mainly about this individual lexicon that I handle the use rate...’

Finally, there are 28 examples where the grammatical function of the focus cannot be clearly determined because the valency of the verb is not immediately obvious. For instance, consider (2.68), where the verb in the pre-*no* clause is *iku-* (= ‘go’). If *iku-* is construed as an intransitive verb, the focus is an adjunct and its grammatical function may be viewed as function_{PLACE}. On the other hand, if *iku-* is assumed to be a transitive verb, the focus may be taken as bearing the object function. Given that the verb *go* in English is usually considered to be intransitive, the former line of analysis is more plausible. Still, I shall set these cases aside to be on the safe side.

(2.68) Clefts, extracted from the CSJ (S00F0193)

[watashi-ga yoku iku no]-wa [Harajuku-ni aru Togo-jinja

[I-NOM often go NO]-TOP [H-in exist T-shrine

toiu toko] nandesu...

TOIU place] COP

‘It is the Togo shrine, located in Harajuku, that I often go to...’

As demonstrated in Table 2.11, if a focus bears a core grammatical function such as subject and object, it is unlikely to be case-marked (cf., Sunakawa 2005). A similar observation has also been made in the literature on the syntax of clefts. Hiraiwa & Ishihara (2012), among others, state that clefts with an accusative-case marked focus

are degraded for some speakers and that clefts with a nominative-case marked focus is degraded for many speakers. This remark, however, should not be taken as indicating that clefts with such foci are ungrammatical. Cho et al. (2008) report that clefts with a nominative-case marked focus are attested in the web. Further, Koizumi (2000) notes that clefts with a nominative-case or accusative-case marked focus are acceptable when a numeral quantifier is added as in (2.69) or when there is more than one focus as in (2.70). (These types of example were not found in the CSJ.)

- (2.69) [Porco-ga kanashim-ase-ta no]-wa josei-o futa-ri da.
 [P-NOM feel.sad-CAUSE-PAST NO]-TOP lady-ACC 2-CL COP
 Lit. 'It is two ladies x that Porco made x feel sad.'

- (2.70) [Kisu-o-shita no]-wa Fio-ga Porco-ni da.
 [kiss-ACC-did NO]-TOP F-NOM P-DAT COP
 Lit. 'It is Fio_i Porco_j that e_i kissed e_j.'

In sum, the syntactic category of a focus ranges from core functions such as subject and object to function_{PLACE} and function_{TIME}, the tendency being that a particle is absent from the focus bearing a core grammatical function. Still, case-marking of a focus with a core grammatical function is possible and indeed fully acceptable if a numeral quantifier is used or if there are multiple foci. Thus, an account of clefts should not preclude case-marked foci structurally even if they bear core grammatical functions.¹¹

¹¹ The case-dropping tendency seems to be stronger in Korean. Kang (2006: 254-5) reports that in Korean, clefts are completely unacceptable if a focus has a nominative-case particle or an accusative-case particle.

(i) [John-ul manna-n kes]-un Mary(*-ga)-i-tta. [Korean]
 [J-ACC meet-PRE KES]-TOP M(-NOM)-COP-DECL
 'It was Mary that met John.'

5.6 Information Status of Foci

The focus part in clefts usually conveys new information, but it has been pointed out that in English clefts, the focus may present old information (Declerck 1984, 1988, Prince 1978). This type of clefts is called ‘unstressed-anaphoric-focus clefts’ (Declerck 1984: 265). In (2.71), the focus *that evidence* imparts old information in that it is anaphoric to *interesting independent evidence* in the preceding string.

- (2.71) However, it turns out that there is interesting independent evidence for this rule and it is to that evidence that we must now turn. (Declerck 1984: 263)

Kumamoto (1989) points out that this also holds of Japanese clefts as in (2.72), which is translated and adapted from Kumamoto (1989: 26). The focus item *kono-tame* (= ‘for this reason’) is anaphoric to the preceding discourse (omitted in (2.72)): ‘In Europe and America, where dismissals are conducted relatively easily, each company tries to stabilise its revenue by reducing the number of workers rather than by stabilising the amount of products.’

- (2.72) [Fukyou-ki-ni Nihon-de-wa shutoshite kigyuu-keiei-no-
[recession-period-at Japan-in-TOP mainly business-management-GEN-
akka-ga mondai tonari, oubei-de-wa moppara
deterioration-NOM problem become Europe.America-in-TOP mainly
shitsugyousha-no-zouka-ga chuumokusareru no]-wa
unemployed.persons-GEN-increase-NOM be.watched NO]-TOP
kono-tame dearu.
this-for COP

‘It is for this reason that in the recession period, in Japan, the problem is mainly about the deterioration of business management, while in Europe and America, attention is mainly paid to the increase of unemployed persons.’

I investigated whether the CSJ also contains this type of clefts. Since it is not easy to measure the information status of a focus objectively, I made use of demonstratives,

which are said to have both anaphoric and deictic usages (Kuno 1973a: 282). The procedure is in three steps. First, 55 cases of clefts (6 clefts_{+C} and 49 clefts_{-C}) were extracted where the focus contains demonstratives. Second, they were narrowed down to 14 cases (2 clefts_{+C} and 12 clefts_{-C}) where a demonstrative has an anaphoric usage (rather than a deictic usage). Finally, I checked each of these 14 examples and selected 7 cases (2 clefts_{+C} and 5 clefts_{-C}) where the focus clearly conveys old information. These are summarised in Table 2.12.

Procedure	Occurrences (clefts _{+C})	Occurrences (clefts _{-C})	Total
Step 1	6	49	55
Step 2	2	12	14
Step 3	2	5	7

Table 2.12. Procedures to detect the information status of a focus in the CSJ

A relevant example is (2.64), repeated here as (2.73). The speaker talks about his living place, and the demonstrative *sore* (= ‘that’) at a focus position is anaphoric to the name of a river which is mentioned in the preceding discourse (omitted in (2.73)).

(2.73) Clefts, extracted from the CSJ (S03M0996)

[warui-hou-no yuumeina no]-wa sore desu-ga...

[bad-side-GEN famous NO]-TOP that COP-but

‘It is that thing (= the river) that is notorious, but...’

In this way, there are at least 7 examples where the focus conveys old information. They have a theoretical implication that an account of clefts must not encode any notion of new or old information in characterising the focus part.

5.7 Long-Distance Dependencies and Islands

Finally, let us consider the structural relation between a gap and a focus. All examples cited so far are ‘short-distance’ clefts, where a focus is associated with a gap in the main clause within the pre-*no* part. By contrast, there are 8 cases of ‘long-distance’ clefts (7

clefts_{-C} and 1 cleft_{+C}). For instance, in (2.74), the focus *terebi* (= ‘TV’) is associated with the gap of the most embedded predicate *chigau-* (= ‘different’).

(2.74) Clefts, extracted from the CSJ (S00M0213)

[[Eikoku-to-Nihon-to ichiban chigau-na to] omo-tta no]-wa
 [[UK-and-Japan-and most different-CFP COMP] think-PAST NO]-TOP
 terebi deshi-te...

TV COP-and

‘It is TV programmes that I thought were the most different between the UK and Japan, and...’

In addition to these 8 examples, there is one potential instance of long-distance clefts; in (2.75), the *no*-part has clausal embedding. But it is contextually clear that (2.75) is not an example of long-distance cleft; the *kara*-part is associated with a gap of the predicate *iwareru-* (= ‘is said’), not the most embedded predicate *yoku-* (= ‘good’).

(2.75) Clefts, extracted from the CSJ (S03F0087)

[[yoku-nai-ne to] iwareru no]-wa osoraku [saikin tokuni
 [[good-NEG-SFPCOMP] is.said NO]-TOP probably [recently particularly
 warui-imeeji-ga Ikebukuro-ni tsuiteiru]-kara da...

bad-image-NOM I-to attach]-because COP

‘It is probably because recently in particular, a bad image has been attached to Ikebukuro that the city is said to be not good...’

There is a single example that may be seen as an island-involving case. In (2.76), the focus *Asakusa* could be interpreted within the adjunct clause *tabete*.¹²

¹² Another interpretation is that the focus *Asakusa* corresponds to an adjunct gap of the predicate *tanoshii-* (= ‘enjoyable’). In this case, the string is not an island-involving cleft, with the reading ‘It is after all Asakusa_i that people have fun in e_i when they eat.’ In the reading given in (2.76), the place where the speaker feels fun does not have to be Asakusa; for instance, he/she could feel fun at home by remembering eating in Asakusa. It is not obvious whether this reading is intended in (2.76). But even if not, it is possible to construe (2.76) in this way, in which case it is not island-sensitive. See Sect. 5, Chap. 5 for artificial examples of island-involving clefts.

(2.76) Clefts, extracted from the CSJ (S03F0087)

[[tabete] tanoshii no]-wa yappari Asakusa desu...

[[eat] enjoyable NO]-TOP after.all Asakusa COP

Lit. 'It is after all Asakusa_i that people have fun when they eat in e_i...'.

The distribution of 'short-distance,' 'long-distance,' and 'island-involving' examples is presented in Table 2.13. Though long-distance clefts and island-involving clefts are not abundant in the CSJ, they exist in spontaneous speech. Therefore, an account of clefts should not rule them out structurally.

Type	Occurrences (clefts _{+C})	Occurrences (clefts _{-C})
short-distance	24	289
long-distance	1	7
island-involving	0	1
Total	25	297

Table 2.13. Long-distance and island-involving cases of clefts in the CSJ

5.8 Summary

This section has reported the results of my corpus work on Japanese clefts, which have uncovered the following points:

- Since particle omission is due to pragmatic factors, an analysis of clefts must be neutral to the two types of clefts. On the other hand, it must also account for structural differences (e.g., island-sensitivity).
- A focus may have a range of syntactic categories and grammatical functions, as long as it is a nominal or nominalised. An account of clefts must predict both the category/function-flexibility of foci and the nominal status of foci.
- A focus may convey either new or old information. Thus, an account of clefts must be neutral to the type of information status of foci.
- Long-distance clefts and island-involving clefts are possible. Thus, an account of clefts should not prevent them structurally.

In Chap. 5, a dynamic account of clefts will be proposed. The main tenet is that the entry of *no* (to be proposed for *no*-nominalisation in Chap. 4) and the entry of the topic marker *wa* are combined to bring force the new entry of *no-wa*. This new entry meets the above four theoretical requirements in the following ways:

- In both clefts_{+C} and clefts_{-C}, the same entry of *no-wa* is operative, with their structural differences emerging as an outcome of how the content of a focus is incorporated into the main propositional structure.
- The entry of *no-wa* requires that the focus should be of type-e, but does not care about its syntactic category or its grammatical function.
- The entry of *no-wa* does not encode any information status of foci.
- The entry of *no-wa* does not pose any structural condition on where the gap is to be located.

6 Theoretical Significances

Japanese raises a challenge to the modelling of incremental language understanding. The language is head-final, with all arguments being freely permuted, dropped, and underspecified for definiteness and plurality. It is not always morphologically signalled whether a clause is matrix or subordinate. These characteristics of the language will be accommodated by a grammar to be developed in Chap. 3.

In addition to the basic properties of the language, spontaneous data impose further conditions on an account of *no*-nominalisation and clefts.¹³ The *no*-part in *no*-nominalisation is a nominal which is underspecified for definiteness, plurality, and the type of denotation (e.g., object, action, event, proposition). The *no*-part is also neutral

¹³ These corpus findings do not include data on change relatives and head-internal relatives, a topic addressed in Chap. 6. This is partly because their occurrences are known to be quite low compared with other types of relatives (e.g., head-external relatives). It is also because my CSJ search is based on the procedures to specify the *no-wa* sequence; Ito (1986: 127) points out that in head-internal relatives, the *no*-part is generally not marked by the topic particle *wa*.

for grammatical function which it may bear in the containing clause; in participant nominalisation, the uncertainty of grammatical function is observed in the gap as well. Moreover, participant nominalisation exhibits long-distance dependencies and island-insensitivity. These data will shape an account of *no*-nominalisation in Chap. 4.

Particle dropping in clefts is determined pragmatically, and thus the two types of clefts (clefts_{+C} and clefts_{-C}), though they have apparently structural differences, should not be structurally distinguished. A focus may be associated with a gap in the pre-*no* clause across a clause-boundary or an island-boundary, and the focus may bear a variety of syntactic categories and grammatical functions as long as it is a nominal or a nominalised element. A focussed item is, by definition, a focus of the cleft string, but it may impart not only new information but also old information. A dynamic account capturing these characteristics of clefts will be articulated in Chap. 5.

This empirical grounding will be coupled with several artificial data in later chapters. Before going into an analysis of *no*-involving constructions, what I shall aim to achieve in the next chapter is to provide a theoretical grounding of the thesis: a grammar model reflecting language dynamics.

Chapter 3

Theoretical Scaffolding

1 Dynamic Syntax (DS)

The theme running through this thesis is the statement that ‘language dynamics is a significant concept in linguistic theorising’ (Chap. 1). The concept of ‘dynamics’ that I shall be invoking involves the direct reflection of incremental processing in real time: the parser processes a string word-by-word and updates an interpretation progressively. It was initially assumed that in verb-final languages such as Japanese, no projection of structure was possible until the verb was parsed (Pritchett 1992). But it is now granted that the parser builds a structure as a string is unfolded (e.g., Kahraman 2011, Kamide 2006, Kiaer in press, Miyamoto 2008). The dynamic nature of incremental parsing is modelled internally to the grammar within the framework of Dynamic Syntax (DS) as progressive growth of semantic representation (Cann et al. 2005, Kempson et al. 2001, Kempson et al. 2011).¹

¹ As emphasis is put upon parsing, the reader may wonder whether DS ignores the other side of language use: production. Cann et al. (2007) and Purver et al. (2006) argue that production is also modelled by the same machinery. See also Eshghi et al. (2011) and references therein. This neutral stance with regard to production and comprehension is supported by dialogue data, where it is frequently observed that a speaker and a hearer switch their role during conversation (Pickering & Garrod 2004). Howes (2012) shows that this cross-speaker compound contribution can be naturally modelled within the DS framework. See also Gregoromichelaki et al. (2011).

In this chapter, I shall develop a DS grammar of Japanese, following Cann et al. (2005), Kempson et al. (2001), and Kempson et al. (2011), but making some amendments to the formalism based on Seraku (2013b, in press a).

DS is a grammar formalism that models knowledge of language. Thus, DS is a theory of competence and is regarded as generative grammar in the sense of Chomsky (1965). Unlike other frameworks, however, competence is defined as a set of constraints on performance, more specifically, the gradual construction of interpretation in context. With such constraints, the parser processes a string left-to-right and builds a semantic structure incrementally, **without** a separate level of syntactic representation: ‘syntax’ is no more than a set of procedures to update a semantic tree progressively in context.

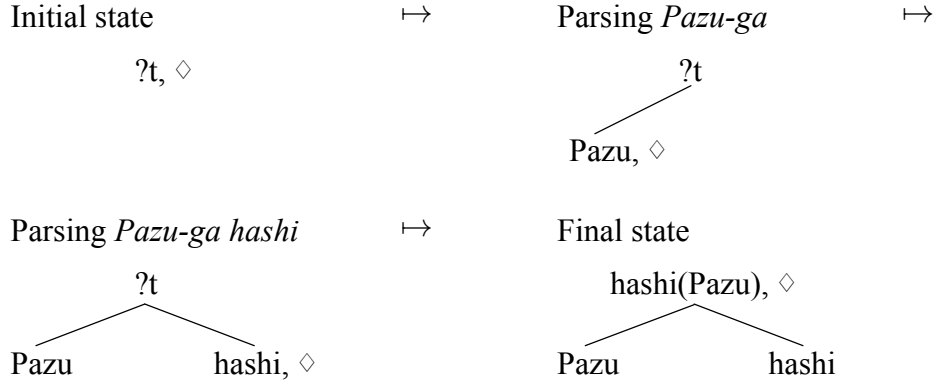
To illustrate the point, consider the string (3.1). As this string is parsed left-to-right, a semantic tree is gradually built up, as in (3.2) (ignoring the tense encoded in *tta*). The initial state sets out a node decorated with ?t, indicating that this node will be inhabited by a type-t (i.e., propositional) content. In every state of structure building, there is always a single node that is **under development**, and such an active node is marked by a pointer $\diamond$. It is worth stressing once again that the DS tree is a **semantic** representation: an interpretation is accumulated as (3.1) is parsed, and the final state of the tree update represents the interpretation of the string: ‘Pazu ran.’

(3.1) Pazu-ga hashi-tta.

P-NOM run-PAST

‘Pazu ran.’

(3.2) Tree updates for parsing (3.1) incrementally



The tree transitions in (3.2) are considerably simplified for illustration purposes. To spell out such tree transitions, Sect. 2 defines the declarative aspect of the DS formalism (i.e., what DS trees are and how they are decorated) and Sect. 3 defines the procedural aspect (i.e., how DS trees are generated). These are illustrated by sample tree transitions in Sect. 4. The basic machinery is enriched with the LINK device in Sect. 5 and with the DS notion of context in Sect. 6. Finally, Sect. 7 clarifies central DS assumptions by situating the DS formalism in the context of contemporary linguistic theories.

2 The Declarative Part

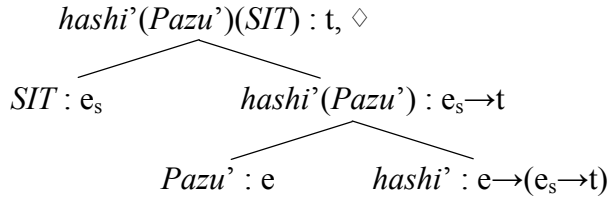
2.1 Trees and Tree Descriptions

DS trees are semantic representations, to be distinguished from syntactic structure. Standardly, syntactic structure, at least at the ‘surface’ representation involving the word-string itself, expresses (i) the linear sequence of words, (ii) the syntactic category of each node, and (iii) the hierarchy in which the nodes are syntactically combined. By contrast, DS semantic structure (i) does not reflect linear order, (ii) decorates nodes with content and type, and (iii) displays the hierarchy whereby meaning composition and type deduction take place. Moreover, there is no independent level of syntactic structure in the sense indicated above, that of a surface representation inhabited by the words in

sequence. The system is mono-level (phonology apart), a point I return to in Sect. 7, where the DS architecture is compared and contrasted with other grammar formalisms.

DS trees are always **binary-branching**. By convention, an argument node is on the left and a functor node is on the right. (This is not a reflection of any surface ordering.) Each node, if fully annotated, is decorated with a pair $\alpha : \beta$, where α is a **formula** (i.e., semantic content) and β **labels** (i.e., properties of the content). A formula is expressed in italics with an apostrophe, as in *Pazu'*, echoing the convention of Montague (1973). Labels include various types of decorations. An example of a label is a logical type, indicating the combinatorial potential of the content. Thus, the final state in the tree update (3.2) is to be represented as (3.3).

(3.3) Parsing *Pazu-ga hashi-tta*

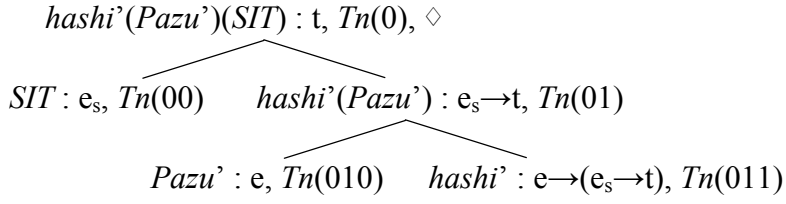


Here, *SIT* is a formula denoting a situation, and e_s is a logical type of situation terms. Building on the event semantics tradition (Davidson 1967, Higginbotham 1985, Parsons 1990), Gregoromichelaki (2006, 2011) assumes in DS that all verbs select a situation term as an argument.² The situation type e_s is sortally different from the individual type e_i under the more general, entity type e . For convenience, however, the individual type e_i is notated simply as e , as in (3.3). Within DS, the set of logical types is finite, and no operations to generate types (e.g., type-lifting) are stipulated; see the appendix for the list of logical types used in this thesis.

² Kratzer (1995), among others, claims that event terms are involved only in stage-level predicates, whereas Higginbotham (2000), among others, assumes that every predicate selects an event term, together with the assumption that event terms denote not only events but states, situations, etc. (Ikawa 2012). Event terms are cast as situation terms within DS; see also Kim (2009) and Kratzer (2011).

Another example of a label is a node identifier, $Tn(\alpha)$, where Tn is a tree-node predicate. If a node is annotated with $Tn(\alpha)$, $Tn(\alpha 0)$ indicates its argument daughter and $Tn(\alpha 1)$ its functor daughter. A root node is marked by $Tn(0)$, and from this it follows that the argument daughter is marked by $Tn(00)$ and the functor daughter by $Tn(01)$. With this addition of node identifiers, (3.3) is detailed as (3.4).

(3.4) Parsing *Pazu-ga hashi-tta*



Labels also include a decoration using the Logic Of Finite Trees (LOFT) (Blackburn and Meyer-Viol 1994). LOFT is a language to talk about trees, describing other nodes from the perspective of a current node. LOFT-operators are defined as follows. First, there are operators to model **immediate dominance**, with $\langle \downarrow_0 \rangle$ for argument daughters and with $\langle \downarrow_1 \rangle$ for functor daughters. For instance, $\langle \downarrow_0 \rangle(e)$ indicates that the argument daughter of the node so characterised is of type- e . The inverses, $\langle \uparrow_0 \rangle$ and $\langle \uparrow_1 \rangle$, describe a mother node from the perspective of the argument daughter node and the functor daughter node. For instance, the LOFT decorations in (3.5) hold of the top node in the tree (3.4).

(3.5) $\langle \downarrow_0 \rangle(\text{SIT}), \langle \downarrow_0 \rangle(e_s), \langle \downarrow_0 \rangle(Tn(00))$

Second, operators with the Kleene star $*$ model **dominance** in general, with $\langle \downarrow_* \rangle$ for ‘dominate’ and $\langle \uparrow_* \rangle$ for ‘be dominated by.’³ Since ‘immediately dominate’ is a special case of ‘dominate,’ the LOFT decorations (3.6) also hold of the top node in (3.4).

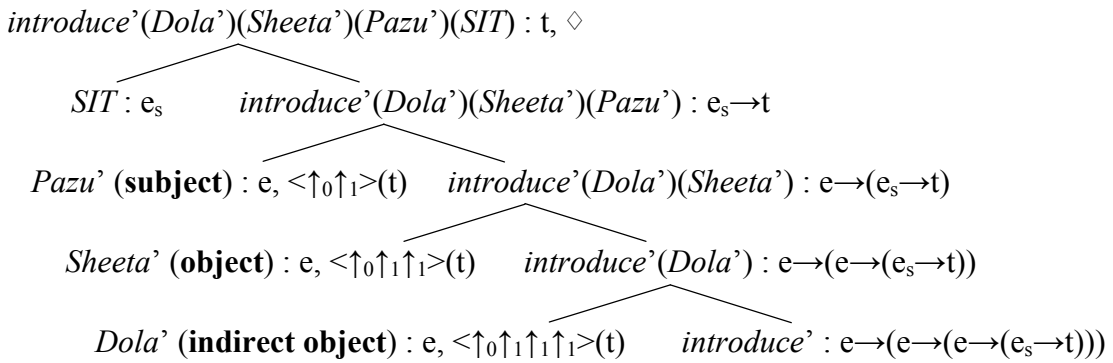
³ The Kleene star operator is a disjunction of the set of instances of a given relation, recursively reiterated, hence also ranges over the empty set. Formally, $\langle \downarrow_* \rangle(\alpha)$ is defined as $\alpha \vee \langle \downarrow_* \rangle(\alpha)$, and $\langle \uparrow_* \rangle(\alpha)$ is defined as $\alpha \vee \langle \uparrow_* \rangle(\alpha)$.

$$(3.6) \quad \langle \downarrow_* \rangle (SIT), \langle \downarrow_* \rangle (e_s), \langle \downarrow_* \rangle (Tn(00))$$

The operators $\langle \downarrow_* \rangle$ and $\langle \uparrow_* \rangle$ are used to analyse ‘long-distance dependencies’ (cf., Chomsky 1977) by inducing an unfixed node which can be resolved across an arbitrary distance within a single tree (Sect. 3.3). Finally, the ‘down’ operator $\langle D \rangle$ and the ‘up’ operator $\langle U \rangle$ model the **weakest** relation, and may describe a node within another tree across an adjunct or an island structure boundary (Sect. 5).⁴

Since LOFT describes a node relation in a tree, it may be utilised to specify grammatical functions. In DS, there are **no** primitive concepts of grammatical function; they are determined as an epiphenomenon, being characterisable directly through the LOFT notation. ‘Subject’ is assigned to the content at the $\langle \uparrow_0 \uparrow_1 \rangle (t)$ -node, ‘object’ to the content at the $\langle \uparrow_0 \uparrow_1 \uparrow_1 \rangle (t)$ -node, and ‘indirect object’ to the content at the $\langle \uparrow_0 \uparrow_1 \uparrow_1 \uparrow_1 \rangle (t)$ -node. This is illustrated by the tree generated by the simple sentence with a ditransitive verb *Pazu introduced Sheeta to Dola*, as in (3.7). I shall occasionally use terms such as ‘subject node’ and ‘subject argument,’ but the usage of these terms is for convenience.

(3.7) Parsing *Pazu introduced Sheeta to Dola*



All argument nodes local to the predicate *introduce'* are characterisable as $\langle \uparrow_0 \uparrow_* \rangle (t)$. Given the use of the Kleene star $*$, $\langle \uparrow_0 \uparrow_1 \uparrow_* \rangle (t)$ may be $\langle \uparrow_0 \uparrow_1 \rangle (t)$ (subject), $\langle \uparrow_0 \uparrow_1 \uparrow_1 \rangle (t)$ (object), and $\langle \uparrow_0 \uparrow_1 \uparrow_1 \uparrow_1 \rangle (t)$ (indirect object), an argument somewhere along the functor

⁴ Other LOFT operators will be introduced in due course. See the appendix for the list of LOFT operators used in this thesis.

spine in (3.7). As we shall see in Sect. 3.3, this LOFT specification of argument nodes is essential in defining ‘locally unfixed nodes’ and its resolution process.

2.2 Epsilon Calculus

As mentioned above, a node, if fully developed, consists of the pair of a formula and labels. So far, formulae of proper names have been considered. In the case of proper names, their content is notated simply by attaching an apostrophe to the name, as in *Pazu*'. But in the case of quantified expressions, a more elaborated representation is required. Within DS, every quantified noun is uniformly mapped onto a type-e term in the epsilon calculus, which involves the quantificational force within the term, unlike Montague (1973).⁵ This sub-section introduces the epsilon calculus as a logical underpinning of the DS account of quantification.

The epsilon calculus is a formal study of arbitrary names in natural deduction proofs in the first-order predicate logic (Hilbert & Bernays 1939), and the usefulness of epsilon calculus applications for linguistics has been examined in von Heusinger & Egli (2000), Kempson et al. (2001), and others. In natural language, quantified expressions occur in all the same environments as other NPs do, whereas in the predicate logic, quantifying expressions are operators directly preceding an open propositional formula. For example, the string *A pigeon flies* is mapped onto $\exists x.pigeon'(x) \& fly'(x)$, setting aside the tense. Here, $\exists$ is a quantifying operator and $pigeon'(x) \& fly'(x)$ forms an open propositional formula. In particular, note that the quantified expression *a pigeon* has no direct counterpart in the logical representation $\exists x.pigeon'(x) \& fly'(x)$. By contrast, in the epsilon calculus, every quantified NP is **uniformly** mapped onto a **type-e** term, called an ‘epsilon term,’ defined as a triple: (i) operator, (ii) variable, and (iii) restrictor.⁶ In

⁵ It is possible to represent the content of proper names in the epsilon calculus by employing the iota operator ι (Cann et al. 2005); see also Russell (1905).

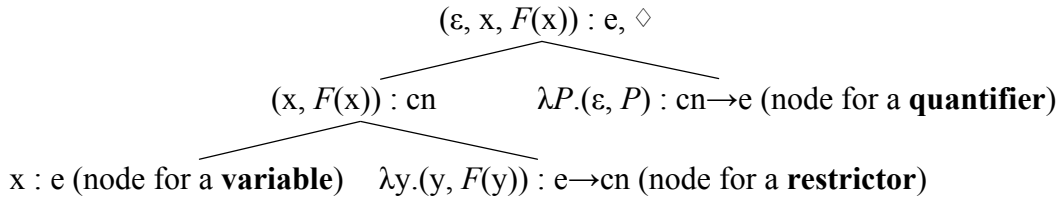
⁶ Thus, DS dispenses with devices that are stipulated for quantifiers at an object position such as ‘quantifier movement’ and ‘type-shifting’ (Heim & Kratzer 1998). See Charnavel (2010) for experimental evidence that lends credence to this DS stance.

existential quantification, an epsilon term is as in $(\varepsilon, x, F(x))$, where (i) the epsilon operator ε is analogous to $\exists$, (ii) x is a variable bound by the operator, and (iii) $F(x)$ is a restrictor. In universal quantification, a binder that corresponds to $\forall$ is the tau operator τ . The binders ε and τ reflect the duality of $\exists$ and $\forall$, and one is derivable from the other. The epsilon term $(\varepsilon, x, F(x))$ denotes an arbitrary witness of the set characterised by F if the set is not empty; if the set is empty, the term picks out any arbitrary entity. Thus, the equivalence (3.8) holds between the epsilon calculus and the first-order predicate logic.

$$(3.8) \quad F(\varepsilon, x, F(x)) \equiv \exists x.F(x)$$

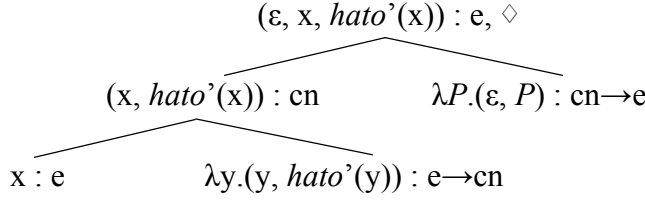
Each epsilon term has internal structure (3.9). A variable is introduced as an argument of the restrictor, and the application of the restrictor to the variable yields the content of a common noun; cn is a logical type for common nouns. If a quantifier is applied to the content of a common noun, it gives rise to an epsilon term.

(3.9) Internal structure of epsilon terms



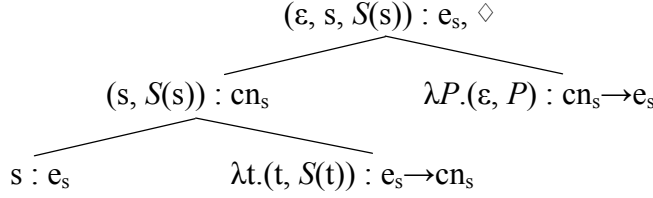
For instance, the quantified expression *hato* (= ‘a pigeon’)⁷ is mapped onto the epsilon term $(\varepsilon, x, \textit{hato}'(x))$ with the structure (3.10). (As we shall see in Sect 3., each lexical item encodes an instruction to construct a tree. Thus, the structure (3.10) is built up by the parser following the entry of the lexical item *hato*.)

⁷ Bare nouns in Japanese are underspecified for definiteness and plurality, and they are inferred contextually (see Sect. 1, Chap. 2).

(3.10) Epsilon term for *hato* (= ‘a pigeon’)

A situation term *SIT* is also notated in the epsilon calculus as in (3.11), where *S* is a situation predicate. This is a temporary expedient to be replaced with a full-fledged predicate which models, among other things, tense; see Sect. 4.4, Chap. 4.⁸

(3.11) Situation term



To avoid this structural complexity, however, epsilon terms will be represented as if they lack internal structure in what follows.

Once a proposition is constructed, it is subject to QUANTIFIER EVALUATION (Q-EVALUATION). This process embellishes each term in the proposition so that it reflects the **full** semantic dependencies in the proposition into the restrictor of the term. In the simplest case, where there is no scope dependence, we see most transparently the relation of the epsilon term under construction and that same term once evaluated. For instance, the schematic formula (3.12) is evaluated as (3.13). Each of the predicates *G* and *F* is applied to the term *a*, which reflects these predicates into the restrictor, and the resulting propositions are combined by an appropriate connective (e.g., & for existential quantification, $\rightarrow$ for universal quantification).

(3.12) $G(\varepsilon, x, F(x))$

⁸ For instance, a situation predicate modelling a past event is $\lambda z.[z \subseteq R \& R < s_{\text{NOW}}]$, where *R* is an underspecified time interval and *s_{NOW}* is an utterance time. For details, see Sect. 4.4, Chap. 4.

$$(3.13) \quad G(a) \& F(a) \quad \text{where } a = (\epsilon, x, G(x) \& F(x))$$

The proposition (3.12) contains only a single term, but the process of Q-EVALUATION is complicated when there is more than one term. Consider the string (3.14) and its tree representation (3.15). As mentioned above, τ is the tau binder that corresponds to the universal operator $\forall$ in the predicate logic.

$$(3.14) \quad \begin{array}{l} \text{Dono-hato-mo} \quad \text{ton-da.} \\ \text{every-pigeon-also} \quad \text{fly-PAST} \\ \text{'Every pigeon flew.'} \end{array}$$

$$(3.15) \quad \text{Parsing } \textit{Dono-hato-mo ton-da}$$

$$\begin{array}{c} \text{ton}'(\tau, x, \text{hato}'(x))(\epsilon, s, S(s)) : t, \diamond \\ \swarrow \quad \searrow \\ (\epsilon, s, S(s)) : e_s \quad \text{ton}'(\tau, x, \text{hato}'(x)) : e_s \rightarrow t \\ \swarrow \quad \searrow \\ (\tau, x, \text{hato}'(x)) : e \quad \text{ton}' : e \rightarrow (e_s \rightarrow t) \end{array}$$

If a proposition contains n-many terms, Q-EVALUATION takes place n-many times. This has the effect of explicating the scope of quantifiers involved.⁹ The procedure is that a term that has narrower scope is evaluated first. Suppose that in (3.15), the situation term $(\epsilon, s, S(s))$ out-scopes the individual term $(\tau, x, \text{hato}'(x))$. In this case, $(\tau, x, \text{hato}'(x))$ is Q-EVALUATED first. Thus, the proposition in (3.15) is updated into (3.16), where the content of the whole proposition is pushed down to the restrictor of the term a, the evaluated version of $(\tau, x, \text{hato}'(x))$.

$$(3.16) \quad \text{hato}'(a) \rightarrow \text{ton}'(a)(\epsilon, s, S(s)) \quad \text{where } a = (\tau, x, \text{hato}'(x) \rightarrow \text{ton}'(x)(\epsilon, s, S(s)))$$

⁹ To model quantifier scope, a 'scope statement' is constructed at a propositional node. A scope statement is in the form $\text{Scope}(x < y)$, expressing that a term that binds variable x out-scopes a term that binds variable y . Every time a quantified expression is scanned, a scope statement is gradually updated. Scope ambiguities are modelled in terms of underspecification in the scope statement (Kempson et al. 2001: Chap. 7).

This is then evaluated with respect to the situation term $(\epsilon, s, S(s))$ and updated into (3.17), where b , the evaluated version of $(\epsilon, s, S(s))$, reflects the content of the whole proposition (3.17) into the restrictor of the term. This process uniformly applies to **every** occurrence of the situation term. For instance, it applies to $(\epsilon, s, S(s))$ inside the individual term a in (3.16), with the updated individual term being notated as a_b .

$$(3.17) \quad S(b) \& [hato'(a_b) \rightarrow ton'(a_b)(b)]$$

$$\text{where } b = (\epsilon, s, S(s) \& [hato'(a_s) \rightarrow ton'(a_s)(s)])$$

$$a_b = (\tau, x, hato'(x) \rightarrow ton'(x)(b))$$

$$a_s = (\tau, x, hato'(x) \rightarrow ton'(x)(s))$$

The evaluated proposition (3.17) explicates the scope relation: there is a situation with the property S (see the discussion around (3.11)), where every pigeon flew. The nitty-gritty of Q-EVALUATION is that (i) each term in a proposition denotes an entity that reflects every description of the proposition into the restrictor and (ii) the resulting proposition is explicit with regard to the scope of quantifiers involved.

2.3 Summary

A DS representation is a binary semantic tree with an argument on the left and the functor on the right. In every step of tree update, there is always a single node under development, marked by a pointer $\diamond$. Each node is decorated with a pair of formula (i.e., content) and labels (e.g., a logical type, a tree-node identifier, etc., as expressed using LOFT). The content of quantified expressions (including situation terms) is represented as an epsilon term, a type- ϵ term consisting of a binder, a variable, and a restrictor, as defined in the epsilon calculus. Once a proposition is built up, all epsilon terms within the proposition are evaluated; each term reflects the full content of the proposition within the restrictor of the term. This process, Q-EVALUATION, explicates the scope dependencies in the proposition. Now that the declarative aspect of the DS formalism is in place, the next issue is how the DS tree is generated in an incremental fashion.

3 The Procedural Part

3.1 Actions for Tree Update

DS trees grow on the basis of left-to-right parsing of a string in context, and the final state represents an interpretation of the string. In non-final states, a DS tree is always partial in that there exists a node decorated with **requirements**. A requirement is notated as $?α$, requiring that $α$ will hold at the node. For instance, $?e$ requires that the node will be of type- e . Each node is created with some form of requirements, and these are a driving force of tree-update. In this sense, DS tree update is goal-directed.

Generally, a formal grammar consists of an axiom to specify an initial node and a set of rules which enriches the tree (Partee et al. 1990: 437). Within DS, too, the starting point of tree update is determined by an AXIOM, and it is subsequently updated by a set of rules to induce a set of actions to satisfy a range of requirements. A DS tree is **well-formed** iff no outstanding requirements are left in the tree. A string is said to be **grammatical** iff there exists a series of tree-updates from the AXIOM that leads to a well-formed tree.

The AXIOM sets out the **initial** node (3.18), with the pointer $◇$ indicating that this node is under development. $?t$ requires that this node will be of type- t .¹⁰ $Tn(U)$ is an underspecified node address and $?∃x.Tn(x)$ requires that the ‘meta-variable’ U (i.e., a temporary place-holder for a proper value) will be saturated (i.e., a node address will be fixed). This tree-node underspecification ensures that the initial node may (but does **not** have to) be a root of the whole tree. This is contrasted with previous DS accounts (Cann et al. 2005, Kempson et al. 2001, Kempson et al. 2011), in which the AXIOM introduced the root node of the overall tree.

¹⁰ This is also true of non-declaratives. See Kempson et al. (2001: Chap. 5) for interrogatives, and see Chatzikyriakidis (2009: Sect. 5.2) for imperatives.

(3.18) AXIOM¹¹
 $?t, Tn(U), ?\exists x.Tn(x), \diamond$

The parser develops this initial node gradually and monotonically by running an action to update a tree. DS defines three types of actions: general, lexical, and pragmatic actions. They are all defined as a program in **conditional** format.¹²

(3.19) Format of DS tree-update actions

IF	...	(condition to be fulfilled by an active node highlighted by $\diamond$)
THEN	...	(action to be run if the condition is fulfilled)
ELSE	...	(action to be run if the condition is not fulfilled)

This conditional format may be of arbitrary complexity; for example, the THEN-block may nest another conditional sequence:

(3.20) Embedding of a conditional sequence

IF	...	
THEN	IF	...
	THEN	...
	ELSE	...
ELSE	...	

Furthermore, all of general, lexical, and pragmatic actions are defined with **the same vocabulary** (e.g., formula value, logical type, LOFT decorations). The three types of DS actions are discussed in what follows.

General actions are stored in the DS system and not lexicalised. For instance, there is a set of general actions to induce a structurally unfixed node in a tree; concrete

¹¹ For the sake of simplicity, $Tn(U)$ and $? \exists x.Tn(x)$ are suppressed in the remainder of the thesis. Instead, a saturated node-address with an arbitrary symbol α is used, as in $Tn(\alpha)$.

¹² Two caveats are in order. First, the general action is usually called ‘computational action,’ but I use the term ‘general action’ because every DS action is computational in that it operates on an input representation and outputs a distinct representation. Second, the general action is in Input-Output format in Cann et al. (2005) and Kempson et al. (2001), but this thesis follows Cann & Kempson (2008: 190) and elsewhere in using the conditional format for all DS action packages. This meshes well with the view that some general actions are calcified as lexical actions through routinisation processes (Bouzouita 2008).

examples of general actions will be presented in Sect. 3.3 below. The application of general actions is **optional**; the parser may run general actions at any time as long as the IF-block is satisfied by an active node. Due to the optional nature of general actions, an unambiguous string could be analysed in terms of distinct tree-updates leading to the same final state. Consequently, there is no one-to-one relation between a construction and a set of actions to generate the construction. In this spirit, this thesis does not propose any fixed sets of actions for the constructions such as ‘clefts’ and ‘relatives,’ but explores possible combinations of actions that map a string onto a well-formed tree, which eventually characterises ‘cleft constructions,’ ‘relative constructions,’ etc.

The second type of tree-update action is **lexical actions**, encoded in each item (both content and function words, including affixes). Lexical actions differ from general actions **solely** in terms of optionality: a program encoded in a lexical item must fire every time the lexical item is scanned. Consider the entry of *Pazu* (3.21). The IF-line specifies an input condition: the active node (marked by a pointer $\diamond$) should be a type-e-requiring node. If this is met, the action sequence in the THEN-line fires. In (3.21), this action sequence contains a single atomic action, $\text{put}(Pazu' : e)$, where $\text{put}(\alpha)$ is an action to posit α at the node. Thus, $\text{put}(Pazu' : e)$ decorates the current node with the content *Pazu'* and the logical type *e*. If the input condition is not satisfied, the action sequence in the ELSE-block applies. In (3.21), ABORT quits the tree update.

(3.21) Entry of *Pazu*

IF	?e
THEN	$\text{put}(Pazu' : e)$
ELSE	ABORT

The action encoded in *Pazu*, which only annotates a node, is simple. It should not be assumed, however, that every noun encodes only a node-decoration action. Sect. 2.2 stated that bare nouns such as *hato* (= ‘a pigeon’) were always mapped onto an epsilon term with internal structure (though an internal structure is often suppressed for reasons

of brevity); thus, they are assigned an instruction to build a node (as well as decorating a node). The lexically triggered node-building process is illustrated in the next subsection, where the lexical encoding of verbs is discussed.

Finally, **pragmatic actions** are a set of actions whose rule-structures are stored in the DS system but whose execution largely involves pragmatic inference. A case in point is SUBSTITUTION, an action to contextually assign a value to a meta-variable. More specifically, if a current node is decorated with a meta-variable *U* of type-e (i.e., a place-holder for a type-e content), the parser saturates *U* by copying a previous type-e content onto the node (i.e., anaphora) or by detecting a salient individual and replacing *U* with the term denoting the individual (i.e., deixis) (see the discussion around (3.22)). Thus, there is an intimate interaction between the DS grammar and the pragmatics system. This thesis does not presuppose any particular pragmatic theory, but a possible candidate is Relevance Theory (Sperber & Wilson 1995, Wilson & Sperber 2012); see Cann et al. (2005) and Seraku (2008) for discussion. The reader may adopt her or his favoured pragmatic theory (e.g., Neo-Gricean theories (Horn 1984, Levinson 2000)) as long as (i) it respects the DS grammar as a system independent of pragmatics but (ii) it intimately interacts with the DS grammar at every step of tree update.¹³

To sum up the flow of language understanding, the parser sets out the initial state (specified by the AXIOM) and develops it in a step-wise manner by executing general, lexical, or pragmatic actions to satisfy an array of requirements. The AXIOM and some general actions are universal,¹⁴ and all the other actions may vary across languages. For instance, the general action *ADJUNCTION introduces only a type-e-

¹³ It has recently been assumed that pragmatic inference is essential for constructing both the explicit content of an utterance (i.e., the proposition expressed) and the implicit content of an utterance (i.e., implicature) (e.g., Carston 2002, Recanati 2003). It is in principle possible for DS to represent the implicit content of an utterance (Seraku 2008), but what are meant by pragmatic effects in this thesis concern the construction of the explicit content of an utterance.

¹⁴ For cross-linguistically invariant general actions, see ‘general actions for basic tree-updates’ in the appendix.

requiring unfixed node in Japanese (Sect. 3.3). Another example is the pragmatic action SUBSTITUTION, where there may be language-specific binding-constraints.

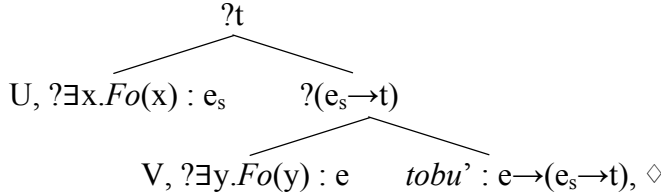
Before closing, it should be noted that though DS models incremental tree-update in online processing, DS only characterises a **possible range** of interpretations that are associated with a string in context. What DS itself does not model is a choice mechanism which permits the parser to choose one interpretation in context; this choice mechanism is provided by the interaction between the DS system and the pragmatics system. To take as an example the string *Pazu saw Uncle Pom with a lamp*, the DS system predicts that the string may be associated with two interpretations (i.e., ‘Pazu saw Uncle Pom who had a lamp’ and ‘Pazu saw Uncle Pom by the light of a lamp’), and it is through the interaction between the DS grammar and the pragmatics system that one particular reading is selected. This view should not be taken as suggesting that the parser chooses one interpretation only after the DS system outputs a possible set of readings; rather, the interaction between the two systems takes place at **each** stage of incremental processing.

3.2 Verbal Encoding

This sub-section illustrates lexical actions by contemplating the encoding of verbs. In Japanese, all arguments may be dropped (a so-called ‘pro-drop’ language; Sect. 1, Chap. 2). For instance, the verb *tobu-* (= ‘fly’), if suffixed by the past tense marker *da*, can stand without a subject being mentioned, as in *ton-da* (= ‘Some individual flew.’). It is thus assumed that verbs in Japanese project a **propositional schema** with argument slots that license contextual identification. For instance, the verb *tobu-* projects an open proposition where argument nodes are decorated with meta-variables. In (3.22), *U* is a meta-variable for a situation term, and *V* is a meta-variable for a subject term. These are accompanied by a requirement for a formula value such as $?\exists x.Fo(x)$. (*Fo* is a predicate

for formula values; see the appendix.) These requirements are explicitly expressed in (3.22), but they will be generally suppressed to avoid notational clutter.

(3.22) Output structure of parsing *tobu-* (= ‘fly’)



In this tree, the subject node is decorated with the meta-variable *V*. If the value is contextually accessible, *V* is saturated by the pragmatic action SUBSTITUTION (see the last sub-section). For example, in a context where a pigeon flew, *V* is substituted with $(\epsilon, x, \textit{hato}'(x))$, an epsilon term denoting a pigeon.

A set of actions to construct the tree (3.22) is encoded in the verb *tobu-* as in (3.23). This action package consists of several atomic actions. **make** creates a node, **go** moves the pointer $\diamond$, and **put** posits a decoration. Recall that $\langle \downarrow_0 \rangle$ refers to an argument daughter and $\langle \downarrow_1 \rangle$ refers to a functor daughter (Sect. 2.1).

(3.23) Entry of *tobu-* (= ‘fly’)

```

IF      ?t
THEN    make/go( $\langle \downarrow_0 \rangle$ ); put( $U, ?\exists x.Fo(x) : e_s$ ); go( $\langle \uparrow_0 \rangle$ );
        make/go( $\langle \downarrow_1 \rangle$ ); put( $?(e_s \rightarrow t)$ );
        make/go( $\langle \downarrow_0 \rangle$ ); put( $V, ?\exists y.Fo(y) : e$ ); go( $\langle \uparrow_0 \rangle$ );
        make/go( $\langle \downarrow_1 \rangle$ ); put( $tobu' : e \rightarrow (e_s \rightarrow t)$ )
ELSE    ABORT

```

What (3.23) declares is that if a current node is a type-*t*-requiring node, the following actions fire. The first line in the THEN-block: (i) creates an argument daughter node and moves the pointer $\diamond$ to the node; (ii) annotates it with a meta-variable *U*, the requirement $?\exists x.Fo(x)$, and the type e_s ; (iii) returns the pointer $\diamond$ to the mother. The second line in the THEN-block: (i) creates a functor daughter node and moves the pointer $\diamond$ to the node; (ii) annotates it with the requirement $?(e_s \rightarrow t)$. The third line in the

THEN-block: (i) creates an argument daughter node and moves the pointer $\diamond$ to the node; (ii) annotates it with a meta-variable V , the requirement $\exists y.Fo(y)$, and the type e ; (iii) returns the pointer $\diamond$ to the mother. The final line in the THEN-block: (i) creates a functor daughter node and moves the pointer $\diamond$ to the node; (ii) annotates it with the content *tobu*' and the type $e \rightarrow (e_s \rightarrow t)$. ABORT in the ELSE-block ensures that this action sequence is not operated unless the IF-block is satisfied.

In (3.22), a subject node (as well as the other nodes in the tree) is built by the parse of the verb *tobu*-. As pointed out in Sect. 1, Chap. 2, however, Japanese is strictly verb-final and any argument NPs come **before** the verb is scanned. Then, how does the parser process argument NPs? This issue is solved by 'structural underspecification and resolution,' to which we shall turn in the next sub-section.

3.3 Structural Underspecification

Japanese is strictly verb-final, and this raises a challenge to modelling incrementality in online processing: the verb comes at the end of a string, but it is the verb that projects the basic structure of the string. DS reconciles incremental processing with monotonic tree growth by letting a node be **underspecified for a position** when it is induced in the incremental parse process and **resolved subsequently**. With this positional uncertainty, the parser may build a partial tree or tree relation without waiting for a verb to come and without having to destroy any attribute of the emergent tree.¹⁵ DS defines three general actions to introduce an unfixed node with a **different locality specification**.

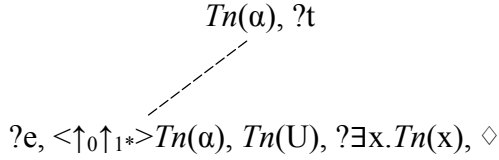
- (3.24) Structural underspecification with three types of locality specification
- a. LOCAL *ADJUNCTION: an unfixed node that must be resolved 'locally'
 - b. *ADJUNCTION: an unfixed node that may be resolved 'non-locally'
 - c. GENERALISED ADJUNCTION: an unfixed node that may be resolved 'globally'

¹⁵ This makes it unlike other frameworks which involve a 'build and revise' approach (Phillips 2003). See Sect. 7 for relevant discussion.

Each general action may be executed only if a pointer $\diamond$ is at a node decorated with the requirement $?t$, to the effect that an unfixed node is always defined with respect to a type- t -requiring node.

First, LOCAL *ADJUNCTION introduces an unfixed node to be resolved within a **local** proposition. In (3.25), a locally unfixed relation is indicated by a dashed line. The requirement $? \exists x. Tn(x)$ declares that the meta-variable U in $Tn(U)$ will be saturated. That is, it states that this unfixed node will be assigned a fixed tree-node address. (This requirement is explicitly expressed in (3.25) but will be omitted in later sections.)

(3.25) LOCAL *ADJUNCTION (tree display)



The locality restriction is expressed by the label $\langle \uparrow_0 \uparrow_1^* \rangle Tn(\alpha)$ at the unfixed node. $\langle \uparrow_0 \uparrow_1^* \rangle Tn(\alpha)$ declares that if a pointer $\diamond$ moves one-node up through an argument path and possibly keeps moving up through functor paths, it will reach a $Tn(\alpha)$ -node. Thus, $\langle \uparrow_0 \uparrow_1^* \rangle Tn(\alpha)$ may be $\langle \uparrow_0 \rangle Tn(\alpha)$, $\langle \uparrow_0 \uparrow_1 \rangle Tn(\alpha)$, $\langle \uparrow_0 \uparrow_1 \uparrow_1 \rangle Tn(\alpha)$, etc. What it prevents is a crossing of a type- t -requiring node, in which case the relation includes more than one $\uparrow_0$, as in $\langle \uparrow_0 \uparrow_1 \uparrow_0 \uparrow_1 \rangle Tn(\alpha)$, which contradicts $\langle \uparrow_0 \uparrow_1^* \rangle Tn(\alpha)$. Therefore, the unfixed node induced by LOCAL *ADJUNCTION will have to be fixed under the **closest** type- t -requiring node. This general action is defined in (3.26).

(3.26) LOCAL *ADJUNCTION (rule formulation)

```

IF       $Tn(\alpha), ?t$ 
THEN   make/go( $\langle \downarrow_1^* \downarrow_0 \rangle$ ); put( $?e, \langle \uparrow_0 \uparrow_1^* \rangle Tn(\alpha), Tn(U), ?\exists x. Tn(x)$ )
ELSE   ABORT

```

There are two means of resolving this type of unfixed relation. First, the parser may run the general action UNIFICATION: a type- e -requiring unfixed node is **merged** with a type- e fixed node, as a result of which the fixed node is annotated with the union

of the decorations of the two nodes (see Sect. 3.3.2, Chap. 5).¹⁶ Alternatively, a locally unfixed node may be **lexically** resolved by the parse of a case particle. For instance, consider the entry of the nominative-case particle *ga* (3.27).

(3.27) Entry of the nominative-case particle *ga* (to be refined in Sect. 5)

```

IF      e
THEN    IF      <↑0↑1*>(Tn(α), ?t)
        THEN    put(?<↑0>(es→t))
        ELSE    ABORT
ELSE    ABORT

```

If a current node is a locally unfixed type-*e* node, the parser in processing *ga* posits $?<↑_0>(e_s \rightarrow t)$ as shown in (3.28), where β is an arbitrary type-*e* term. This is an ‘output filter’ on a DS tree since it requires that the node will be a subject node (cf., (3.7)).¹⁷

(3.28) Output structure of parsing the case particle *ga* (to be refined in Sect. 5)

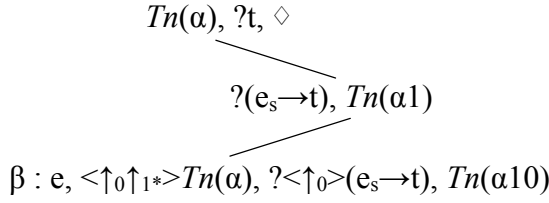
$$\begin{array}{c}
 Tn(\alpha), ?t \\
 \swarrow \\
 \beta : e, <↑_0↑_1*>Tn(\alpha), ?<↑_0>(e_s \rightarrow t), Tn(U), ?\exists x.Tn(x), \diamond
 \end{array}$$

When propositional structures are embedded, there are multiple subject node positions. Thus, $?<↑_0>(e_s \rightarrow t)$ itself cannot narrow down possible subject nodes into a unique one. But the label $<↑_0↑_1*>Tn(\alpha)$, provided by LOCAL *ADJUNCTION, declares that the current unfixed node will be resolved within the **local** structure. Thus, the combination of $?<↑_0>(e_s \rightarrow t)$ and $<↑_0↑_1*>Tn(\alpha)$ determines a **unique** subject position. Therefore, the unfixed node in (3.28) must be resolved as a subject node within the local structure.

¹⁶ UNIFICATION is called MERGE in Cann et al. (2005: 63) and Kempson et al. (2001: 86).

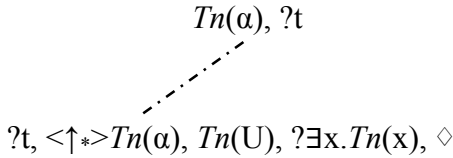
¹⁷ Cann et al. (2005: 254) stipulate that *ga* does not provide an output filter but it resolves an unfixed node immediately, provided that long-distance scrambling of a *ga*-marked item is degraded. But Mihara (1992: 176-7) notes that such scrambling is possible when the embedded verb is compatible only with the scrambled subject NP (but not with the embedded subject NP). Thus, this thesis assumes that *ga* encodes an output filter, like other case particles.

(3.29) Resolving the unfixed node in (3.28)



Let us turn to the second action in (3.24). *ADJUNCTION introduces an unfixed node to be resolved in a propositional domain possibly contained as an argument within another proposition. This ‘**non-local**’ property is expressed by $<\uparrow^*>Tn(\alpha)$ at the unfixed node, which allows a pointer $\diamond$ to cross a type- t -requiring node. Thus, $<\uparrow^*>Tn(\alpha)$ may be $<\uparrow_0 \uparrow_1 \uparrow_0 \uparrow_1 > Tn(\alpha)$, for example. An unfixed relation introduced by *ADJUNCTION is shown by a dashed-dotted line.

(3.30) *ADJUNCTION (tree display)



In Kempson & Kiaer (2010: 162), *ADJUNCTION induces a **type-e**-requiring or a **type-t**-requiring unfixed node. Following Seraku (2013b), however, I shall assume that *ADJUNCTION induces only a **type-t**-requiring unfixed node (at least in Japanese).¹⁸ The underlying idea is the division of labour: LOCAL *ADJUNCTION concerns NP arguments, while *ADJUNCTION concerns clause embedding. A type- t -requiring node introduced by

¹⁸ As will be demonstrated in Chap. 5, this assumption is essential for accounting for the case-marking pattern of clefts, stripping, and sluicing in Japanese. The heart of *ADJUNCTION is to introduce an unfixed node that may be resolved non-locally (but not across adjunct structures). As long as this basic restriction is maintained, it is plausible for each individual language to impose its own constraints. Cann (2011: 285) claims that LOCAL *ADJUNCTION in English encodes an additional restriction that it cannot fire if a current node dominates a node. Further, Cann et al. (2005) assume that *ADJUNCTION also encodes the same restriction. This restriction in *ADJUNCTION is proposed to account for certain English grammatical facts; this additional restriction is not adopted in the present thesis, which articulates a grammar of Japanese. I shall leave for future studies an account of cross-linguistic variation in terms of added restrictions in general actions.

*ADJUNCTION may be fixed by the general action UNIFICATION or by the lexical action encoded in the complementiser *to* (see Sect. 4.2). *ADJUNCTION is defined in (3.31)

(3.31) *ADJUNCTION (rule formulation)

```

IF       $Tn(\alpha), ?t$ 
THEN    $\text{make/go}(<\downarrow*>); \text{put}(?t, <\uparrow*>Tn(\alpha), Tn(U), ?\exists x.Tn(x))$ 
ELSE   ABORT

```

The reader might wonder how the assumption that *ADJUNCTION induces only a **type-t**-requiring unfixed node is compatible with the long-distance dependency constructions, where a displaced element is of **type-e**. This issue will be addressed in Chap. 5, where long-distance clefts are treated by the feeding applications of *ADJUNCTION and LOCAL *ADJUNCTION (cf., Kempson & Kiaer 2010).

Finally, the third general action in (3.24), GENERALISED ADJUNCTION, induces a node that may be settled within **any** connected domain including even adjuncts and islands. This type of unfixed relation is marked with $<U>Tn(\alpha)$, where the ‘up’ operator $<U>$ models the **weakest** relation and allows a pointer $\diamond$ to cross an adjunct or an island boundary (see Sect. 5). An unfixed relation introduced by GENERALISED ADJUNCTION is expressed by a dotted line.

(3.32) GENERALISED ADJUNCTION (tree display)

$$\begin{array}{c}
 Tn(\alpha), ?t \\
 \vdots \\
 ?t, <U>Tn(\alpha), Tn(U), ?\exists x.Tn(x), \diamond
 \end{array}$$

In Cann et al. (2005: 242), GENERALISED ADJUNCTION generates only a type-t-requiring unfixed node. It is in principle possible for GENERALISED ADJUNCTION to induce a type-e-requiring unfixed node, but to keep things simple, I shall assume that it is **only** LOCAL *ADJUNCTION that induces a type-e-requiring unfixed node in Japanese.

(3.33) GENERALISED ADJUNCTION (rule formulation)

IF $Tn(\alpha), ?t$
 THEN **make/go**(<D>); **put**(<U> $Tn(\alpha), Tn(U), ?\exists x.Tn(x)$)
 ELSE ABORT

GENERALISED ADJUNCTION has been utilised in modelling Japanese relatives. In the previous DS literature (Cann et al. 2005, Kempson & Kurosawa 2009, Kempson et al. 2011), however, the mechanism of resolving a globally unfixed relation for the relative clause was obscure; at least, it cannot be fixed by UNIFICATION, which only applies to non-globally unfixed relations. In Sect. 5, I argue that the globally unfixed relation is resolved by the node-identification process (Seraku in press a).

‘Structural underspecification and resolution’ is a central DS mechanism, but similar mechanisms are proposed elsewhere. First, some sentence processing studies also assume positional underspecification (Miyamoto 2002). But in these studies, this is a property of the parser, not the core grammar. Second, some syntactic theories define a similar device as a property of the grammar such as ‘functional uncertainty’ of Lexical Functional Grammar (Kaplan & Zaenen 1989). Unlike DS, the functional uncertainty is expressed over syntactic structure. Further, ‘structural underspecification’ within DS is subject to a tight constraint imposed by LOFT, and this restriction predicts the case-marking pattern of clefts, stripping, and sluicing in Japanese (Sects. 4 and 6, Chapter 5).

3.4 Summary

DS trees grow as the parser processes a string left-to-right. The AXIOM specifies the initial tree state as a single node decorated with (i) the requirement $?t$ that this node will be propositional, (ii) the node-address $Tn(U)$, and (iii) the requirement $? \exists x.Tn(x)$ that the meta-variable U will be saturated (i.e., the node-address will be fixed). This initial state is progressively developed by a combination of general, lexical, and pragmatic actions. Since Japanese is a pro-drop language, it is verbs that project a structure with argument slots. Despite this central role played by verbs, the parser starts to build up a

structure before a verb comes in the surface sequence: the parser processes NPs by dint of ‘structural underspecification and resolution.’ DS defines three general actions for structural underspecification, each of which introduces an unfixed node with a certain locality specification. In the present thesis, it is assumed that in Japanese, only LOCAL *ADJUNCTION introduces a type-e-requiring unfixed node, while the other actions (i.e., *ADJUNCTION, GENERALISED ADJUNCTION) introduce a type-t-requiring unfixed node. DS structure-building is completed once all requirements in the tree are fulfilled, in which case (and only in which case) the tree is well-formed. A string is then said to be grammatical iff there is a series of tree-updates from the AXIOM that reaches a well-formed tree state.

4 Sample Tree Transitions

With the declarative and procedural aspects of the DS formalism, it is now possible to go through DS tree updates with concrete examples: a simple sentence (Sect. 4.1) and a complex sentence with clause embedding (Sect. 4.2).

4.1 Simple Sentence

Let us consider how the string (3.34) is parsed word-by-word and a DS semantic tree is updated accordingly until it arrives at a final state which represents the interpretation of the string: ‘Pazu ran.’

- (3.34) Pazu-ga hashi-tta.
 Pazu-NOM run-PAST
 ‘Pazu ran.’

The initial state is determined by the AXIOM, with the requirement ?t, declaring that this node will be propositional. This initial node will be developed progressively by a combination of general, lexical, and pragmatic actions, as motivated by some form of

requirements. (For the sake of brevity, the node-address in (3.35) has been specified as $Tn(\alpha)$, with the use of an arbitrary constant α .)

$$(3.35) \quad \text{AXIOM} \\ Tn(\alpha), ?t, \diamond$$

First, the parser runs the general action LOCAL *ADJUNCTION to induce a type-e-requiring unfixed node.

$$(3.36) \quad \text{LOCAL *ADJUNCTION}$$

$$\begin{array}{c} Tn(\alpha), ?t \\ \diagdown \\ ?e, <\uparrow_0 \uparrow_1^* > Tn(\alpha), \diamond \end{array}$$

This unfixed node is decorated with $?e$, a requirement that this node will be of type-e and $<\uparrow_0 \uparrow_1^* > Tn(\alpha)$, a locality restriction that this node will be resolved within a local propositional structure.

The initial item *Pazu* decorates this unfixed node with the semantic content *Pazu'* and the logical type *e*.

$$(3.37) \quad \text{Parsing } Pazu$$

$$\begin{array}{c} Tn(\alpha), ?t \\ \diagdown \\ Pazu' : e, <\uparrow_0 \uparrow_1^* > Tn(\alpha), \diamond \end{array}$$

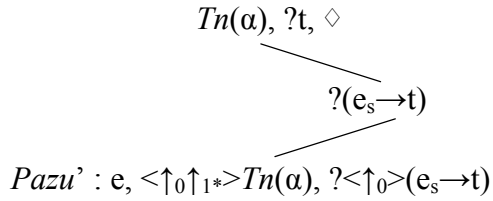
The requirement $?e$ at the unfixed node has been removed in (3.37) because it is met by the label *e*.¹⁹

The unfixed node is further decorated by the nominative-case particle *ga*, which provides $?<\uparrow_0 > (e_s \rightarrow t)$. Given the rationale in Sect. 3.3, this allows a process of

¹⁹ This removal process is formulated as THINNING. This general action deletes a requirement once it is satisfied (see the appendix).

resolving the unfixed node, fixing it as a subject node within a local proposition; and the pointer $\diamond$ returns to the top node.²⁰

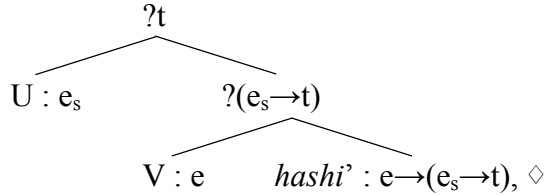
(3.38) Parsing *Pazu-ga*



The dashed line is now solid in (3.38), which visually indicates that the structurally underspecified relation has been updated to a fixed one.

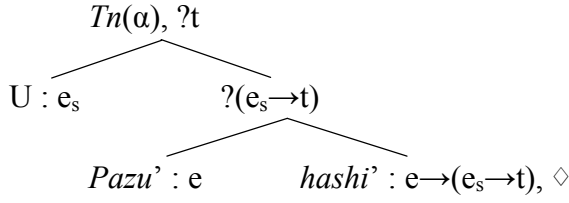
The next item to be parsed is the verb *hashi-* (= ‘run’). As shown in (3.39), the parse of *hashi-* generates an open propositional structure where a situation node and a subject node are both decorated with a meta-variable.

(3.39) Output structure of parsing *hashi-* (= ‘run’)



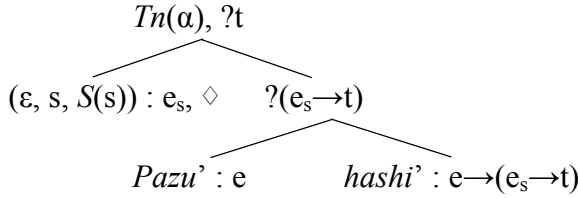
In (3.38), a subject node is already present. This pre-existing subject node harmlessly collapses with a subject node newly introduced by *hashi-*. This is because the subject node created by *hashi-* is decorated with a meta-variable (i.e., V in (3.39)) and it is fully **commensurate** with any specified formula (e.g., *Pazu'* in (3.38)). (In (3.40), irrelevant labels are glossed over.)

²⁰ This pointer movement is formulated as COMPLETION. This general action moves a pointer $\diamond$ upwards once the current node is decorated with a logical type (see the appendix).

(3.40) Parsing *Pazu-ga hashi*

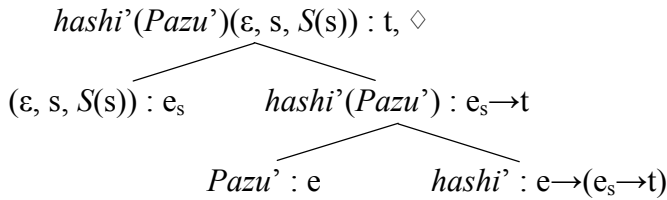
The meta-variable U at the situation node may be pragmatically substituted with the situation term $(\epsilon, s, S(s))$.²¹ The predicate S , if fully developed by the parse of a tense marker (see below), models, among other things, the tense of the string.

(3.41) SUBSTITUTION



In (3.41), two daughter nodes at the bottom are specified for content and type, licensing functional application and type deduction. This process is formalised as the general action **ELIMINATION**, which operates upon two daughters to yield a decoration on the mother. The same procedure applies to the intermediate argument-functor pair to calculate the content and the type of the top node.

(3.42) ELIMINATION (twice)



²¹ For **SUBSTITUTION** to occur, a pointer $\diamond$ needs to (i) move up to the top node and (ii) move down to the situation node. The former is formalised as **COMPLETION** (see footnote 20), while the latter is formalised as **ANTICIPATION**. The general action **ANTICIPATION** moves a pointer $\diamond$ down to a node that is decorated with a requirement (see the appendix).

Finally, the past-tense suffix *tta* adds tense information to the tree. Tense is represented as a restrictor within an event term (Cann 2011), but this complication is ignored until Sect. 4.4, Chap. 4. Thus, the tree (3.42) is in a final state. The top node is decorated with a propositional formula, and it is subject to QUANTIFIER EVALUATION. In the evaluated proposition (3.43), the full content of the proposition is reflected into the restrictor of the situation term *a*. This proposition represents the interpretation of the string: ‘Pazu ran.’

(3.43) Evaluating the proposition in (3.42)

$$S(a) \& hashi'(Pazu')(a) \quad \text{where } a = (\epsilon, s, S(s) \& hashi'(Pazu')(s))$$

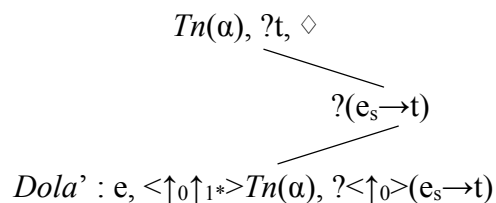
4.2 Clause Embedding

Let us turn to a slightly more complex example involving clause embedding (3.44), where the complementiser *to* is used. As before, since the language is rigidly verb-final, the verb comes at the end of both the subordinate and the matrix clauses.

(3.44) Dola-ga [Pazu-ga hashi-tta to] tsutae-ta.
 D-NOM [P-NOM run-PAST COMP] tell-PAST
 ‘Dola told that Pazu ran.’

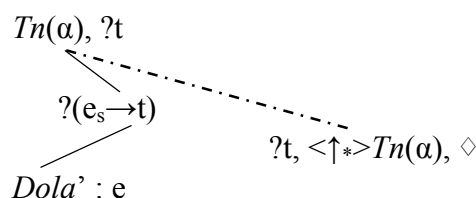
As always, the AXIOM specifies an initial node with the requirement ?*t*. Then, the parser runs LOCAL *ADJUNCTION to induce a type-e-requiring unfixed node, which is decorated by the initial item *Dola*. This unfixed node is resolved as a subject node by the nominative-case particle *ga*, which posits $? \langle \uparrow_0 \rangle (e_s \rightarrow t)$. So far, the tree update has been exactly the same as the one given in the last sub-section.

(3.45) Parsing *Dola-ga*



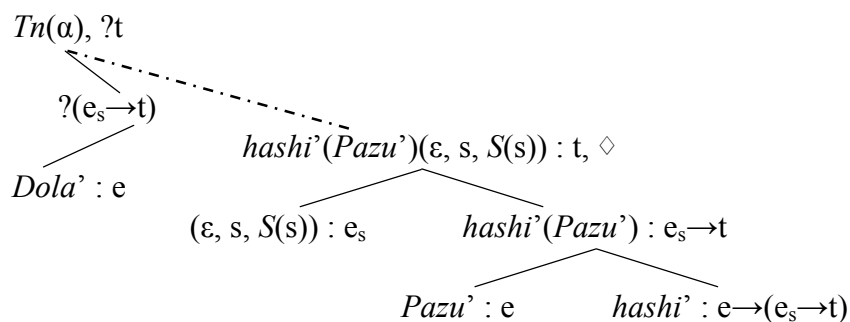
In (3.44), the next item *Pazu* belongs to an embedded clause. Thus, the parser runs *ADJUNCTION²² to induce a **type-t**-requiring unfixed node to create an initial node for an embedded structure as in (3.46), where some decorations are disregarded.

(3.46) *ADJUNCTION



It is under the current type-t-requiring node that the embedded clause *Pazu-ga hashi-tta* is scanned. The output is (3.47) (cf., the final state (3.42) in the last sub-section).

(3.47) Parsing *Dola-ga Pazu-ga hashi-tta*



Now, the complementiser *to* comes in. Unlike the English complementiser *that* which can mark both subjects and complements, *to* does not mark the sentential part as a subject of the embedding verb (Sect. 1, Chap. 2; see Kuno 1973a: 218, 1973b: 139). I

²² Alternatively, the parser could run GENERALISED ADJUNCTION. This general action is essential if the embedded clause is part of a complex NP (see Sect. 5.2). In (3.44), the embedded clause is not part of a complex NP, and the use of *ADJUNCTION suffices.

shall assume that the complementiser *to* **always** identifies the current type-*t* node as an **object** node in a fresh structure. The entry of *to* is defined as (3.48).²³ The IF-block states that the current node must be decorated with a proposition φ and the type *t*. If this input condition is met, the parser builds a partial structure bottom-up and identifies the type-*t* node as an object node within this partial structure. With this entry of *to*, the tree (3.47) is updated into (3.49).

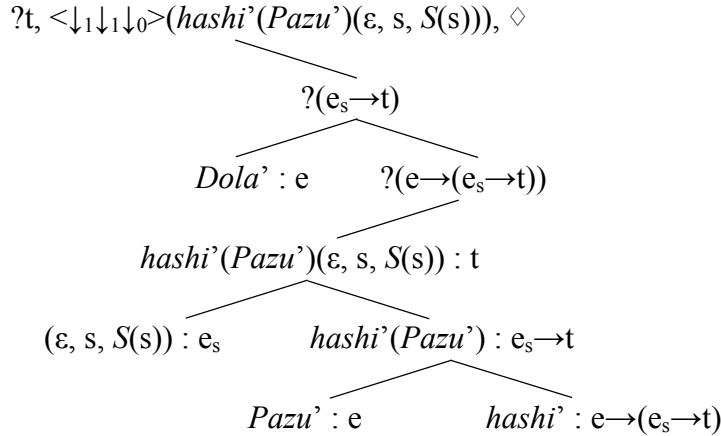
(3.48) Entry of the complementiser *to*

```

IF       $\varphi : t$ 
THEN   make/go( $\langle \uparrow_0 \rangle$ ); put( $\langle ?(e \rightarrow (e_s \rightarrow t)) \rangle$ );
        make/go( $\langle \uparrow_1 \rangle$ ); put( $\langle ?(e_s \rightarrow t) \rangle$ );
        make/go( $\langle \uparrow_1 \rangle$ ); put( $\langle ?t, \langle \downarrow_1 \downarrow_1 \downarrow_0 \rangle(\varphi) \rangle$ )
ELSE   ABORT

```

(3.49) Parsing *Dola-ga Pazu-ga hashi-tta to*

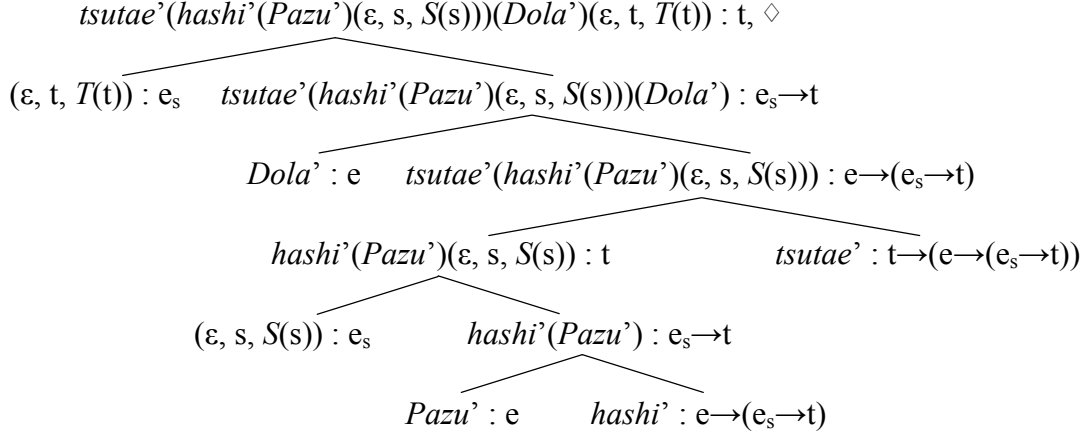


The rest of the process is as usual. The matrix verb *tsutae-* (= ‘tell’) fleshes out the matrix structure, creating a situation node, a subject node, and an object node. A meta-variable at the situation node is pragmatically saturated as, say, $(\varepsilon, t, T(t))$. The

²³ This entry differs from the entry of *to* in Cann et al. (2005: 245) in three respects. First, Cann et al. (2005) assume that the current node must be globally unfixed. Given our assumption that the initial node may be a non-root, this condition is no longer required. Second, Cann et al. (2005) characterise the building action weakly as make/go($\langle \uparrow_* \rangle$). In (3.48), this is detailed as make/go($\langle \uparrow_0 \rangle$) and make/go($\langle \uparrow_1 \rangle$) since the node will always be an object node. Finally, Cann et al. (2005) stipulate that if there is a structure above the current node, the parser marks the current node as an embedded structure. This stipulation is not adopted in (3.48) because in such cases, a new structure created by *to* harmlessly collapses with the pre-existing structure.

subject node collapses with the pre-existing node decorated by *Dola*. The object node collapses with the pre-existing node constructed by the embedded clause. The final state is given in (3.50). The proposition at the top node represents the interpretation of (3.44): ‘Dola told that Pazu ran.’ (Q-EVALUATION is omitted in this and subsequent examples.)

(3.50) Parsing *Dola-ga Pazu-ga hashi-tta to tsutae-ta*



4.3 Summary

The nuts and bolts of the DS formalism are now in our hands. The declarative part concerns trees and tree descriptions. DS trees are binary branching, each node being annotated with the pair: (i) a formula (i.e., semantic content, as expressed in the epsilon calculus in the case of quantified expressions) and (ii) labels (e.g., logical type, tree-node identifier, as expressed using LOFT). The procedural part concerns actions for tree updates. The initial node is determined by an AXIOM and it is subsequently developed by running general, lexical, and pragmatic actions until it reaches a well-formed state (i.e., a tree without outstanding requirements). A string is said to be grammatical iff it may be mapped onto a well-formed tree state. Tree transitions may involve structural underspecification. In Japanese, LOCAL *ADJUNCTION introduces a type-e-requiring node for processing NPs, while *ADJUNCTION and GENERALISED ADJUNCTION introduce a type-t-requiring node for processing subordinate clauses. In the ensuing two sections,

this basic architecture of the DS framework is augmented by defining LINK (Sect. 5) and specifying the DS notion of context (Sect. 6).

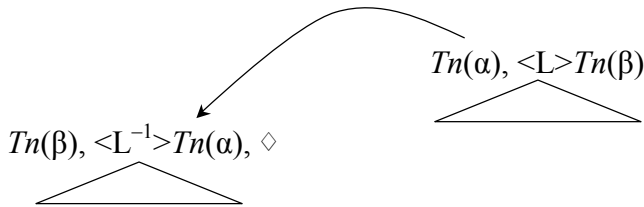
5 Paired Trees in LINK

5.1 LINK Relations

Up until now, we have focussed on unfolding one individual tree at a time (possibly, embedding a propositional structure as an argument). Within DS, however, not only the tree but also a ‘forest’ of trees may be constructed. That is, two discrete trees may be progressively built up in tandem, one of which is associated with the other through a shared element. This tree pairing is formalised as LINK, a term-sharing mapping from one tree onto the other.

LINK is governed by two LOFT-operators. From the perspective of a main tree, a LINKed, adjunct node is referred to by the LOFT-operator $\langle L \rangle$. Conversely, from the perspective of an adjunct tree, the inverse $\langle L^{-1} \rangle$ describes a node in a main tree from which a LINK relation starts. A LINK relation is expressed by the curved arrow, as in (3.51). $\langle L \rangle Tn(\beta)$ dictates that if the parser follows a LINK path, it will reach the $Tn(\beta)$ -node in the LINKed tree. In a similar vein, $\langle L^{-1} \rangle Tn(\alpha)$ dictates that if the parser follows an inverse LINK path, it will reach the $Tn(\alpha)$ -node in the main tree.

(3.51) LINK



What underlies a LINK relation is the presence of a **type-e** element which is **shared** by the paired trees. During the course of LINK transition, the parser typically

provides a requirement on a shared content.²⁴ For instance, consider the relative clause sequence *A boy who plays the trumpet likes pigeons*. The head noun *a boy* decorates a node in a main tree, and it is LINKed to a subordinate tree, during which the parser posits a requirement that the subordinate tree will contain a node decorated with the content of the head noun *a boy*. This requirement will be met when a meta-variable at the subject node in the subordinate tree is saturated as the content of *a boy*, yielding the closed proposition ‘A boy plays the trumpet.’

LINK relations are **free from** type-restrictions; in fact, in the schematic display (3.51), the $Tn(\alpha)$ -decorated node and the $Tn(\beta)$ -decorated node are not annotated with a logical type. What determines the LINK relation is not the logical typing but the tree-node address as expressed using the operators $\langle L \rangle$ and $\langle L^{-1} \rangle$, together with a shared type-e element. In the case of the relative clause construction (see the next sub-section), a LINK relation holds between a type-t node (i.e., the node of the relative clause) and a type-e node (i.e., the node of the head noun). In the case of clausal coordination (see Sect. 4.3, Chap. 4), a LINK relation holds between two type-t nodes.

Once a LINK relation is established, the content of one tree is incorporated into the content of the other tree. Suppose that the parser processes the sequence *a boy who plays the trumpet*. When a LINK relation is created between the main tree (i.e., the tree for the head noun *a boy*) and the subordinate tree (i.e., the tree for the relative clause *plays the trumpet*), the parser enriches the term of the head noun by incorporating the content of the relative clause into the restrictor of the term. Accordingly, the composite term denotes an individual who is a boy and who plays the trumpet. That is, LINK transitions consist of two parts: (i) the **introduction** of a LINK relation and (ii) the **enrichment** of the LINK relation. Each of these processes may be realised as an instance of general or lexical actions. For instance, we shall see shortly that in head-external relatives, both processes are formalised as general actions.

²⁴ In other cases, the parser directly decorates a node with a shared content (rather than positing a requirement). See the entry of the nominaliser *no* in Sect. 3, Chap. 4.

In sum, two distinct trees (of whatever type) may be constructed in tandem and paired through the forced sharing of a type-e content. A LINK relation is unidirectional, with one tree being subordinated to the other; this structural relation is expressed by using the LOFT operators $\langle L \rangle$ and $\langle L^{-1} \rangle$. Once a LINK relation is established, there is a second step: the content of one tree is incorporated into the content of the other tree. This two-stage LINK transition is exemplified in the next two sub-sections: the head-external relative in Japanese (Sect. 5.2) and the *wa*-topic construction (Sect. 5.3).

5.2 Head-External Relatives

In Japanese, there are a wide variety of relative clauses. This sub-section takes up the standard type of relatives, the head-external relative (3.52) (e.g., Hoshi 1995, Inoue 1976, Kaplan & Whitman 1995, Kuno 1973a). Other, less familiar types of relatives (e.g., head-internal relatives) will be examined in Chap. 6.

(3.52) Head-external relatives

[[Sheeta-ga motsu] ishi]-ga hika-tta.
 [[S-NOM hold] stone]-NOM shine-PAST
 ‘A stone which Sheeta holds shone.’

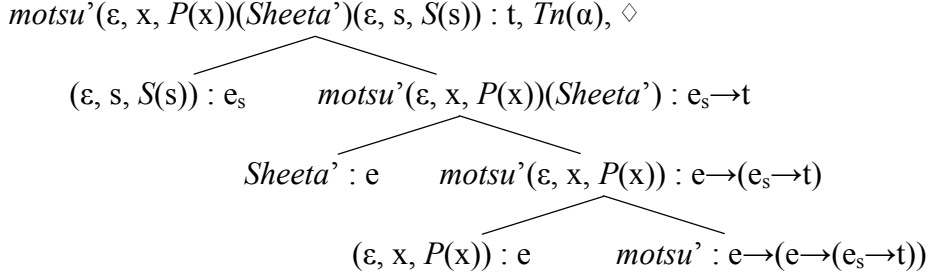
5.2.1 The Analysis

Based on Kempson & Kurosawa (2009) and Seraku (in press a), the tree update for the head-external relative (3.52) proceeds as follows. The AXIOM sets out an initial type-t-requiring node, under which a propositional structure is developed by the relative clause *Sheeta-ga motsu*.²⁵ The gap, or the object argument of *motsu* (= ‘hold’), is represented

²⁵ Since the node set out by the AXIOM may be a non-root, the relative clause is parsable directly under the initial node. Yet, as general actions are optional, nothing stops the parser from running GENERALISED ADJUNCTION to introduce a globally unfixed type-t-requiring node for processing the relative clause part. See Sect. 5.2.3.

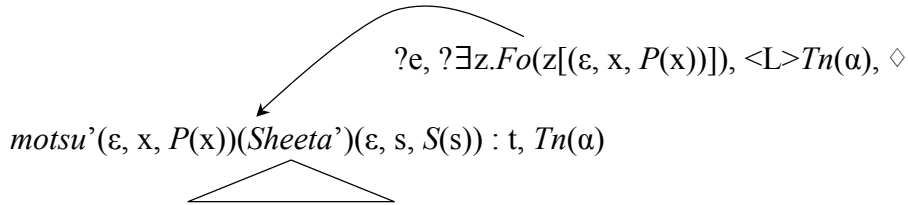
as the epsilon term $(\epsilon, x, P(x))$, where P is an abstract predicate (Kempson & Kurosawa 2009: 65).²⁶

(3.53) Parsing *Sheeta-ga motsu*



Now, the LINK transition comes in. As a first step, the general action LINK INTRODUCTION initiates an inverse LINK relation from the current node in the adjunct tree to a new type-e-requiring node in an unfolding main tree. This new type-e-requiring node is decorated with $?\exists z.Fo(z[(\epsilon, x, P(x))])$, requiring that this node will be decorated with a term z that contains $(\epsilon, x, P(x))$ as a sub-term. That is to say, the content of the gap, $(\epsilon, x, P(x))$, serves as a shared term in the paired trees.

(3.54) LINK INTRODUCTION



The action package of LINK INTRODUCTION is given in (3.55). The IF-block requires that a current node should be of type-t and that a node somewhere below should be decorated with a term α . In (3.54), α is instantiated as $(\epsilon, x, P(x))$. The inner THEN-

²⁶ Japanese being pro-drop, a node for a gap could be decorated with, say, $(\epsilon, x, \text{thing}'(x))$. In Kempson & Kurosawa (2009), the licensing condition on the predicate P is not fully detailed; and this thesis also leaves this issue for future work. Another possible option is to annotate a node for a gap with a variable (Cann et al. 2005, Kurosawa 2003). I thank Ronnie Cann and Ruth Kempson for pertinent discussions.

block declares that, if the IF-blocks are satisfied, the parser initiates an inverse LINK relation to a fresh node and annotates it with ?e and $? \exists z.Fo(z[\alpha])$.

(3.55) LINK INTRODUCTION

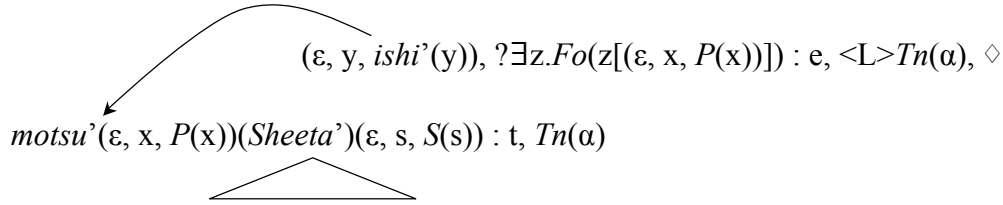
```

IF      t
THEN    IF      <D>(α)
        THEN    make/go(<L-1>); put(?e, ?∃z.Fo(z[α]))
        ELSE    ABORT
ELSE    ABORT

```

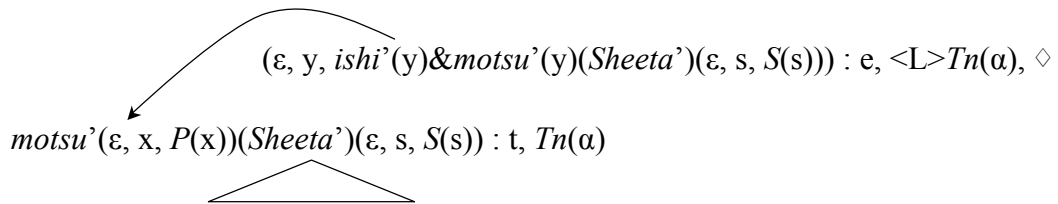
The current node in (3.54) is decorated by the head noun *ishi* (= ‘stone’), as a result of which the tree (3.54) is updated into (3.56).

(3.56) Parsing *Sheeta-ga motsu ishi*



As a second step, the current node in the emergent main tree is enriched with the content of the adjunct tree. This process is formulated as the general action LINK ENRICHMENT.²⁷ The parser takes the content of the relative clause and replaces the term $(\epsilon, x, P(x))$ with the variable y so that the variable y is restricted by the descriptions of the head noun and the relative clause. This composite term represents an individual which is a stone and which is held by Sheeta.

(3.57) LINK ENRICHMENT



²⁷ This is called LINK EVALUATION in Cann et al. (2005). To differentiate this operation from QUANTIFIER EVALUATION, this thesis calls it LINK ENRICHMENT.

LINK ENRICHMENT is defined as (3.58). The term $(\epsilon, y, \varphi(y))$ is the content of a head, and ψ is the content of a relative clause, which contains the term $(\epsilon, x, P(x))$ as an argument. The parser reflects ψ into the term $(\epsilon, y, \varphi(y))$ as an additional restrictor by replacing the term $(\epsilon, x, P(x))$ within ψ with the variable y .

(3.58) LINK ENRICHMENT

```

IF       $(\epsilon, y, \varphi(y)) : e$ 
THEN   IF       $\langle L \rangle(\psi[(\epsilon, x, P(x))])$ 
        THEN    $\text{put}(\epsilon, y, \varphi(y) \& \psi[y/(\epsilon, x, P(x))])$ 
        ELSE   ABORT
ELSE   ABORT

```

After these two steps in the LINK transition, what comes next in the string (3.52) is the nominative-case particle *ga*. This is where a technical problem arises: the current node in (3.57) is not suited to process the nominative-case particle *ga*.²⁸ To pinpoint the problem, consider the entry of *ga* (3.27), repeated here as (3.59).

(3.59) Entry of the nominative-case particle *ga* (to be refined in Sect. 5)

```

IF       $e$ 
THEN   IF       $\langle \uparrow_0 \uparrow_1^* \rangle(Tn(\alpha), ?t)$ 
        THEN    $\text{put}(\langle \uparrow_0 \rangle(e_s \rightarrow t))$ 
        ELSE   ABORT
ELSE   ABORT

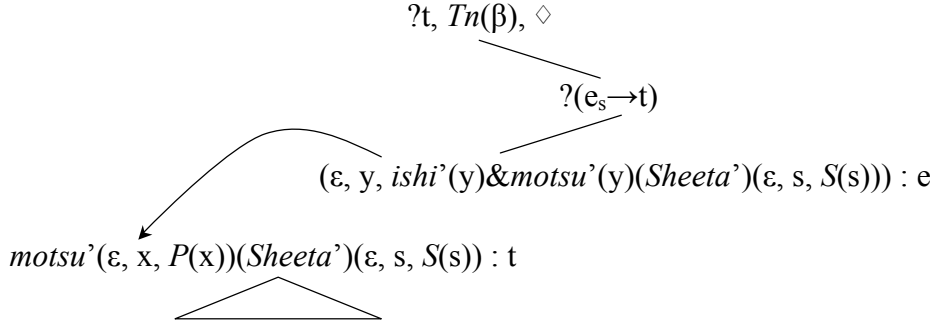
```

The two IF-blocks state that the current node must be a locally unfixed type- e node. In (3.57), the current node is fixed, but only in relation to the tree LINKed to it, and not to anything else, in particular not, as yet, to any containing structure. This issue has not been addressed in the extant DS literature. To rectify this problem, I propose that the case particle, if parsed at a node without any dominating nodes, constructs a partial

²⁸ Cann et al. (2005) and Kempson & Kurosawa (2009) assume that the node of the head noun is unfixed relative to the node set out by the AXIOM, but this unfixed relation is not encoded in the LINK INTRODUCTION rule in Cann et al. (2005) and Kempson & Kurosawa (2009). Even if it is encoded, there is another problem. As pointed out in Seraku (in press a), such a tree multiplies unfixed relations and leads to inconsistency in the node description; see the ‘Unique-unfixed-node Constraint’ in Sect. 4, Chap. 5.

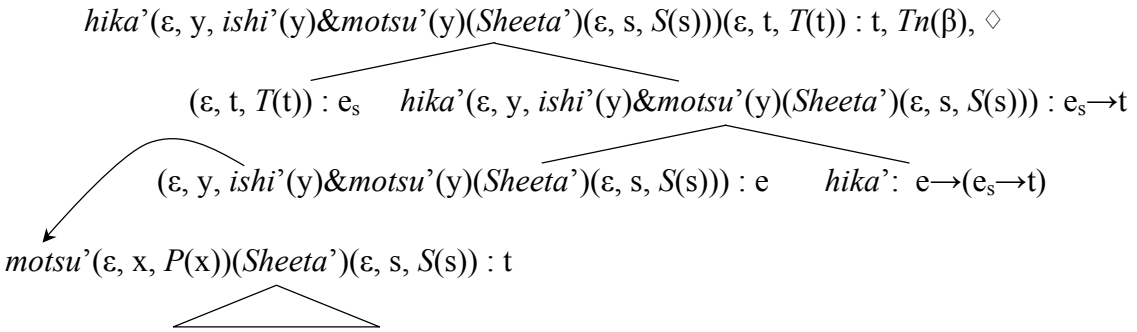
structure **bottom-up** until it reaches a type- t -requiring node, as in (3.60).²⁹ (The revised entry of *ga* will be given shortly.)

(3.60) Parsing [*Sheeta-ga motsu ishi*]-*ga*



Finally, the main tree is fleshed out by the matrix verb *hika-* (= ‘shine’), and ELIMINATION (i.e., functional application, type deduction) cleans up the tree. In the final output (3.61), the formula at the top node represents the propositional content ascribed to the head-external relative (3.52): ‘A stone which Sheeta holds shone.’

(3.61) Parsing [*Sheeta-ga motsu ishi*]-*ga hika-tta* + ELIMINATION (twice)



5.2.2 Case Particles Revisited

The last sub-section proposed a bottom-up structure building process triggered by case particles. This is formalised by revising the lexical entry of the nominative-case particle *ga* (3.59) as in (3.62), where the newly added part is within a square. The label $\langle \uparrow \rangle \perp$

²⁹ The top node is decorated with the tree-node address $Tn(\beta)$, where β is an arbitrary constant. Formally, this is to be expressed as $Tn(U)$ with a meta-variable U , together with the requirement $\exists x.Tn(x)$. See (3.62) in Sect. 5.2.2.

means that no node is present above the current node. If this condition is met, the parser constructs a schematic propositional structure bottom-up. Similarly, the entry of the accusative-case particle *o* is defined as (3.63).

(3.62) Entry of the nominative-case particle *ga* (penultimate version)

```

IF      e
THEN    IF      <↑0↑1*>(Tn(α), ?t)
        THEN    put(?<↑0>(es→t))
        ELSE    IF      <↑>⊥
                THEN    make/go(<↑0>); put(?es→t);
                        make/go(<↑1>); put(?t, Tn(U), ?∃x.Tn(x))
                ELSE    ABORT
ELSE    ABORT

```

(3.63) Entry of the accusative-case particle *o* (penultimate version)

```

IF      e
THEN    IF      <↑0↑1*>(Tn(α), ?t)
        THEN    put(?<↑0>(e→(es→t)))
        ELSE    IF      <↑>⊥
                THEN    make/go(<↑0>); put(?e→(es→t));
                        make/go(<↑1>); put(?es→t);
                        make/go(<↑1>); put(?t, Tn(U), ?∃x.Tn(x))
                ELSE    ABORT
ELSE    ABORT

```

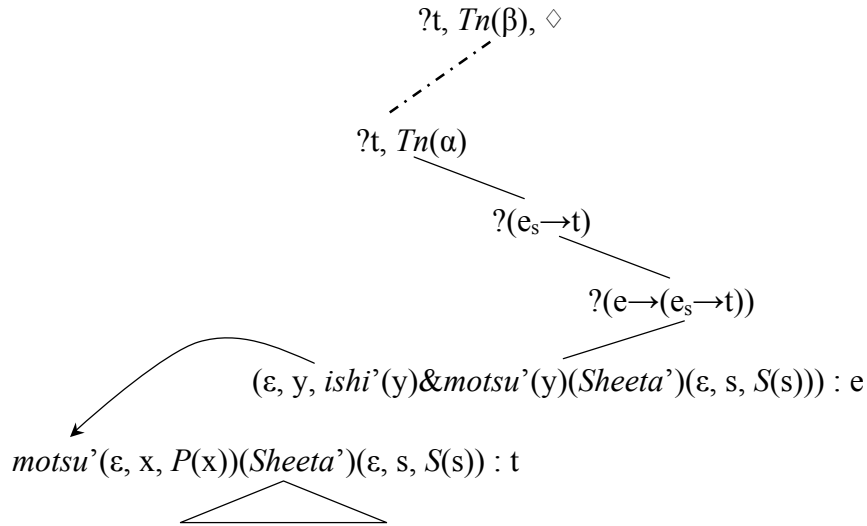
These entries are still in need of revision. According to (3.63), a structure is constructed bottom-up within a **local** domain, and long-distance dependencies cannot be accommodated. In (3.64), the noun phrase containing the relative clause is long-distance scrambled from the object position of the embedded verb *uba-* (= ‘take by force’).

(3.64) Long-distance scrambling of the head-external relative sequence

[[Sheeta-gamotsu] ishi]-o_i Pazu-ga
 [[S-NOM hold] stone]-ACC P-NOM
 [Muska-ga e_i uba-tta to] tsutae-ta.
 [M-NOM take.by.force-PAST COMP] tell-PAST
 ‘Pazu told that Muska took by force [a stone which Sheeta held].’

I assume that the case particle not only builds a structure bottom-up but also **optionally** builds a **higher** type-t-requiring node above the structure, as in (3.65). It is under this higher type-t-requiring node that the matrix subject *Pazu* will be processed.

(3.65) Parsing [*Sheeta-ga motsu ishi*]-o



In this tree display, the type-t-requiring unfixed node may be resolved **non-locally**, as shown by the dashed-dotted line (as in the case of *ADJUNCTION). This is because scrambling may occur across an arbitrary number of clausal boundaries. The modified lexical entries of the case particles are presented in (3.66)-(3.67). The added actions, *make/go*($\langle \uparrow_* \rangle$) and *put*($?t, Tn(V), ?\exists y.Tn(y)$), create a type-t-requiring node above the local structure. These actions are optional as expressed by parentheses; I assume that they are usually not performed, provided that long-distance scrambling is not common.

(3.66) Entry of the nominative-case particle *ga* (final version)

```

IF      e
THEN    IF       $\langle \uparrow_0 \uparrow_1 \rangle (Tn(\alpha), ?t)$ 
        THEN    put( $? \langle \uparrow_0 \rangle (e_s \rightarrow t)$ )
        ELSE    IF       $\langle \uparrow \rangle \perp$ 
                THEN    make/go( $\langle \uparrow_0 \rangle$ ); put( $? (e_s \rightarrow t)$ );
                        make/go( $\langle \uparrow_1 \rangle$ ); put( $?t, Tn(U), ?\exists x.Tn(x)$ );
                        (make/go( $\langle \uparrow_* \rangle$ ); put( $?t, Tn(V), ?\exists y.Tn(y)$ ))
                ELSE    ABORT
ELSE    ABORT

```

(3.67) Entry of the nominative-case particle *o* (final version)

```

IF      e
THEN    IF      <↑0↑1*>(Tn(α), ?t)
        THEN    put(?<↑0>(e→(es→t)))
        ELSE    IF      <↑>⊥
                THEN    make/go(<↑0>); put(?e→(es→t));
                        make/go(<↑1>); put(?es→t);
                        make/go(<↑1>); put(?t, Tn(U), ?∃x.Tn(x));
                        (make/go(<↑*>); put(?t, Tn(V), ?∃y.Tn(y)))
                ELSE    ABORT
ELSE    ABORT

```

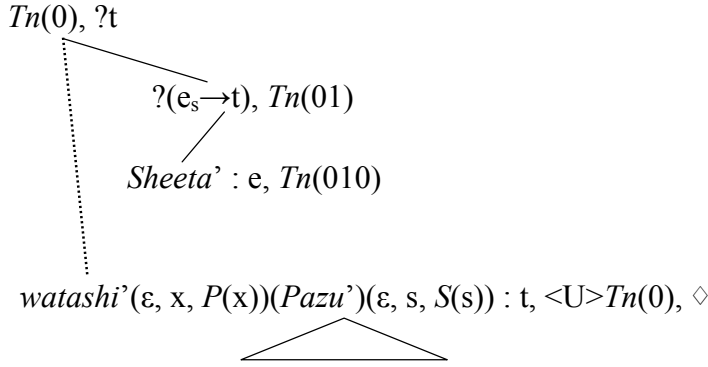
In the remainder of the thesis, I shall assume these entries of case particles (3.66)-(3.67). The bottom-up structure-building operation encoded in case particles is particularly important for modelling *no*-nominalisation (Sect. 3, Chap. 4).

5.2.3 Tree-Node Identification

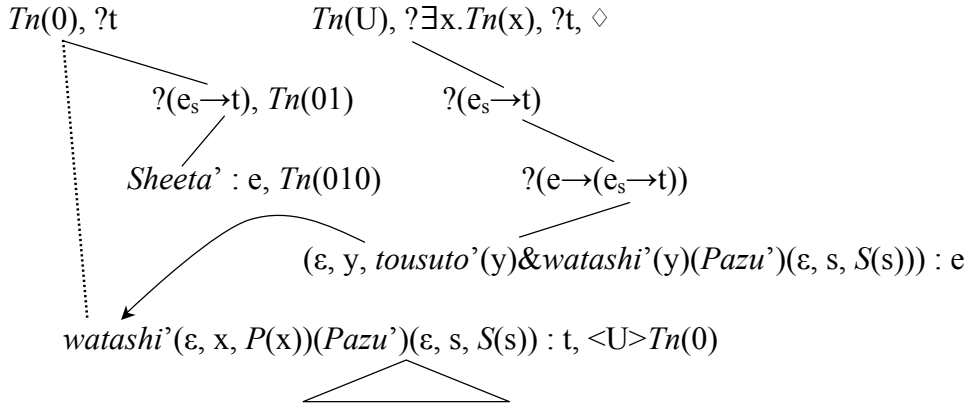
Consider (3.68), where the relative clause sequence is preceded by part of the matrix clause, *Sheeta-ga*. In this type of example, what matters is the identification of tree-node addresses, as we shall see below.

(3.68) *Sheeta-ga* [[Pazu-ga watashi-ta] tousuto]-o tabe-ta.
 S-NOM [[P-NOM hand-PAST] toast]-ACC eat-PAST
 ‘Sheeta ate a piece of toast which Pazu handed to her.’

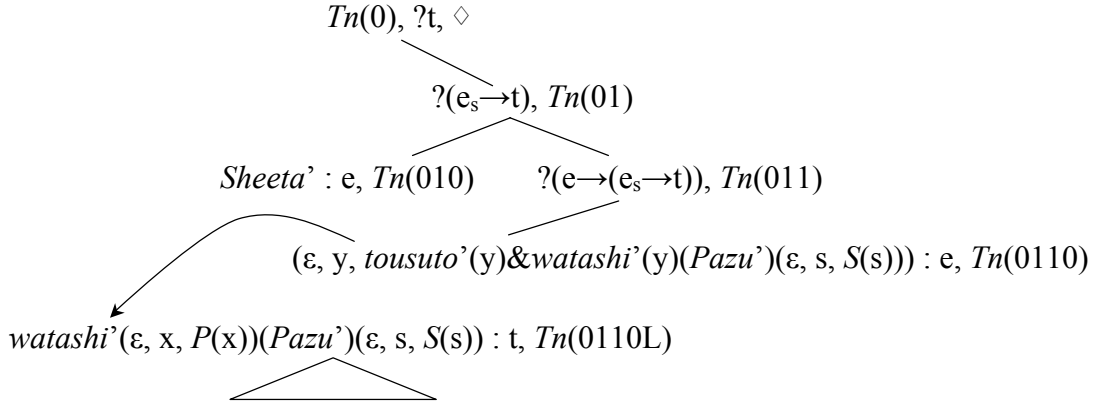
The parser scans *Sheeta-ga* under the type-t-requiring node specified by the AXIOM. Then, the parser runs GENERALISED ADJUNCTION to induce a type-t-requiring unfixed node, under which the relative clause is processed. (In (3.69), each node is explicitly decorated with a tree-node address; as we shall see, this decoration is essential for resolving the globally unfixed relation introduced by GENERALISED ADJUNCTION.)

(3.69) Parsing *Sheeta-ga Pazu-ga watashi-ta*

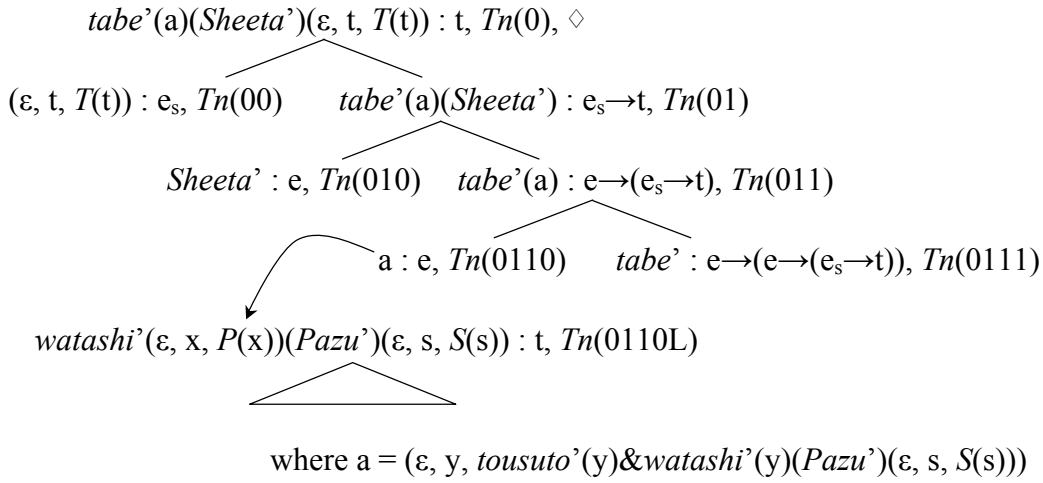
LINK INTRODUCTION introduces an inversely LINKed type-e-requiring node, which is decorated by the head noun *tousuto* (= ‘toast’). This node is enriched with the content of the relative clause by LINK ENRICHMENT, and is identified as an object node by the accusative-case particle *o*, which builds a partial structure bottom-up.

(3.70) Parsing *Sheeta-ga [Pazu-ga watashi-ta tousuto]-o*

The current node is decorated with the underspecified node-address $Tn(U)$. If the meta-variable U is saturated as 0, this node is **identified** with the initial node specified by the AXIOM; concomitantly, the two nodes annotated with $?(e_s \rightarrow t)$ are **identified** as one and the same node, bearing the same node-address $Tn(01)$. This process is what I call ‘tree-node identification’ (Seraku in press a).

(3.71) SUBSTITUTION (saturating U in $Tn(U)$ as 0)

The structure building is completed by the parse of the rest of the matrix clause. The final state is (3.72), where the content of the complex NP is abbreviated as a , which denotes an entity which is a piece of toast and which Pazu passed.

(3.72) Parsing *Sheeta-ga* [*Pazu-ga watashi-ta tousuto*]-o *tabe-ta*

The node-identification results in resolving the globally unfixed relation; in (3.71), the dotted line has disappeared since the type- t unfixed node has been fixed as a top node of the adjunct tree. This has a positive theoretical consequence, since past DS works have not explicated how the globally unfixed relation might be fixed (Cann et al. 2005, Kempson & Kurosawa 2009, Kempson et al. 2011).

5.2.4 Novelties

The LINK transition has been illustrated by analysing the head-external relative. The analysis follows Kempson & Kurosawa (2009) in defining LINK INTRODUCTION and LINK ENRICHMENT but it advances the previous account. First, Sect. 5.2.1 showed that the relative clause part was **directly** parsable under the initial node set out by the AXIOM because the initial node may be a non-root (Seraku in press a). Second, Sect. 5.2.2 enriched the lexical entry of case particles with the **bottom-up** process. This eschews the multiplication of unfixed nodes, a formal problem of the previous analyses (Cann et al. 2005, Kempson & Kurosawa 2009); see footnote 28. Finally, Sect. 5.2.3 introduced the **tree-node identification** process. This operation explicates how globally unfixed relations are resolved during gradual tree growth.

5.3 Topic Constructions

Another construction that is characterisable by means of LINK transitions is the *wa*-topic construction in Japanese (3.73).

(3.73) The *wa*-topic construction

Pazu-wa hashi-tta.

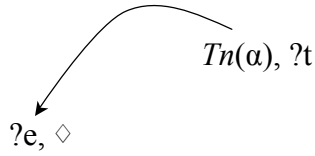
P-TOP run-PAST

‘Pazu ran.’

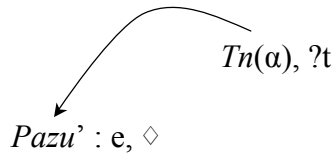
As in the case of head-external relatives, the LINK transition proceeds in two steps: the introduction of a LINK relation and the enrichment of the LINK relation.

Firstly, after the AXIOM sets out an initial node, the parser induces a LINK relation to a type-e requiring node. This tree transition is formalised as the general action LINK INTRODUCTION (3.74).

(3.74) LINK INTRODUCTION

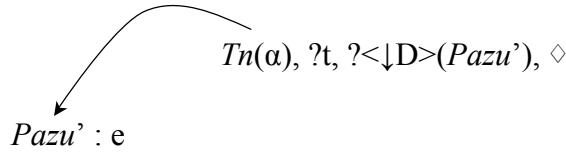


The current type-e-requiring node is decorated by the initial item *Pazu*. This updates the tree (3.74) into (3.75).

(3.75) Parsing *Pazu*

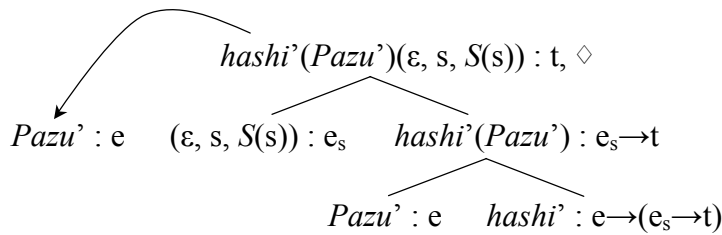
Second, the LINK relation is enriched. In head-external relatives, this was achieved by the general action LINK ENRICHMENT. In the topic construction (3.73), however, this action is encoded in the topic particle *wa*. That is, the pointer $\diamond$ returns to the propositional node, and a requirement is posited that a node somewhere below (excluding the existing LINKed node) will be decorated with the content of *Pazu*. This is expressed as $?<\downarrow D>(Pazu')$. $<D>(\alpha)$ declares that α holds **somewhere below** (i.e., allowing a crossing of a LINK boundary). Therefore, if the requirement were simply $?<D>(Pazu')$, it would be trivially satisfied by the existing LINKed node decorated with *Pazu'*. To prevent this trivial satisfaction, another operator $\downarrow$, which **cannot** cross a LINK boundary, is added. The entry of *wa* is given in (3.77), where β is instantiated as *Pazu'* in the current example.³⁰

³⁰ For more complex (but not directly pertinent) issues of *wa*, see Cann et al. (2005: 268). Also, note that the operator $<D>$ governs the weakest relation. This has an implication for the issue of syntactic islands. In the current example (3.73), which does not involve an island, the term *Pazu'* will decorate a node in the emergent structure, as shown in (3.78). But if a string contains an island, the content of a *wa*-marked item may be located within an island, or in DS terms, within a LINKed structure due to the weakest relation allowed by $<D>$. This correctly models that *wa*-marked items are not sensitive to islands (Kuno 1973a).

(3.76) Parsing *Pazu-wa*(3.77) Entry of the topic marker *wa* (preliminary version)

IF $\beta : e$
 THEN IF $\langle L^{-1} \rangle(?t)$
 THEN $go(\langle L^{-1} \rangle); put(? \langle \downarrow D \rangle(\beta))$
 ELSE ABORT
 ELSE ABORT

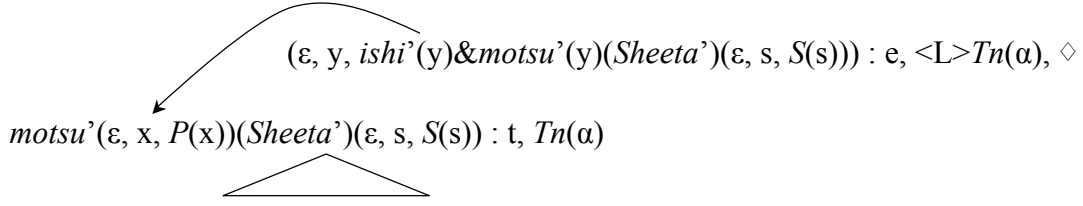
After this two-stage process of LINK transition, the rest of the tree update is as usual. The verb *hashi-* (= ‘run’) projects a propositional schema with two slots. A meta-variable at a situation node is substituted with $(\epsilon, s, S(s))$. A meta-variable at a subject node is substituted with *Pazu'*, given that the current tree is constructed with respect to the content *Pazu'*. This substitution process meets $? \langle \downarrow D \rangle(Pazu')$. The output is (3.78), where the proposition at the top node represents the interpretation of (3.73): ‘Pazu ran.’

(3.78) Parsing *Pazu-wa hashi-tta*

The topic marker *wa* may be attached to a complex NP as in (3.79). The parse of this string up to the head noun *ishi* (= ‘stone’) plus LINK ENRICHMENT leads to the tree (3.80). (This tree is exactly the same as (3.57) in Sect. 5.2.1.)

(3.79) The *wa*-topic construction (involving the head-external relative)

[[Sheeta-gamotsu] *ishi*]-*wa* *hika-tta*.
 [[S-NOM hold] stone]-TOP shine-PAST
 ‘A stone which Sheeta holds shone.’

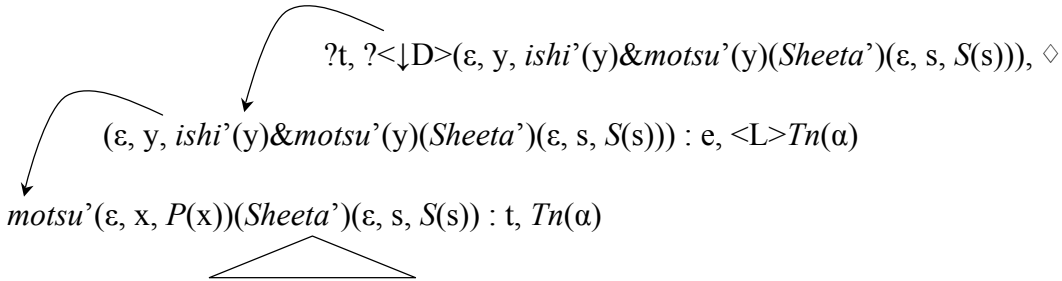
(3.80) Parsing *Sheeta-ga motsu ishi* + LINK ENRICHMENT

Here, the same sort of problem arises as the parsing of case particles (Sect. 5.2.2). The topic marker *wa* cannot be processed at the current node in (3.80). According to the input condition in (3.77), the current node must be decorated with a type-e term and there must be an **inversely LINKed type-t-requiring** node. In (3.80), however, there is no inversely LINKed type-t-requiring node.

To solve the problem, I modify the entry (3.77) as (3.81). The condition on the inversely LINKed type-t-requiring node is dropped. Instead, **make/go**($\langle L^{-1} \rangle$) **builds** an inversely LINKed node, to be decorated with ?t and ? $\langle \downarrow D \rangle(\beta)$.³¹ With these actions, the tree (3.80) is updated into (3.82), where β is instantiated as the content of the complex NP, $(\epsilon, y, ishi'(y) \& motsu'(y)(Sheeta')(\epsilon, s, S(s)))$.

(3.81) Entry of the topic marker *wa* (final version)

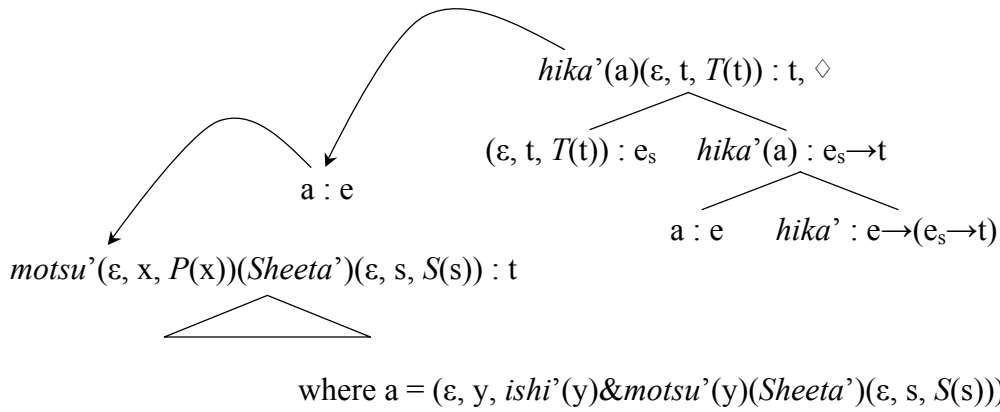
IF $\beta : e$
 THEN **make/go**($\langle L^{-1} \rangle$); **put**(?t, ? $\langle \downarrow D \rangle(\beta)$)
 ELSE ABORT

(3.82) Parsing [*Sheeta-ga motsu ishi*]-*wa*

³¹ This action is performed even when there is already an inverse LINK relation as in (3.75). In such cases, the two inverse LINK relations collapse harmlessly.

The remainder of the process is as usual. The verb *hika-* (= ‘shine’) fleshes out the current type-*t*-requiring node. In particular, the object meta-variable is substituted with $(\epsilon, y, \text{ishi}'(y) \& \text{motsu}'(y)(\text{Sheeta}')(\epsilon, s, S(s)))$, satisfying the requirement on a shared content. The final output is given in (3.83). The formula at the top node represents the interpretation of (3.79): ‘A stone which Sheeta holds shone.’

(3.83) Parsing [*Sheeta-ga motsu ishi*]-*wa hika-tta*



The entry of the topic marker *wa* (3.81) is pertinent to the discussion of the *wa*-marked instances of *no*-nominalisation (Sect. 3.4, Chap. 4) and particularly to the analysis of cleft constructions (Chap. 5).

In closing, let us consider the notion of topic. In general, topic is old or given information with respect to which the non-topicalised part of the sentence is construed (Erteschik-Shir 2007, Lambrecht 1996). In some frameworks, topic is expressed over a syntactic or other level of representation. For instance, the cartographic approach to the left periphery (Rizzi 1997) postulates a syntactic projection for topic, TopP, whereas Lexical Functional Grammar defines the attribute TOPIC over i(nformation)-structure (Butt & King 2000: 12). By contrast, DS does **not** posit a primitive concept of topic; the topic effect emerges as a by-product of incremental processing (Cann et al. 2005: 183-4, Kempson et al. 2006). More specifically, topic effects arise when a structure is set as a context against which the parser builds up a propositional structure. In our current cases, the topic marker *wa* encodes an instruction to introduce an inverse LINK relation to a

new propositional node and posit a requirement that the node will be developed with the LINKed tree as context. This captures the topic effect associated with the *wa*-marked topic construction. See Chap. 5 (Sect. 3.4.4) for the analysis of topic effects in clefts.

5.4 Summary

LINK is a formal device pairing two discrete trees through a shared type-e element. A LINK transition consists of the introduction of a LINK relation and the enrichment of the LINK relation. These processes are illustrated by the head-external relative and by the topic construction.

In the head-external relative, the parser builds a tree for the relative clause and then LINKs it to a fresh type-e-requiring node (LINK INTRODUCTION). After this fresh node is decorated by the head noun, the content of the relative clause is incorporated into the node (LINK ENRICHMENT). As a result of this second process, the composite term denotes an individual that reflects the contents of the head noun and the relative clause. When a relative clause is sandwiched by parts of a matrix clause, the tree-node identification is essential (Seraku in press a), which resolves a globally unfixed node introduced by GENERALISED ADJUNCTION. In this course of a LINK transition, the tree update remains monotonic in processing the relative clause sequence left-to-right.

Concomitantly, the entry of case particles was endowed with the bottom-up structure-building action. This is because the node of the head noun does not stand with respect to a propositional node after the application of LINK ENRICHMENT. In this structural environment, the case particle constructs a partial structure bottom-up (and optionally builds a higher type-t-requiring node on top of the partial structure). This partial structure is, then, fleshed out by the embedding clause.

In the topic construction, both the introduction and the enrichment of a LINK relation are formalised as the lexical actions encoded in the topic marker *wa*. (Still, the general action LINK INTRODUCTION (3.74) is maintained because it is essential for the

examples where the initial expression is *wa*-marked as in (3.73).) The topicalised item is parsed at a LINKed type-e-requiring node, and the main tree will be subsequently built up with the LINKed node as context. This LINK transition captures the topic effect as an epiphenomenon of incremental processing.

The LINK machinery is vital to the analysis of *no*-involving constructions to be investigated in this thesis: *no*-nominalisation (Chap. 4), cleft constructions (Chap. 5), and several types of relative clause constructions (Chap. 6).

6 Dynamic Construal of Context

Context within DS reflects not only a set of previous **trees** but also a set of **actions** that update those trees (e.g., Cann et al. 2007, Eshghi et al. 2011, Kempson et al. to appear, Purver et al. 2006). In this conception, context becomes increasingly rich, reflecting (i) the linear parsing, (ii) attendant tree growth, and (iii) the actions that contributed to such tree growth.³²

The inclusion of both structures and actions in context bears significance for ellipsis, bringing new light to capture the strict/sloppy ambiguity. Consider the strict and the sloppy readings of the English VP ellipsis case (3.84).

(3.84) Pazu respects his mother. Duffy does too.

- a. ‘Pazu_i respects his_i mother. Duffy_j respects his_i mother too.’ (strict reading)
- b. ‘Pazu_i respects his_i mother. Duffy_j respects his_j mother too.’ (sloppy reading)

Two modes of resolving ellipsis are adopted: re-use of structure and re-use of actions. Firstly, the parser may copy the content of the antecedent (i.e., ‘respect Pazu’s mother’) onto the ellipsis site (e.g., the node decorated by *does*). In this case, the elided part is interpreted identically to the antecedent part, yielding a strict reading. The re-use of

³² Formally, context is defined on the basis of a parser tuple $\langle T, W, A \rangle$, where T is a (possibly, partial) tree, W is a sequence of words so far processed, and A is a sequence of actions that have fired (Cann et al. 2007, Purver et al. 2006). See the appendix for details.

structure is formulated as the pragmatic action SUBSTITUTION. Secondly, another means of resolving ellipsis is to select a sequence of previous actions to build the antecedent structure (e.g., actions encoded in *respects*, *his*, and *mother*) and to reiterate it at the ellipsis site. Uncertainties that arise in this reiteration are updated with respect to the emergent structure; in particular, a meta-variable imposed by *his* is saturated relative to the node for *Duffy* (rather than *Pazu*), hence a sloppy reading. The re-run of actions is formulated as the pragmatic action REGENERATION.

In a nutshell, ellipsis ambiguity sheds light on the nature of context: the parser may directly carry over a previous content to an ellipsis site (i.e., strict reading) or re-run a set of previous actions and enrich an ellipsis site relative to the unfolding structure (i.e., sloppy reading). This dynamic conception of context is important to discussion of clefts, stripping, and sluicing in Chap. 5, where the copula *da-* posits a propositional meta-variable and licenses the re-use of actions stored in context.

7 Coda: Cross-Theory Comparisons

In closing this DS exegesis, let us step back and compare the DS formalism with contemporary syntactic theories. Arguably, the most widespread theory at present is what Culicover & Jackendoff (2005) dub MGG (Mainstream Generative Grammar), a syntactic theory that has been developed by Noam Chomsky in a series of works (e.g., 1957, 1965, 1981, 1995); see Graff & van Urk (2012) for recent works. As common ground, MGG and DS aim at modelling **knowledge** of language rather than use of language. Furthermore, MGG and DS present **generative** grammar(s) in the sense of Chomsky (1965).³³ Despite the parallelisms, there are crucial differences between these two theoretical paradigms.

³³ The term ‘generative’ is to be understood to mean ‘explicit’ (Chomsky 1965). In connection with this, an important issue is whether DS is a ‘generative-enumerative’ system or a ‘model-theoretic’ system (Pullum & Scholz 2001). Simply put, a model-theoretic system is a monotonic framework of declarative constraints (Sag & Wasow 2011). Given that DS consists of both the

First of all, the DS concept of competence is defined as **directly** reflecting language performance: competence is a set of procedures to build a context-dependent interpretation on the basis of left-to-right incremental parsing. This is contrasted with the MGG notion of competence, which is generally thought of as a system of principles and rules to assign a legitimate structure to a string in a static manner, abstracted from the way they are put to use in dialogue and discourse.

DS and MGG are also contrasted in terms of the architectural design of a grammar. DS is a **constraint-based, lexicalist** theory. Unlike MGG (but like Lexical Functional Grammar (LFG) (Bresnan 2001, Dalrymple 2001) and Head-driven Phrase Structure Grammar (HPSG) (Sag et al. 2003)), a grammar is articulated in terms of constraint satisfaction rather than a series of derivations based on transformations (or the equivalents). In particular, DS dispenses with ‘movement’ operations and accounts for displacement phenomena in terms of how various forms of requirements regarding structural underspecification are satisfied. LFG, HPSG, and DS are all constraint-based theories, but DS differs from these others in that each lexical item not only provides constraints on a tree structure but also **triggers** the tree-building process through a set of requirements. Within LFG, a set of f-descriptions constructs f-structure; in this sense, they may well be said to build up a structure. But the role of the tree-building process is the heart of DS tree update.

DS is a **mono-level** theory (phonology apart). Unlike MGG (and unlike LFG (Asudeh 2006) and HPSG³⁴), DS postulates only semantic representation **without** an independent level of syntactic structure. Within DS, there is no syntax over and above a set of procedures to update a semantic tree incrementally. Various phenomena that have been treated in MGG are analysed in terms of how an interpretation is progressively

declarative part and the procedural part, it is not clear how DS is characterised in this regard. I shall leave this issue for future research.

³⁴ Although HPSG represents various types of information uniformly in the Attribute-Value Matrix notation, syntactic, semantic, and pragmatic information are expressed separately, with wholly independent vocabulary (Ginzburg & Cooper 2004, Sag et al. 2003).

built. Another key feature of DS representation is that it models a **context-dependent** interpretation. Thus, the DS architecture is not encapsulated in Fodor's (1983) sense, and pragmatic inference plays a role during DS computations (cf., Wedgwood 2005).

Finally, DS is a '**grammar as parser**' theory. Unlike most works within MGG (and also within HPSG and LFG), the grammar *per se* adopts the dynamics of an online parser: the DS grammar/parser defines the mechanisms available for building structured interpretations that a string in context may be mapped onto, with the resolution of local ambiguities in incremental processing relegated to various 'search strategies' developed in natural language processing research (Sato 2011). This is in contrast with MGG, where it is generally assumed that there is a parser to complement a core grammar; see Frazier & Fodor (1978) and Kimbal (1973) for early work on 'perceptual strategies' of the parser, and Mulders (2005) and Pritchett (1992) for the parser that is based on core-grammar principles. The DS grammar-parser has the following features. First, it is strictly **incremental**: a tree is built up as the surface string is parsed left-to-right, with empty items being generally **not** hypothesised. Second, DS tree-update is **monotonic**. In this respect, DS clearly contrasts with Phillips (2003), who proposes an incremental grammar within MGG, where the mechanism is one of 'build and revise.' Third, DS structure building is generally **top-down** but also equipped with the **bottom-up** process. Fourth, tree growth is **predictive**, with a requirement restricting a set of actions to be executed. Sato (2011: 213-4) points out that this predictivity holds even if no node has been instantiated since the initial state specified by the AXIOM comes with requirements.

The upshot is that within DS, competence is regarded as a set of constraints on performance, where competence is more transparently dependent on performance than MGG assumes. The DS conception of competence is a set of procedures to construct a context-dependent interpretation progressively and monotonically on the basis of word-by-word, left-to-right parsing. In this light, the DS notion of 'dynamics' differs from 'dynamics' in some dynamic semantics theories; see Kamp & Reyle (1993), where

discourse representation structure is updated sentence-by-sentence rather than word-by-word. In short, DS is a model of the incremental processing dynamics, with this parsing device being at the heart of natural language grammars.

Chapter 4

The Dynamics of *No*-Nominalisation

1 Overview

Against the empirical foundation (Chap. 2) and the theoretical backbone (Chap. 3), this and subsequent chapters provide an account of *no*-involving constructions from the angle of language dynamics. In the present chapter, I shall articulate an account of the particle *no* itself by surveying *no*-nominalisation within the framework of Dynamic Syntax (DS) (Cann et al. 2005, Kempson et al. 2001, Kempson et al. 2011).

Nominalisation is a morphological or grammatical process through which a non-nominal element becomes a nominal element. One way of classifying it is the divide between lexical nominalisation as in (4.1) and grammatical nominalisation as in (4.2) (Shibatani 2009: 187, Shibatani & Bin Makhashen 2009: 22).

(4.1) employ → employer, employee

(4.2) Kanta runs. → Kanta's running

Lexical nominalisation is a morphological process, the output (e.g., *employer*) being usually registered in the lexicon. Grammatical nominalisation is a syntactic process, the output (e.g., *Kanta's running*) being usually not registered in the lexicon.

The particle *no* yields two types of grammatical nominalisation: participant nominalisation such as (4.3) and situation nominalisation such as (4.4) and (4.5).¹

(4.3) Participant nominalisation

Mei-wa [Chibi-Totoro-ga otoshi-ta no]-o hiro-tta.
 M-TOP [CT-NOM drop-PAST NO]-ACC pick.up-PAST
 ‘Mei picked up a thing which Chibi-Totoro dropped.’

(4.4) Situation nominalisation

Mei-wa [Nekobasu-ga hashiru no]-o mi-ta.
 M-TOP [N-NOM run NO]-ACC see-PAST
 ‘Mei saw an event where Nekobasu was running.’

(4.5) Situation nominalisation

Mei-wa [Totoro-ga donguri-o atsumeteiru no]-o shitteiru.
 M-TOP [T-NOM acorn-ACC collect NO]-ACC know
 ‘Mei knows that Totoro collects acorns.’

In participant nominalisation, *no* turns the pre-*no* clause into a nominal that denotes an **individual** (e.g., persons, objects), that is, a participant of a situation described by the pre-*no* clause. In (4.3), the *no*-part denotes an object which Chibi-Totoro dropped. In situation nominalisation, *no* turns the pre-*no* clause into a nominal that denotes a **situational** entity (e.g., actions, events, propositions) described by the pre-*no* clause. The *no*-part in (4.4) denotes an event where Nekobasu was running, and the *no*-part in (4.5) denotes a proposition that Totoro collects acorns.

¹ One may wonder whether the strings in (4.4)-(4.5) could be regarded as examples of ‘head-internal relatives,’ a relative construction where a head noun is found inside the relative clause and the relative clause is followed by the particle *no* (Chap. 6). The strings in (4.4)-(4.5), in fact, match the surface description of head-internal relatives. For instance, if a head-internal relative construal were possible, the string in (4.5) could mean ‘Totoro collects acorns, and Mei knows Totoro.’ But head-internal relative readings are absent from (4.4)-(4.5). This is because head-internal relatives impose interpretive conditions on the relation between the content of the relative clause and that of the embedding clause, and these conditions are not met in (4.4)-(4.5). See the discussion of the Relevancy Condition (Kuroda 1992: 147) in Sect. 2.2, Chap. 6.

In the literature, some attempts have been made to deal with these two types of nominalisation in a uniform fashion (Kitagawa 2005, Kitagawa & Ross 1982, Murasugi 1991, Shibatani 2009, Tonoike 1990). Still, no detailed account has been proposed. Almost all of these studies are conducted within Mainstream Generative Grammar. In this theoretical camp, a lexical item is seen as a feature bundle, and two lexical items at different nodes are generally assigned different features. Thus, a challenge is how to differentiate the two types of *no*-nominalisation structurally without assigning different features to two *nos*. This challenge may be sidestepped by the ‘*no*-insertion’ rule in Kitagawa (2005), which inserts *no* under a certain syntactic condition. But there is another issue. As will be observed in Sect. 4, when clause coordination is involved, the two types of *no*-nominalisation exhibit a certain asymmetry, and it is not obvious whether and how previous studies handle this asymmetry without losing uniformity in explanation. The goal of this chapter is twofold: (i) to justify an approach integrating the two types of *no*-nominalisation and (ii) to develop and defend such an account in the light of language dynamics.

Sect. 2 offers various arguments (e.g., diachronic, cross-linguistic) for a unitary treatment of *no* (Seraku in press c). Sect. 3 constructs a dynamic account to unify the phenomenon: the two types of *no*-nominalisation boil down to a parser’s choice of what term it copies in processing *no*. This account predicts a number of *no*-nominalisation properties, including the corpus findings in Chap. 2 (Seraku 2012b). Sect. 4 extends the analysis to new data. It is observed that when coordinate clauses are nominalised by *no*, the two types of *no*-nominalisation behave distinctly. This discrepancy follows from our account if the issues of ‘clausal coordination’ and ‘tense’ are given a Dynamic Syntax form of analysis. Finally, Sect. 5 concludes the chapter.

2 Motivating a Unitary Analysis

2.1 Introduction

This section offers several reasons why the two types of *no*-nominalisation should be treated in a **unitary** fashion. That is, it will be clarified why it is better to assume that there is only a single item *no* than to assume that there are two different *nos*.

A complicating issue is that the particle *no* has other functions too, the most widely known of which is the genitive.² In (4.6), *no* expresses a relation between Mei and maize. For instance, the relation may be an ownership relation (i.e., Mei possesses that maize), a fondness relation (i.e., Mei loves that maize), etc.³

(4.6) Genitive *no*

Yasuko-wa Mei-no-toumorokoshi-o tabeta.

Y-TOP M-GEN-maize-ACC ate

‘Yasuko ate Mei’s maize.’

One important question is whether the genitive function of *no* is also to be treated uniformly. In what follows, it will be argued that the genitive *no* in (4.6) should be modelled separately from *no* in *no*-nominalisation. My contention, then, is that there are two items whose form is *no*, one of which appears in *no*-nominalisation and the other of which appears in the genitive construction. The connections between form and function are displayed in Figure 4.1.

² As another usage of *no*, some speakers accept a complementiser use of *no* in a certain type of relative clause, where *no* appears between the adnominal predicate and the element modified. For instance, Frellesvig & Whitman (2011: 77) cites Soga & Fujimura’s (1978: 41) example (i). This thesis sets aside these data due to fluctuating judgments across speakers.

(i) aisuru no amari
love NO excess
‘(due to) an excess of love’

³ In (4.6), the genitive *no* expresses a certain relation between the two NPs. In other examples, the genitive *no* has different functions (e.g., argument-marking function). See Nishiyama (2003) for a meticulous classification of diverse functions of the genitive *no*.

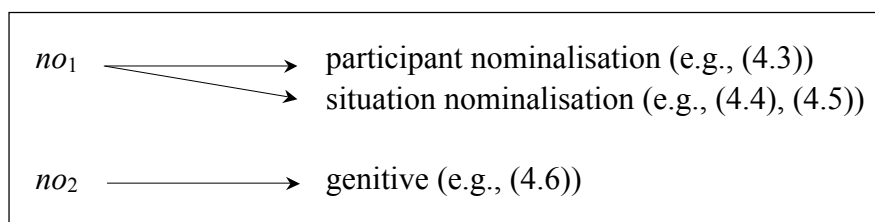


Figure 4.1. Form-function correspondences

Several motivations will be provided for a uniform treatment of *no*-nominalisation (excluding the genitive *no*). For illustration purposes, I shall refer to (4.3), repeated here as (4.7), as a representative example of participant nominalisation, and refer to (4.4), repeated here as (4.8), as a representative example of situation nominalisation.

(4.7) Participant nominalisation

Mei-wa [Chibi-Totoro-ga otoshi-ta no]-o hiro-tta.
 M-TOP [CT-NOM drop-PAST NO]-ACC pick.up-PAST
 ‘Mei picked up a thing which Chibi-Totoro dropped.’

(4.8) Situation nominalisation

Mei-wa [Nekobasu-ga hashiru no]-o mi-ta.
 M-TOP [N-NOM run NO]-ACC see-PAST
 ‘Mei saw an event where Nekobasu was running.’

2.2 Methodological Considerations

A central issue in semantics and pragmatics is whether an attested meaning is lexically encoded (a matter of semantics) or inferentially derived (a matter of pragmatics). One methodological principle is Modified Occam’s Razor: other things being equal, ‘senses are not to be multiplied beyond necessity’ (Grice 1978: 118-9). That is, if there are multiple attested meanings of a given form, and if one of those meanings is the basis for inferring all others, then multiple lexemes should not be posited, other things being equal. The underlying idea is that pragmatic inference is governed by pragmatic principles, which are independently motivated for, say, deriving implicatures, and that a

pragmatic, non-ambiguity account is simpler (hence, favourable) than a semantic, ambiguity account.

Given this principle of theoretical parsimony, a unitary analysis of the three functions of *no* (i.e., participant nominalisation, situation nominalisation, genitive) is preferable, other things being equal. As will be shown below, however, ‘other things’ are not equal if attention is paid to other facets of the phenomena.

2.3 Functional Considerations

The particle *no* in participant nominalisation (4.7) and situation nominalisation (4.8) serves as a nominaliser, whereas *no* in the genitive construction (4.6) does not. As for the genitive *no* in (4.6), nominalisation does not occur because what precedes *no* is the noun *Mei* in the first place. As for (4.7) and (4.8), there are several pieces of evidence that indicate that nominalisation has taken place.

First, nominalisation is recognised by the presence of ‘noun phrase markers’ (Comrie & Thompson 2007: 353). One such marker is a case particle. As already evinced, the accusative-case particle *o* is attached to *no* in both (4.7) and (4.8).

Second, the predicates *otoshi-* (= ‘drop’) and *hashiru-* (= ‘run’) in (4.7) and (4.8) are conjugated as an ‘ad-nominal’ form (rather than a ‘conclusive’ form, which is used to end a clause). At first sight, it is not easy to tell that the predicates in (4.7) and (4.8) are in an ad-nominal form because verbs in Contemporary Japanese have lost the morphological distinction between the conclusive form and the ad-nominal form (Frellesvig 2010). Still, adjectival verbs maintain the distinction, like the conclusive form *kireida-* (= ‘beautiful’) and the ad-nominal form *kireina-*. If this adjectival verb is used in the two types of nominalisation, only the ad-nominal form is grammatical. This pattern is expected if we assume the *no*-part is nominalised.

(4.9) Participant nominalisation

Mei-wa [kireina/*kireida no]-o hirotta.
 M-TOP [beautiful NO]-ACC picked.up
 ‘Mei picked up a beautiful thing.’

(4.10) Situation nominalisation

Mei-wa [donguri-ga kireina/*kireida no]-o shitteiru.
 M-TOP [acorn-NOM beautiful NO]-ACC know
 ‘Mei knows that acorns are beautiful.’

Finally, if a subject is in a nominal-modifying clause (e.g., relative clause), it may be marked by the genitive-case particle *no*, as well as the nominative-case particle *ga* (so-called ‘Nominative/Genitive Conversion’ (Harada 1971)). Thus, the nominative-case particle *ga* may be replaced with the genitive-case particle *no* in both participant and situation nominalisation (4.7) and (4.8), as in (4.11) and (4.12).

(4.11) Participant nominalisation

Mei-wa [Chibi-Totoro-ga/no otoshita no]-o hirotta.
 M-TOP [CT-NOM/GEN dropped NO]-ACC picked.up
 ‘Mei picked up a thing which Chibi-Totoro dropped.’

(4.12) Situation nominalisation

Mei-wa [Nekobasu-ga/no hashiru no]-o mi-ta.
 M-TOP [N-NOM/GEN run NO]-ACC see-PAST
 ‘Mei saw an event where Nekobasu was running.’

I do not present an account of why case alternation can happen in these contexts, but the above data indicate that the *no*-part in (4.11) and (4.12) is nominalised.

In sum, the instances of *no* in (4.7) and (4.8), but not in the genitive (4.6), share the nominalising function. This meshes well with Figure 4.1: there is a single item *no* which encodes the two types of nominalising functions, and there is another, distinct item *no* which encodes the genitive function.

2.4 Diachronic Considerations

A widespread supposition in Japanese diachronic linguistics is that *no* in situation nominalisation is derived from *no* in participant nominalisation (Horie 1998, Kinsui 1995, Nishina & Yoshimura 2005, Yanagida 1993, Yap et al. 2004, Yap et al. 2011, Yap & Matthews 2008, Yoshimura 2005, 2010).

A problem for most previous studies is that no distinction is explicitly made between participant nominalisation and the ‘NP + *no*’ construction (4.13); for the ‘NP + *no*’ construction, see Saito & Murasugi (1990) and Saito et al. (2008).

(4.13) The ‘NP + *no*’ construction

Yasuko-wa Mei-no-o tabeta.

Y-TOP M-NO-ACC ate

‘Yasuko ate Mei’s.’

There are reasons for distinguishing between *no* in participant nominalisation and *no* in ‘NP + *no*.’ First, there is a functional difference: only the former is an instance of nominalisation. In the latter case, what precedes *no* is an NP, which itself is already a nominal. Second, Wrona (2012) reports that there is a time gap of more than 500 years between the first attested occurrence of the former (about 17th century) and that of the latter (about 12th century or earlier); see also Shibatani (2013).⁴

Contrary to the prevalent view that *no* in situation nominalisation was derived from *no* in participant nominalisation, Wrona (2012) argues that once the nominalising use of the single item *no* became entrenched, the participant/situation nominalising functions arose at the same time. This view is historically justified: there are no diachronically attestable precedence relations between *no* in participant nominalisation and *no* in situation nominalisation (Nishi 2006). Moreover, the early instances of *no*-

⁴ It should be stressed that I do not deny the possibility that *no* in ‘NP + *no*’ paved the way for the emergence of *no* in participant nominalisation; what I would like to claim is that these two *nos* constitute separate items.

nominalisation are often ambiguous between participant and situation nominalisation. Consider (4.14), adapted from Wrona (2012: 209).

- (4.14) Ambiguity case [Early Modern Japanese (*Ukiyoburo*, 1809)]
 [Kure-ni kafu-no]-o wasure-ta kara
 [evening-in buy-NO]-ACC forget-PAST because
 a. ‘Because I forgot the one I was going to buy in the evening, ...’
 b. ‘Because I forgot that I bought something in the evening, ...’

As long as Wrona’s (2012) claim is on the right track, a uniform approach to *no* in participant and situation nominalisation is bolstered. As for the genitive *no* like (4.6), it had appeared in Old Japanese (8th century). Thus, while it is likely that the genitive *no* impinged upon the emergence of *no* in participant and situation nominalisation, it would be historically licit to assume that the genitive *no* forms a separate lexical item. These motivate my contention in Figure 4.1, a uniform account of *no* that encompasses the participant and situation nominalising functions (but not the genitive function).

2.5 Cross-Linguistic Considerations

According to Lyons (1977: 443), there are three distinct ontological statuses for classification of entities: ‘first-order’ entities denote individuals such as persons and objects, ‘second-order’ entities denote actions and events, and ‘third-order’ entities denote propositions. As pointed out in Yap et al. (2011: 3), nominalisation may be cross-linguistically classified in terms of this ontological distinction. For instance, the first-order entity nominalisation is illustrated by Cantonese in (4.15), and the second-order entity nominalisation is illustrated by Toqabaqita (Oceanic) in (4.16).

- (4.15) The first-order entity nominalisation [Cantonese]
 Daai mou ge ho ji jap nei.
 wear hat NMNZ can enter
 ‘The ones who wear hats may enter.’ (Yap et al. 2011: 3)⁵
- (4.16) The second-order entity nominalisation [Toqabaqita]
 Fasi-laa qoe qana baqu qena ki
 plant-NMNZ 2SG GENP banana that PL
 ‘Your planting (of) those bananas.’ (Lichtenberk 2011: 703)

No-nominalisation in Japanese may be classified in this fashion (Horie 1998: 174). In participant nominalisation (4.7), repeated here as (4.17), the *no*-part denotes a first-order entity: an object which was dropped by Chibi-Totoro.

- (4.17) The first-order entity nominalisation (Participant nominalisation)
 Mei-wa [Chibi-Totoro-ga otoshi-ta no]-o hiro-tta.
 M-TOP [CT-NOM drop-PAST NO]-ACC pick.up-PAST
 ‘Mei picked up a thing which Chibi-Totoro dropped.’

The *no*-part in situation nominalisation denotes a second-order entity or a third-order entity. In (4.8), repeated here as (4.18), the *no*-part denotes a second-order entity: an event where Nekobasu was running. In (4.5), repeated here as (4.19), the *no*-part denotes a third-order entity: the proposition that Totoro collects acorns.

- (4.18) The second-order entity nominalisation (Situation nominalisation)
 Mei-wa [Nekobasu-ga hashiru no]-o mi-ta.
 M-TOP [N-NOM run NO]-ACC see-PAST
 ‘Mei saw an event where Nekobasu was running.’

⁵ In the original example, a superscript is attached to each item to indicate a tone.

(4.19) The third-order entity nominalisation (Situation nominalisation)

Mei-wa [Totoro-ga donguri-o atsumeteiru no]-o shitteiru.

M-TOP [T-NOM acorn-ACC collect NO]-ACC know

‘Mei knows that Totoro collects acorns.’

In *no*-nominalisation, if a matrix verb takes either a noun or a (nominalised) clause as an argument, the string may be ambiguous between the two types of *no*-nominalisation. In (4.20), the matrix verb *mi-* (= ‘see’) can select either a noun or a (nominalised) clause as a complement; as expected, the string is ambiguous between participant nominalisation (4.20)a and situation nominalisation (4.20)b.

(4.20) Ambiguity case

Satsuki-to-Mei-wa [sora-o hashi-tteiru no]-o mi-ta.

S-and-M-TOP [air-ACC run-PROG NO]-ACC see-PAST

- a. ‘Satsuki and Mei saw something which was moving across the sky.’
- b. ‘Satsuki and Mei saw an event where something was moving across the sky.’

Crucially, the same ambiguity pattern is observed in other languages, like Numhpuk Singho (Tibeto-Burman).

(4.21) Ambiguity case

[Numhpuk Singho]

[Mam htu hpa wa] mu n-nga.

[rice pound NMNZ DEF] also NEG-have

- a. ‘The rice pounding machine is also not here.’
- b. ‘There is also no event of rice pounding.’ (Morey 2011: 297)

In this string, (4.21)a is a case of first-order entity nominalisation, and (4.21)b is a case of second-order entity nominalisation.

The upshot is that it is cross-linguistically attested that the same morpheme is used for first-order entity nominalisation and second-/third-order entity nominalisation. This may be taken as another motivation for a unitary approach to *no* in participant and situation nominalisation.

There is a vexing problem, however. In some languages, the same item has the genitive and the nominalising functions, as in the case of *de* in Mandarin Chinese.

- (4.22) The genitive [Mandarin]
 wo de shu
 I GEN book
 ‘my book’ (Simpson & Wu 2001: 251)

- (4.23) The first-order entity nominalisation [Mandarin]
 Ta zongshi ting [wo shuo de].
 he always listen [I speak NMNZ]
 ‘He always obeys me.’ (Simpson & Wu 2001: 251)

This may suggest that a unified analysis of the genitive *no* and the two types of *no*-nominalisation may be preferable. Yet, the nominaliser *kes* in Korean, which is quite similar to the nominaliser *no* in Japanese, does not have a genitive function (Horie 1998: 178). Given these Korean data, together with the functional and diachronic considerations in the above sub-sections, a uniform treatment of the two types of *no*-nominalisation (excluding the genitive *no*) in Figure 4.1 would not be unreasonable.

2.6 Cross-Dialectal Considerations

The last sub-section provided a typological consideration across languages, and this sub-section turns to a typological consideration within a given language.

In the Tokyo dialect of Japanese, participant and situation nominalisation and the genitive are all realised with *no*. In a number of other dialects, however, it is only participant and situation nominalisation, but crucially not the genitive, that are realised with the same form (Yoshimura 2005, 2010). Consider the Tosa dialect; the genitive is formed with *no* as illustrated by *Kochi-no* in (4.25), but participant and situation nominalisation are both formed with *ga*, as shown in (4.24) and (4.25) respectively (Yoshimura 2010: 601-3).

(4.24) Participant nominalisation [Tosa dialect]

[Kireina hana]-wa nanohana de, [shiroi ga]-wa daikon ja.
 [beautiful flower]-TOP colza COP [white NO]-TOP radish COP
 ‘The beautiful flowers are colza blossoms, and the white ones are radish.’

(4.25) Situation nominalisation [Tosa dialect]

[[Kochi-no katsuo]-ga oishii ga]-wa honto.
 [[Kochi-GEN bonito]-NOM delicious NO]-TOP true
 ‘It is true that Kochi bonitos are delicious.’

Yoshimura (2010: 599-603) cites only a single dialect in which participant and situation nominalisation are formed with different particles. In the Yatsushiro dialect, participant nominalisation is instantiated with *tsu*, and situation nominalisation is instantiated with *to*, as illustrated in (4.26) and (4.27). The genitive is realised as *no*, as shown in (4.28), where *no* is shortened as *n* after a vowel.

(4.26) Participant nominalisation [Yatsushiro dialect]

[Son akaka tsu]-ba totte-yo.
 [that red NO]-ACC take-please
 ‘Please get me that red one.’

(4.27) Situation nominalisation [Yatsushiro dialect]

[Kodomo-n piano-ba hiku to]-ba kiitotta.
 [child-NOM piano-ACC play NO]-ACC heard
 ‘I heard the child play the piano.’

(4.28) Genitive [Yatsushiro dialect]

[Asoko-n uchi-n musuko]-n kinou kekkonsashita.
 [that-GEN house-GEN son]-NOM yesterday got.married
 ‘The son of that house got married yesterday.’

These cross-dialectal data indicate a strong tendency that participant and situation nominalisation are encoded in the same form and that the form is not associated with the genitive construction.

These dialectal data fit in nicely with Figure 4.1, which depicts a unitary analysis of the particle *no* in participant and situation nominalisation (but keeping out the genitive construction).

2.7 Summary

A methodological principle of Modified Occam's Razor favours a uniform analysis of *no* in the three constructions, other things being equal: participant nominalisation, situation nominalisation, and the genitive. I have shown that the 'other things' are not equal from functional, diachronic, cross-linguistic, and cross-dialectal points of view. These various considerations justify a uniform treatment of the two types of *no*-nominalisation, excluding the genitive. That is, there is a single item *no* which manifests itself in participant nominalisation in (4.7) and situation nominalisation in (4.8), reproduced here as (4.29) and (4.30), but which does not appear in the genitive in (4.6), repeated here as (4.31). With this assumption, the next section will develop a uniform account of *no*-nominalisation from the perspective of language dynamics.

(4.29) Participant nominalisation

Mei-wa [Chibi-Totoro-ga otoshi-ta no]-o hiro-tta.
 M-TOP [CT-NOM drop-PAST NO]-ACC pick.up-PAST
 'Mei picked up a thing which Chibi-Totoro dropped.'

(4.30) Situation nominalisation

Mei-wa [Nekobasu-ga hashiru no]-o mi-ta.
 M-TOP [N-NOM run NO]-ACC see-PAST
 'Mei saw an event where Nekobasu was running.'

(4.31) Genitive *no*

Yasuko-wa Mei-no-toumorokoshi-o tabeta.

Y-TOP M-GEN-maize-ACC ate

‘Yasuko ate Mei’s maize.’

3 A Dynamic Account

3.1 Introduction

The key to modelling *no*-nominalisation is the lexical entry of the particle *no*. In Cann et al. (2005: 285), the particle *no* is analysed as a **nominaliser** that maps a preceding clause onto a nominal which denotes an entity reflecting the propositional content of the preceding clause.⁶ In this section, I shall demonstrate that this entry of *no* integrates the two types of *no*-nominalisation.

Sect. 3.2 formally presents the entry of *no* in Cann et al. (2005: 285) and puts forward my proposal that the two types of *no*-nominalisation are reducible to a parser’s choice of what type-*e* term it copies at the time of processing *no*. With this proposal, Sect. 3.3 analyses participant and situation nominalisation, and Sect. 3.4 accounts for a range of *no*-nominalisation characteristics, including the corpus data offered in Chap. 2. Finally, Sect. 3.5 summarises the main results.

3.2 The Proposal

Cann et al. (2005: 285) conceive of the particle *no* as a nominaliser which takes the **propositional** content of the pre-*no* clause and outputs a **type-*e*** term that reflects the propositional content (see also Otsuka 1999). More formally, the particle *no* inversely LINKs the structure of the pre-*no* clause to a fresh node and decorates it with a type-*e* term reflecting the propositional structure. The entry of *no* is defined in (4.32).

⁶ Cann et al. (2005) present the entry of *no* in exploring head-internal relatives. In Chap. 6, I shall show that head-internal relatives cannot be properly handled with this entry of *no*.

(4.32) Entry of the nominaliser *no*

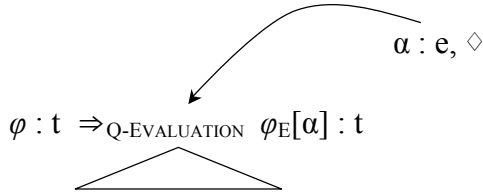
```

IF      t
THEN    IF       $\varphi_E[\alpha]$ 
        THEN     $\text{make/go}(<L^{-1}>); \text{put}(\alpha : e)$ 
        ELSE    ABORT
ELSE    ABORT

```

An attendant tree display is given in (4.33). φ_E is an evaluated proposition of the pre-*no* clause, and α is an arbitrary type-*e* term in φ_E . The tree of the pre-*no* clause is inversely LINKed to the type-*e* node decorated with α .

(4.33) Output structure of parsing *no*



As in the other DS rules, the entry of *no* is in conditional format. The IF-blocks specify the input condition: the current node should be a type-*t* node, and there must be an evaluated proposition φ_E which involves a term α . If this input condition is met, then a sequence of actions is given in the inner THEN-block. First, $\text{make/go}(<L^{-1}>)$ creates an inverse LINK relation from the type-*t* node to a fresh node and moves the pointer $\diamond$ there. Second, $\text{put}(\alpha : e)$ decorates this new node with the content α and the logical type *e*. Finally, if the input condition is not satisfied, the outer or inner ELSE-block applies, in which case the parser quits the tree update.

In Cann et al. (2005), DS trees did not have a node position for the situation term, though there were suggestive remarks in this regard. Therefore, in Cann et al. (2005), the logical type *e* in $\text{put}(\alpha : e)$ was an **individual** type. In the present thesis, the logical type *e* is a **cover** for the individual type e_i (which has been notated as *e* for brevity) and the situation type e_s (Gregoromichelaki 2011). I propose to construe the type *e* in $\text{put}(\alpha : e)$ as the entity type. Consequently, the resulting decoration may be $\alpha :$

e_i or $\alpha : e_s$. (For the sake of simplicity, I shall keep notating the individual type e_i as e .)

This leeway is the source of the two types of *no*-nominalisation.

(4.34) Proposal

The particle *no* inversely LINKs a tree of the preceding clause to a type- e node, to be decorated with an evaluated term. The two types of *no*-nominalisation are reducible to a parser's choice of what term it copies in processing *no*.

- a. If the parser copies an **individual** term, participant nominalisation will emerge.
- b. If the parser copies a **situation** term, situation nominalisation will emerge.

3.3 The Analysis

Based on the lexical entry of *no* (Cann et al. 2005), this section shows that the proposal (4.34) integrates the two types of *no*-nominalisation. Sect. 3.3.1 analyses participant nominalisation, and Sect. 3.3.2 analyses situation nominalisation. An ambiguity case is taken up in Sect. 3.3.3.

3.3.1 Participant Nominalisation

An example of participant nominalisation is (4.35).⁷ The starting point of the DS tree update is, as always, the AXIOM (4.36). This initial state is updated gradually as the string (4.35) is parsed left-to-right.

(4.35) Participant nominalisation

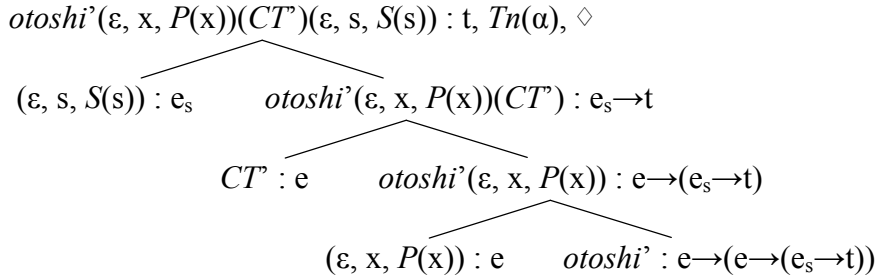
[Chibi-Totoro-ga otoshi-ta no]-o Mei-ga hiro-tta.
 [CT-NOM drop-PAST NO]-ACC M-NOM pick.up-PAST
 'Mei picked up a thing which Chibi-Totoro dropped.'

⁷ The example (4.35) is slightly different from our original (4.3), a major difference being that the matrix subject is placed before the matrix verb *hiro-* (= 'pick.up'). This change is entirely for purposes of illustration. In (4.3), the parser runs *ADJUNCTION after the matrix subject is scanned. *ADJUNCTION creates a type- t -requiring unfixed node, to be developed by the pre-*no* clause. No matter which string the parser processes, the entry of *no* remains the same, and the output tree representation is also the same.

(4.36) AXIOM

$$Tn(\alpha), ?t, \diamond$$

First, the pre-*no* clause constructs a propositional structure under the initial node. This is the usual process illustrated in Chap. 3. LOCAL *ADJUNCTION introduces a type-e-requiring unfixed node, which is decorated by *Chibi-Totoro* and resolved as a subject node by the nominative-case particle *ga*. The verb *otoshi-* (= ‘drop’) projects a propositional schema where a situation, a subject, and an object node are decorated with a meta-variable. First, the subject node harmlessly collapses with the node decorated by *Chibi-Totoro* because a meta-variable is fully commensurate with any specified formula (see Sect. 4.1, Chap. 3). Second, the object meta-variable is pragmatically substituted with $(\varepsilon, x, P(x))$, where P is an abstract predicate (Kempson & Kurosawa 2009: 65). Third, the situation meta-variable is pragmatically substituted with $(\varepsilon, s, S(s))$, where S is a situation predicate whose detail is worked out in Sect. 4.4.

(4.37) Parsing *Chibi-Totoro-ga otoshi-ta*

The top node is decorated with a propositional content, and it is subject to QUANTIFIER EVALUATION (Q-EVALUATION). Suppose that the situation term $(\varepsilon, s, S(s))$ out-scopes the individual term $(\varepsilon, x, P(x))$. Then, the individual term is evaluated first as in (4.38), and then the remaining situation term is evaluated as in (4.39).

(4.38) Evaluating the individual term $(\varepsilon, x, P(x))$ in (4.37)

$$P(a) \& otoshi'(a)(CT'')(\varepsilon, s, S(s))$$

$$\text{where } a = (\varepsilon, x, P(x) \& otoshi'(x)(CT'')(\varepsilon, s, S(s)))$$

(4.39) Evaluating the situation term $(\epsilon, s, S(s))$ in (4.38)

$$S(b) \& [P(a_b) \& \textit{otoshi}'(a_b)(CT^r)(b)]$$

$$\text{where } b = (\epsilon, s, S(s) \& [P(a_s) \& \textit{otoshi}'(a_s)(CT^r)(s)])$$

$$a_b = (\epsilon, x, P(x) \& \textit{otoshi}'(x)(CT^r)(b))$$

$$a_s = (\epsilon, x, P(x) \& \textit{otoshi}'(x)(CT^r)(s))$$

In (4.39), b stands for the evaluated situation term, and a_b stands for the evaluated individual term. These terms reflect the **full** content of the proposition. (a_s is just part of the term b ; a_s is not a full-blown term since the variable s is unbound within it.)

So far, the tree update process is not novel. Now, the nominaliser *no* is parsed. The entry of *no* is repeated here as (4.40).

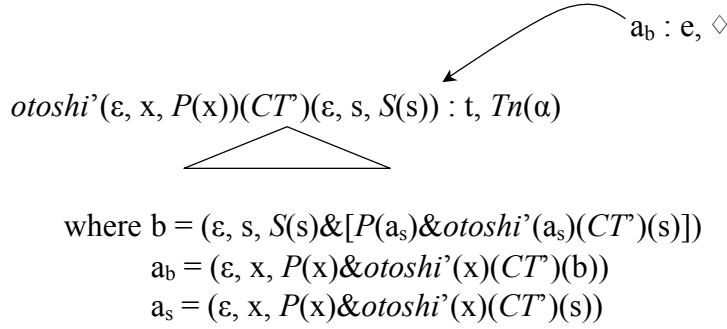
(4.40) Entry of the nominaliser *no*

```

IF      t
THEN    IF       $\varphi_E[\alpha]$ 
          THEN   $\text{make/go}(\langle L^{-1} \rangle); \text{put}(\alpha : e)$ 
          ELSE  ABORT
ELSE    ABORT

```

The input conditions are met by the active node in (4.37), as marked by a pointer $\diamond$. That is, the active node is decorated with the logical type t , and there is an evaluated proposition (4.39) that contains the evaluated terms b and a_b . Since the input conditions are satisfied, the parser creates an inverse LINK relation to a fresh node and decorates it with an evaluated term. The parser has to **decide** which term it copies, the situation term b or the individual term a_b . Either of them could be selected, but if the situation term b is chosen, the tree update will crash at a later point because the matrix predicate *hiro'* (= *pick_up'*) cannot take a situation term as an argument. Therefore, the parser chooses the individual term a_b and decorates the fresh node with the pair $a_b : e$.

(4.41) Parsing *Chibi-Totoro-ga otoshi-ta no*

The next item to be parsed in (4.35) is the accusative-case particle *o*. The entry of *o* has been presented in Sect. 5.2, Chap. 3, reproduced here as (4.42).

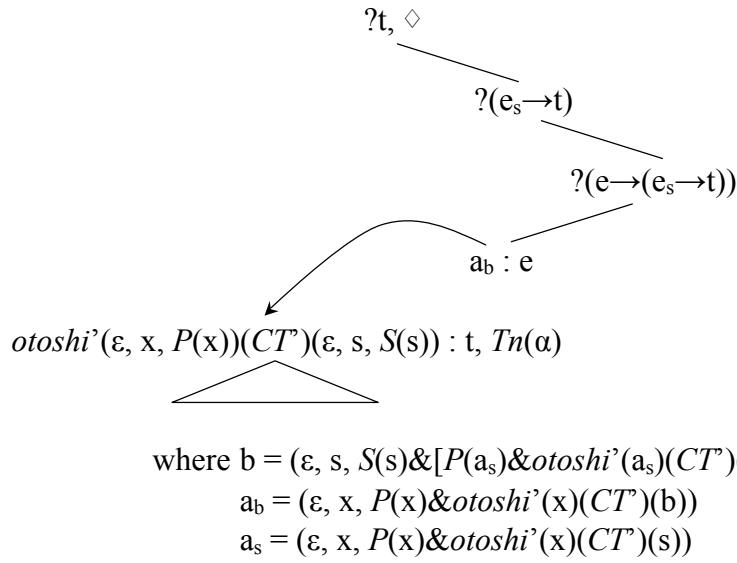
(4.42) Entry of the accusative-case particle *o* (final version)

```

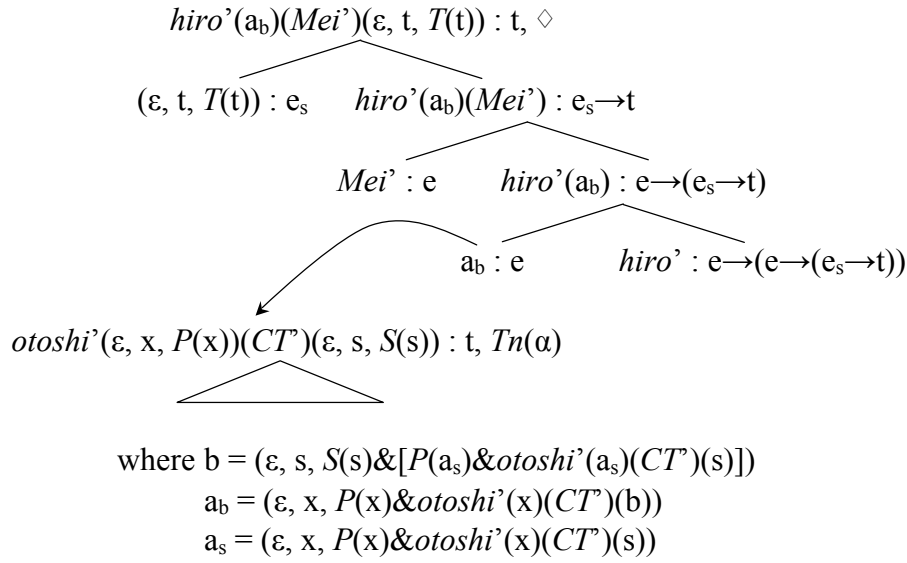
IF      e
THEN    IF      <\uparrow_0 \uparrow_1^*> (Tn(\alpha), ?t)
        THEN    put(?<\uparrow_0> (e \rightarrow (e_s \rightarrow t)))
        ELSE    IF      <\uparrow> \perp
                THEN    make/go(<\uparrow_0>); put?(e \rightarrow (e_s \rightarrow t));
                        make/go(<\uparrow_1>); put?(e_s \rightarrow t);
                        make/go(<\uparrow_1>); put(?t, Tn(\alpha));
                        (make/go(<\uparrow^*>); put(?t, Tn(\beta)))
                ELSE    ABORT
ELSE    ABORT

```

In the tree (4.41), the active node is a type-e node and there is no structure upwards. Thus, the parser constructs a partial structure bottom-up as in (4.43), where the node decorated with a_b is resolved as an object node within the new structure.

(4.43) Parsing [*Chibi-Totoro-ga otoshi-ta no*]-o

In this new structure, the rest of the string (4.35) is parsed. First, LOCAL *ADJUNCTION fires to create a type-e-requiring unfixed node, to be decorated by *Mei* and settled as a subject node by the nominative-case particle *ga*. The structure is fleshed out by the matrix verb *hiro-* (= ‘pick up’), which posits a propositional template where a situation, a subject, and an object node are decorated with a meta-variable. The subject node collapses with the node decorated by *Mei*, and the object node collapses with the node decorated with the individual term a_b . Finally, a meta-variable at the situation node is pragmatically substituted with the situation term $(\epsilon, t, T(t))$.

(4.44) Parsing [*Chibi-Totoro-ga otoshi-ta no*]-o *Mei-ga hiro-tta*

The tree (4.44) is the final state of the tree update for the string instantiating participant nominalisation (4.35), with the proposition at the top node representing the interpretation: ‘Mei picked up a thing which Chibi-Totoro dropped.’⁸ In this tree update, the nominaliser *no* creates a type-*e* node decorated with the individual term a_b . Due to Q-EVALUATION, this term denotes an entity that satisfies the description of the pre-*no* clause, picking out an arbitrary object that Chibi-Totoro dropped.

3.3.2 Situation Nominalisation

The same entry of *no* also accounts for the other type of *no*-nominalisation: situation nominalisation such as (4.45).⁹

(4.45) Situation nominalisation

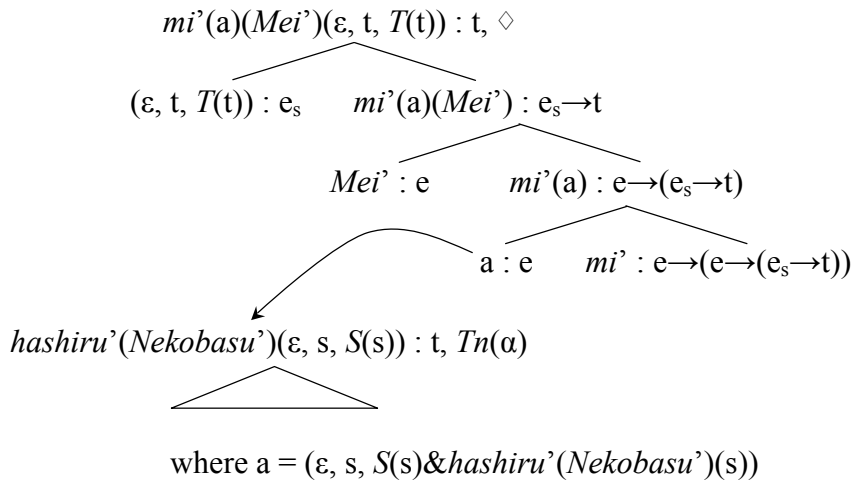
[Nekobasu-ga hashiru no]-o Mei-wa mi-ta.
 [N-NOM run NO]-ACC M-TOP see-PAST
 ‘Mei saw an event where Nekobasu was running.’

⁸ The proposition at the top node will be Q-EVALUATED. This process is neglected here since it complicates our illustration and it is not directly pertinent to our discussion either.

⁹ This is slightly different from the original (4.4) for illustration purposes; see footnote 7.

The rest of the process is as set out in the last sub-section. The current node is marked as an object node by the accusative-case particle *o*, which constructs a partial structure bottom-up. LOCAL *ADJUNCTION introduces a type-*e*-requiring unfixed node, which is decorated by *Mei* and fixed as a subject node by the nominative-case particle *ga*. Finally, the matrix verb *mi-* (= ‘see’) builds a propositional schema; the subject node collapses with the node decorated by *Mei*, the object node collapses with the node annotated with the term *a*, and a situation meta-variable is saturated as $(\varepsilon, t, T(t))$.

(4.49) Parsing [*Nekobasu-ga hashiru no*]-*o Mei-ga mi-ta*



In (4.49), the propositional formula at the top node represents the interpretation of the situation nominalisation string (4.45): ‘Mei saw an event where Nekobasu was running.’¹⁰ The nominaliser *no* introduces a node decorated with the situation term *a*, which, owing to Q-EVALUATION, denotes a situation that reflects the full content of the pre-*no* clause, picking out an arbitrary situation where Nekobasu was running. One benefit of this account is that it accommodates variants of situation nominalisation. In DS, situation terms are assumed to denote a range of situational entities; see also Kim (2009). This accords with the corpus finding in Chap. 2 that the *no*-part in situation

¹⁰ For a full interpretation, the proposition will be Q-EVALUATED. This process is ignored here for the reasons provided in footnote 8.

nominalisation may denote a wide variety of situational entities such as actions, events, propositions, and so on.

3.3.3 Ambiguity Cases

In Sect. 3.3.1, where participant nominalisation was discussed, it was argued that if the parser copies a situation term in processing *no*, the tree update crashes at a later point since a situation term cannot be selected by the matrix predicate. The prediction, then, is that if a matrix predicate is compatible with either an individual term or a situation term, the string should be ambiguous (but see the discussion about (4.48)). In fact, the ambiguity case has already been presented as (4.20), repeated here as (4.50).

(4.50) Ambiguity case

Satsuki-to-Mei-wa [sora-o hashitteiru no]-o mi-ta.

S-and-M-TOP [air-ACC running NO]-ACC see-PAST

- a. ‘Satsuki and Mei saw something which was moving across the sky.’
- b. ‘Satsuki and Mei saw an event where something was moving across the sky.’

At the time of processing *no*, the parser has to make a decision of what type-e term it copies. If the parser copies the evaluated individual term (i.e., the term denoting an object that was moving across the sky), the participant nominalisation reading (4.50)a arises. On the other hand, if the parser copies the evaluated situation term (i.e., the term denoting a situation where something was moving across the sky), the situation nominalisation reading (4.50)b arises.

The distinction between participant and situation nominalisation emerges as an **outcome** of semantic tree growth, especially a parser’s choice as to what type-e term it copies at the time of processing *no*.

3.4 Consequences

No-nominalisation has been integrated from the viewpoint of how an interpretation is gradually built up as a string is processed left-to-right. This dynamic account predicts a number of *no*-nominalisation data, including the corpus findings in Chap. 2.

3.4.1 *No* as a Dependent Item

Makino (1968: 51) observes that *no* cannot stand on its own. Thus, if *no* is not preceded by a clause as in (4.51), the string is ungrammatical. This clearly contrasts with the grammatical example (4.52).

(4.51) Participant nominalisation

*No-o Mei-ga hirotta.
 NO-ACC M-NOM picked.up
 Int. ‘Mei picked up something.’

(4.52) Participant nominalisation

[Chibi-Totoro-ga otoshita no]-o Mei-ga hirotta.
 [CT-NOM dropped NO]-ACC M-NOM picked.up
 ‘Mei picked up a thing which Chibi-Totoro dropped.’

Although Makino considers only participant nominalisation, it is also true of situation nominalisation. Thus, (4.53) should be compared with (4.54).

(4.53) Situation nominalisation

*No-o Mei-wa shitteiru.
 NO-ACC M-TOP know
 Int. ‘Mei knows it.’

(4.54) Situation nominalisation

[Totoro-ga donguri-o atsumeteiru no]-o Mei-wa shitteiru.
 [T-NOM acorn-ACC collect NO]-ACC M-TOP know
 ‘Mei knows that Totoro collects acorns.’

These data are readily accounted for. The entry of *no* requires that a proposition should have been built up **before** the parse of *no*. Formally, this requirement is expressed in the IF-clauses in the entry of *no* (4.32). In (4.51) and (4.53), however, *no* does not have any preceding clause, and a proposition cannot be constructed.¹¹

3.4.2 Expressive Meaning

It is well-known that if the *no*-part denotes a human in participant nominalisation, a derogatory connotation is observed (Kitagawa, 2005: 1259; see also Kuroda 1992). Thus, derogatory expressive meaning is found in participant nominalisation (4.55) and (4.57)a, but not in situation nominalisation (4.56) and (4.57)b.

(4.55) Participant nominalisation

Satsuki-wa [nai-teiru no]-o shika-tta.
 S-TOP [cry-PROG NO]-ACC scold-PAST
 ‘Satsuki scolded the crying person.’

(4.56) Situation nominalisation

Mei-wa [Totoro-ga donguri-o atsumeteiru no]-o shitteiru.
 M-TOP [T-NOM acorn-ACC collect NO]-ACC know
 ‘Mei knows that Totoro collects acorns.’

(4.57) Ambiguity case

Satsuki-wa [hashi-tteiru no]-o mi-ta.
 S-TOP [run-PROG NO]-ACC see-PAST

- a. ‘Satsuki saw someone who was running.’ (Participant nominalisation)
- b. ‘Satsuki saw an event where someone was running.’ (Situation nominalisation)

¹¹ Later chapters analyse other *no*-involving constructions by combining the entry of *no* with the entry of the topic particle *wa* (for clefts) or combining the entry of *no* with the entry of a case particle (for head-internal relatives). These composite entries preserve the input condition that the current node should be decorated with a proposition. Thus, it is correctly predicted that the pre-*no* clause in clefts and head-internal relatives cannot stand on their own either.

What has not been reported in the literature is that expressive meaning is not always derogatory. In (4.55), if the relation between Satsuki and the crying person is that of sisterhood, and Satsuki finds the way the person cries to be cute, then the expressive meaning may be an affectionate familiarity with the denoted person.¹²

To account for the data, I posit the constraint that the denotation of the *no*-part should be an **object** (rather than a human), the idea being that if the *no*-part denotes a human, expressive meaning arises pragmatically.¹³ In (4.55), given the predicate *nai*- (= ‘cry’), the parser foresees that the *no*-part denotes a human, and may decorate a gap node with (4.58) by updating the abstract predicate *P* into the concrete predicate *hito*’ (= *person*’). When the tree transition proceeds, the parser in processing *no* may copy the evaluated version of the term (4.58), which denotes a person who was crying.

(4.58) $(\epsilon, x, \textit{hito}'(x))$

Given the constraint that the denotation of the *no*-part should be an object (rather than a human), the speaker treats a denoted person as if s/he were a thing, which gives rise to a pragmatic implication that the speaker does not treat the person respectfully. This pragmatic inference produces the derogatory connotation. The same analysis applies to (4.57)a. Then, how about the absence of connotation in situation nominalisation (4.56) and (4.57)b? In these situation nominalisation cases, what is copied is a situation term (but not an individual term). Since a situation is not a human, the aforementioned pragmatic inference does not take place, and expressive meaning does not arise.

¹² I would like to thank Yuji Nishiyama and Yukio Otsu, who independently pointed out to me that the type of expressive meaning was not always derogatory.

¹³ This may be modelled in line with Cann & Wu’s (2011) analysis of the *bei* construction in Chinese. The particle *bei* marks the pre-*bei* item as the locus of affect, projecting a structure where the Locus-of-Affect (LoA) predicate takes as an internal argument the content of the pre-*bei* item, and as an external argument the content of the rest of the string. The LoA predicate is underspecified for the type of affect, and this fits well with the context-dependency of *no*-connotation. It is possible to assume that the entry of *no* has a constraint that if a term to be copied denotes a human, it projects a structure involving the LoA predicate.

This pragmatic analysis accounts for the context-dependence of expressive meaning. In (4.55), consider a context where Satsuki scolded her younger sister, Mei, and the speaker is fond of the sisters. In this context, the hearer may assume that the speaker describes the crying person, Mei, in jest by treating her as if she were a thing, which pragmatically yields the speaker's affectionate familiarity with the crying person. This analysis also predicts that if the *no*-part denotes a non-human, the connotation should be absent. This is ascertained by (4.59). Due to the predicate *otoshi-* (= 'drop'), the *no*-part denotes a non-human thing (e.g., acorn). Thus, the pragmatic inference mentioned above is not triggered, and expressive meaning is not produced.

(4.59) Participant nominalisation

Mei-wa [Chibi-Totoro-ga otoshi-ta no]-o hiro-tta.
 M-TOP [CT-NOM drop-PAST NO]-ACC pick.up-PAST
 'Mei picked up a thing which Chibi-Totoro dropped.'

Our account has implications for a cross-linguistic study of nominalisation. Consider the Korean data (4.60), drawn from Seraku (2012b: 160).

- (4.60) [Wu-nun kes]-ul Tom-i pwa-ss-ta. [Korean]
 [cry-PRE KES]-ACC Tom-NOM see-PAST-DECL
 a. *'Tom saw someone who cried.' (participant nominalisation)
 b. 'Tom saw the event of someone's having cried.' (situation nominalisation)

While (4.60)b is acceptable, (4.60)a is not.¹⁴ Of note is that the nominaliser *kes*, unlike the nominaliser *no*, was derived from the lexical noun *kes* meaning 'thing,' and that this meaning persists in the nominaliser *kes* (Horie, 2008: 178). Then, the restriction that the denoted entity should be an object is stronger in *kes* than in *no*; this is why the reading (4.57)a in Japanese is possible but the reading (4.60)a in Korean is not.

¹⁴ The degraded status of (4.60)a does not mean that *kes* lacks participant nominalisation. In fact, if *wu-nun* in (4.60) is replaced with *kkayeci-nun* (= 'break-PRE'), the string exhibits participant nominalisation: 'Tom saw something (e.g., machine) that was being broken.'

There are some previous studies which are relevant to the present discussion. First, McGloin (1985) also suggests, albeit very briefly, a pragmatic analysis of expressive meaning of *no*. In her analysis, however, neither situation nominalisation nor the context-dependency of expressive connotation is addressed. Second, in Mainstream Generative Grammar, Kitagawa (2005) suggests that connotation emerges only if the external-head *pro* has an indefinite referent. But suppose that (4.61) is uttered with a pointing gesture; further, the demonstrative *sono* (= ‘that’) is used in order to ensure that the small *pro* has a definite referent.

(4.61) Participant nominalisation

Satsuki-wa sono [hashi-tteiru no]-o mi-teiru.
 S-TOP that [run-PROG NO]-ACC see-PROG
 ‘Satsuki is seeing that person who is running.’

In (4.61), expressive meaning is observed, contrary to what Kitagawa (2005) would predict. Our analysis postulates neither a null element nor an external-head position; the presence of expressivity is envisaged as a result of pragmatic inference.¹⁵

3.4.3 Nature of Denotation

Kamio (1983) and McGloin (1985) observe that *no* in participant nominalisation cannot refer to abstract entities. Consider the contrast between (4.62) and (4.63), translated and adapted from Kamio (1983: 82). The string (4.63) is acceptable if the *no*-part denotes some non-abstract entity such as a stone, but it cannot be interpreted as the reading provided in the free translation line.

¹⁵ Later chapters analyse clefts and head-internal relatives by combining the entry of *no* and that of a particle (see footnote 11). In these constructions, expressive meaning is never conveyed. I assume that the nominaliser *no*, a locus of a connotation, does not stand as a separate item in the unit ‘*no* + particle.’ See also Sect. 4.4.4, Chap. 6.

(4.62) Participant nominalisation

[[katai shinnen]-o motta] hito
 [[solid belief]-ACC have] person
 ‘a person who has a solid belief’

(4.63) Participant nominalisation

*[[katai no]-o motta] hito
 [[solid NO]-ACC have] person
 Int. ‘a person who has a solid belief’

It seems, however, that the above generalisation is suspicious. In (4.63), the use of the predicate *katai*- (= ‘solid’) is metaphorical; it drives the interpreter to look for a physical object to which the predicate *katai*- normally applies (e.g., stone). This is why it is hard to get the intended reading in (4.63).¹⁶ If a predicate that is congruous with an abstract object is utilised, such as *settokutekina*- (= ‘convincing’), the *no*-part may denote an abstract entity such as arguments, data, ideas, etc., as in (4.64).

(4.64) Participant nominalisation

[gakkai-de [settokutekina no]-o teijishita] hito
 [conference-at [convincing NO]-ACC presented] person
 ‘a person who presented a convincing one (e.g., argument) at a conference’

Given our unitary analysis, it is anticipated that if the *no*-part may denote an abstract entity in participant nominalisation, it should hold of situation nominalisation too. This expectation is borne out. In (4.65), the *no*-part denotes the abstract proposition that two plus two equals four. In (4.66), the modal statement, which seems to denote a proposition, is nominalised by *no*.

¹⁶ I would like to thank Ruth Kempson for bringing this point to my attention.

(4.65) Situation nominalisation

Mei-wa [[ni tasu ni]-ga yon dearu no]-o shitteiru
 M-TOP [[2 plus 2]-NOM 4 COP NO]-ACC know
 ‘Mei knows that 2 plus 2 equals 4.’

(4.66) Situation nominalisation¹⁷

Yasue-wa [Mei-ga kuru kamoshirenai no]-o omoidashita.
 Y-TOP [M-NOM come might NO]-ACC remembered
 ‘Yasue remembered that Mei might come.’

At the same time, however, there is an indication that *no* in situation nominalisation tends to denote a perceptible event (Kuno 1973b: 222; see also Oshima 2010, Watanabe 2008). In (4.67), if *no* is employed, the *no*-part describes the event of Totoro’s singing as a tangible one. In this reading, Mei perceives the event described through her five senses. This is contrasted with the situation nominaliser *koto*. If *koto* is used, the *koto*-part describes the event of Totoro’s singing as a less tangible one. In this reading, Mei does not perceive the event by herself, but someone told Mei that Totoro planned to sing. Put differently, if *no* is used, the embedding verb *kii-* (= ‘hear’) serves as a perception verb; if *koto* is used, *kii-* serves as a hearsay verb.

(4.67) Situation nominalisation

Mei-wa [Totoro-ga utau no/koto]-o kii-ta.
 M-TOP [T-NOM sing NO/KOTO]-ACC hear-PAST
 ‘Mei heard Totoro singing.’ (with the nominaliser *no*)
 ‘Mei heard that Totoro would sing.’ (with the nominaliser *koto*)

I contend that this difference between *no* and *koto* reflects the origins of these items. As noted in Horie (2008: 174), there are no confirmed lexical origins of *no*, but *koto* is a diachronically bleached development of the lexical noun *koto*, meaning ‘matter’ or

¹⁷ This example was constructed based on the suggestion made by an anonymous reviewer of the 26th Pacific Asia Conference on Language, Computation, and Information.

‘event.’ It may then be assumed that *koto* retains the property of denoting an event as a matter, and that this lexical residue is encoded as a constraint in the nominaliser *koto* (but not in the nominaliser *no*). Then, the difference in (4.67) can be analysed as the difference in the encoded constraints of *koto* and *no*.

3.4.4 Definiteness

As reported in Chap. 2 (Sect. 4.2), the *no*-part may denote either definite or indefinite entities. This is predicted by our analysis couched within the epsilon calculus, where the ‘epsilon’ operator ε models **existential** quantification and the ‘iota’ operator ι models Russellian (1905) **uniqueness** (see Sect. 2.2, Chap. 3).

So far, a number of terms with the epsilon operator ε have been presented. An example to be analysed with the iota operator ι is (2.30), repeated here as (4.68). The topic of the speech is an aiding system for those who have hearing impairments.

- (4.68) Participant nominalisation, extracted from the CSJ (A01M0152)
- | | | | | | | |
|------------|-----------------------------|-------|---------|----------|----------------|-----|
| kono | [kakko-ni | aru | no]-wa | kore-ga | kidou | de |
| this | [parenthesis-in | exist | NO]-TOP | this-NOM | air.conduction | COP |
| kochira-ga | kotsudou-no-chouryoku... | | | | | |
| this-NOM | bone.conduction-GEN-hearing | | | | | |
- ‘As for these in the parentheses, this is an air conduction and this is the hearing ability of a bone conduction.’

The demonstrative *kono* forces the parser to construct a term with the iota operator ι . Thus, the parser decorates a new node with the term (4.69). The content of *kakko-ni aru* is abbreviated as *k-a*’. The term denotes a unique entity that is in the parentheses in a situation described by the pre-*no* clause.

- (4.69) $(\iota, x, P(x) \& k-a'(x)(b))$

where $b = (\varepsilon, s, S(s) \& [P(\iota, x, P(x) \& k-a'(x)(s)) \& k-a'(\iota, x, P(x) \& k-a'(x)(s))(s)])$

The same account is viable for situation nominalisation (2.32), reproduced here as (4.70). The topic of the speech is a windmill that was built in front of his local station. The speaker refers to a particular event where the windmill was constructed. There is no morphological indication of the definiteness of the event, but the grammar-pragmatics interface allows non-linguistic information to be reflected in DS computation.

(4.70) Situation nominalisation, extracted from the CSJ (S00M0228)

[sono-suisha-ga n suisha tsukutta no]-wa ii-ndesu-kedo...

[that-windmill-NOMwell windmill created NO]-TOP fine-COP-though

‘As for the event of having created the windmill, it is fine, but...’

Both definiteness and indefiniteness of the *no*-part may be represented with different binders in the epsilon calculus, and the DS grammar-pragmatics interface lets the parser determine an appropriate binder, taking context into consideration.

3.4.5 Grammatical Function of the *No*-Part

Chap. 2 showed that if the *no*-part is marked by the topic particle *wa*, it may have a core grammatical function such as subject and object and it may also provide a pure topic. These were summarised in Table 2.3 (for participant nominalisation) and Table 2.4 (for situation nominalisation), as reproduced below:

Type	Occurrences	Percentage
subject	39	41%
object	13	14%
pure topic	43	45%
Total	95	

Table 2.3. Grammatical function of the *no*-part in participant nominalisation

Type	Occurrences	Percentage
subject	113	70%
object	9	6%
pure topic	40	25%
Total	162	

Table 2.4. Grammatical function of the *no*-part in situation nominalisation

For instance, participant nominalisation (2.33), repeated here as (4.71), shows that the *no*-part is a subject of the matrix predicate *waruku-wa-nai*- (= ‘not bad’).

- (4.71) Participant nominalisation, extracted from the CSJ (M02M0001)
 [kokoe dashiteru no]-wa zenbu waruku-wa-nai...
 [here present NO]-TOP all bad-TOP-NEG
 ‘As for what I present here (= dictionaries), they are all not bad...’

Further, the *no*-part, when marked by the topic particle *wa*, may present a pure topic. For example, (2.41), repeated here as (4.72), shows that the *no*-part presents a topic with respect to which the rest of the string (especially, the coordinate NPs *rouryoku-to-shikin* (= ‘effort and funds’)) is interpreted.

- (4.72) Situation nominalisation, extracted from the CSJ (S00F0131)
 [uma i-ttou yashinau no]-wa tashikani taihenna
 [horse 1-CL foster NO]-TOP definitely much
 rouryoku-to-shikin-ga hitsuyou desu...
 effort-and-fund-NOM be.required COP
 ‘As for fostering even a single horse, a lot of effort and funds are definitely required...’

This is anticipated in the entry of the topic particle *wa* (3.81) in Chap. 3, repeated here as (4.73). In plain English, if a current node is decorated with a type-*e* term β , the parser induces an inverse LINK relation to a new node and annotates it with $?t$ and $?<\downarrow D>(\beta)$. The latter requirement declares that there will be a β -decorated node **somewhere below** (excluding a node directly LINKed from the present node).

- (4.73) Entry of the topic marker *wa* (final version)
 IF $\beta : e$
 THEN **make/go**($<L^{-1}>$); **put**($?t, ?<\downarrow D>(\beta)$)
 ELSE **ABORT**

This entry does not impose any structural constraint on where the content β will reside (excluding a node directly LINKed from the present node). Thus, the content β may occupy a subject node, an object node, etc. As for the pure topic (4.72), I assume that *rouryoku* (= ‘effort’) and *shikin* (= ‘funds’) are argument-taking nouns: ‘effort to do X’ and ‘funds to do X.’ If this assumption is tenable, there is a node for an argument X, and this node may be decorated with the term β , satisfying the requirement $?<\downarrow D>(\beta)$. In this way, the current analysis predicts that the *no*-part, when marked by the topic particle *wa*, may bear an array of grammatical functions.

3.4.6 Grammatical Function of the Gap

Chap. 2 pointed out that the gap in the pre-*no* clause in participant nominalisation may bear a variety of grammatical functions. Table 2.5 is repeated here:

Type	Occurrences	Percentage
subject	56	59%
object	32	34%
function _{TIME}	3	3%
function _{PLACE}	2	2%
unclassifiable	2	2%
Total	95	

Table 2.5. Grammatical function of the gap in participant nominalisation

For instance, in (2.43), reproduced here as (4.74), the gap is the object of the verb *bunsekishi*- (= ‘analyse’). In (2.45), reproduced here as (4.75), the gap is an adjunct of the verb *renshuu-suru*- (= ‘practice’), bearing the function_{PLACE}.

(4.74) Participant nominalisation, extracted from the CSJ (A05M0118)

[jissaini bunsekishi-ta no]-wa doumo sukoshi takame
 [actually analyse-PAST NO]-TOP seemingly a.bit high
 noyou-nandesu...
 seem-COP

‘As for what we actually analysed, their values seem to be a bit high...’

(4.75) Participant nominalisation, extracted from the CSJ (S04F1636)

[renshuu-suru no]-wa kashi-kouto-de ya-ttari koukyou-no-kouto-de
 [practice-do NO]-TOP rental-court-at do-or public-GEN-court-at
 ya-ttari...
 do-or

‘As for the place where we play tennis, we do it at a rental court or we do it at a public court...’

These data follow from the entry of *no* (4.32), repeated here as (4.76). In the inner IF-block, α stands for a term corresponding to the gap in participant nominalisation. In the statement $\varphi_E[\alpha]$, there is **no** structural condition on where the gap is located within the propositional structure of the pre-*no* clause.

(4.76) Entry of the nominaliser *no*

IF	t	
THEN	IF	$\varphi_E[\alpha]$
	THEN	make/go(<L ⁻¹ >); put($\alpha : e$)
	ELSE	ABORT
ELSE	ABORT	

Thus, in (4.74), α is the evaluated individual term that corresponds to the object gap, whereas in (4.75), α is the evaluated individual term that corresponds to the adjunct gap. As for adjuncts, Marten (2002) claims that predicates are underspecified for valency and that the parser optionally builds type-*e* nodes for adjuncts. I do not present an account of adjuncts in this thesis, but it is licit to assume that Japanese verbs project an open proposition with all argument slots and **optionally** introduce **type-*e*** nodes for adjuncts. In (4.75), *renshuu-suru-* may optionally introduce a type-*e* locative adjunct node, and this node is decorated with $(\varepsilon, x, P(x))$, where *P* is an abstract predicate. The rest of the process is as usual; the entry of *no* makes a copy of the evaluated version of the term and puts it at an inversely LINKed node.¹⁸

¹⁸ As for two instances of ‘unclassifiable’ in Table 2.5, Chap. 2 suggested that these examples also contained a gap position, though it was difficult to pinpoint the grammatical function of

The lexical entry of the nominaliser *no*, together with the assumption regarding verbal underspecification (Marten 2002), deals with the range of grammatical functions that the gap in participant nominalisation may bear.

3.4.7 Long-Distance Dependencies and Islands

According to Chap. 2, participant nominalisation displays a long-distance dependency (2.50), repeated as (4.77), and it is even insensitive to islands (2.51), repeated as (4.78).

(4.77) Participant nominalisation

[Porco-ga [Fio-ni katte-ki-ta to] itta no]-wa oishii.
 [P-NOM [F-for buy-come-PAST COMP] said NO]-TOP tasty
 ‘As for what Porco said that he bought for Fio, it is tasty.’

(4.78) Participant nominalisation

[[Porco-ga shuuri-shita toiu usawa]-ga atta no]-wa
 [[P-NOM repair-did TOIU rumour]-NOM existed NO]-TOP
 Folgore-o tousaishiteiru.
 F-ACC have.on.board
 ‘As for the flying boat x such that there was a rumour that Porco repaired x, it has the Folgore engine on board.’

These data are amenable to the entry of *no* (4.32), reproduced here as (4.79). In the inner IF-block, $\varphi_E[\alpha]$ declares that there must be an evaluated proposition φ_E which contains an evaluated term α , but it does **not** impose any structural constraint on the place where α is found. That is, α may correspond to a term at a gap position at arbitrary depth. In the island-involving case (4.78), the term α may correspond to a term at a gap position within the complex NP headed by *uwasa* (= ‘rumour’).

these gaps. No matter what grammatical function they may turn out to bear, these examples are in principle compatible with the entry of the nominaliser *no* (4.76), which is silent as to the type of grammatical function of the gap.

(4.79) Entry of the nominaliser *no*

```

IF      t
THEN    IF       $\varphi_E[\alpha]$ 
        THEN    make/go(<L-1>); put( $\alpha : e$ )
        ELSE    ABORT
ELSE    ABORT

```

There is a residual problem, however. Since the entry of *no* (4.79) does not posit any condition on the place where the term α is located, it follows that the parser may copy an evaluated **situation** term at arbitrary depth. This wrongly predicts that in (4.80), the parser could copy the evaluated term corresponding to the situation described by the most embedded clause. This results in the unacceptable reading: ‘Mei said that Totoro was singing, and her father heard Totoro singing.’

(4.80) Situation nominalisation

Otousan-wa [Mei-ga [Totoro-ga utatteita to] iu no]-o kiita.
 father-TOP [M-NOM [T-NOM was.singing COMP] say NO]-ACC heard
 ‘Mei’s father heard her say that Totoro was singing.’
 *‘Mei said that Totoro was singing, and her father heard Totoro singing.’

But the unavailability of the ‘embedded-situation’ interpretation is not specific to *no*-nominalisation, but it is a general feature of sentential-complement constructions. For instance, consider *koto*-nominalisation (4.81). As mentioned in Sect. 3.4.3, the particle *koto* also enables situation nominalisation. One crucial difference between *no* and *koto* is that unlike *no*, *koto* does not enable participant nominalisation. Let us then assume that the nominaliser *koto* encodes the same entry with *no* except that a type-*e* term to be copied must be always a situation term (but not an individual term).¹⁹ Crucially, as in *no*-nominalisation (4.80), *koto*-nominalisation (4.81) lacks the interpretation which would be available if *koto* allowed the parser to copy a situation term of the most

¹⁹ Another difference is that a situation denoted in *no*-nominalisation tends to be described as a more tangible one than in *koto*-nominalisation. Sect. 3.4.3 suggested that this is because the nominaliser *koto* was derived from the lexical noun *koto* meaning ‘matter’ or ‘event’ while the origin of *no* is still controversial.

embedded clause. I speculate that whatever governs this general feature of sentential-complement constructions accounts for the prohibited reading in (4.80).

(4.81) *Koto-nominalisation*

Otousan-wa [Mei-ga [Totoro-ga utatteita to] itteita
 father-TOP [M-NOM [T-NOM was.singing COMP] said
 koto]-o oboeteiru.
 KOTO]-ACC remember

‘Mei’s father remembers that Mei said that Totoro was singing.’

*‘Mei said that Totoro was singing, and her father remembers that Totoro was singing.’

3.4.8 Loose Ends

Finally, there are two remaining issues. First, it was claimed that the two types of *no*-nominalisation are reduced to a parser’s choice of what term it copies **at the time of processing *no***. Now, consider (4.35) once again, repeated here as (4.82).

(4.82) *Participant nominalisation*

[Chibi-Totoro-ga otoshi-ta no]-o Mei-ga hiro-tta.
 [CT-NOM drop-PAST NO]-ACC M-NOM pick.up-PAST
 ‘Mei picked up a thing which Chibi-Totoro dropped.’

In processing *no*, the parser could copy either an individual or a situation term. But if a situation term is copied, the tree update will crash at a later point because the matrix predicate *hiro*’ (= *pick_up*’) cannot select a situation term. The point is that the parser seems unable to determine what term it should copy **until** it parses a matrix verb.

At present, no experimental evidence is available indicating that the parser is unable to make a decision as to what term it copies at the time of parsing *no*. So, I shall keep assuming the entry of *no* (4.32), repeated here as (4.83).

(4.83) Entry of the nominaliser *no*

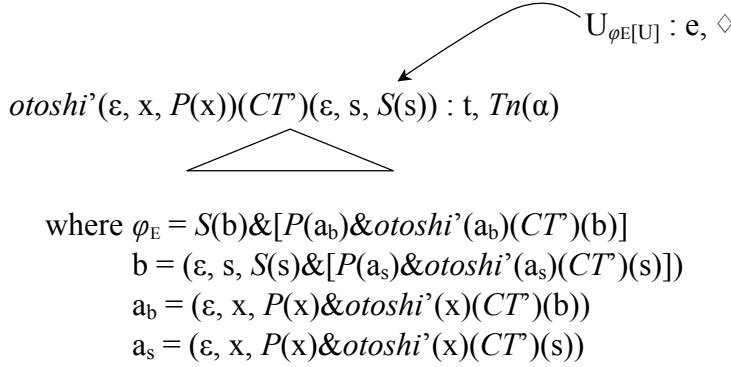
```

IF      t
THEN    IF       $\varphi_E[\alpha]$ 
        THEN    make/go(<L-1>); put( $\alpha : e$ )
        ELSE    ABORT
ELSE    ABORT

```

But if future research provides such experimental data, it is possible to amend (4.83) slightly so that it allows the parser to **delay** the determination of a term to be copied until it scans an embedding verb. In the modified entry, *no* provides a meta-variable U with the restriction $\varphi_E[U]$, as in $U_{\varphi_E[U]}$, requiring that U must be substituted with a term in an evaluated proposition φ_E . For instance, the parse of (4.82) up to the nominaliser *no* leads to the tree (4.84).

(4.84) Parsing *Chibi-Totoro-ga otoshi-ta no*



The active node in (4.84) is resolved as an object node within a new structure by the accusative-case particle *o*. In this new structure, the parser runs LOCAL *ADJUNCTION to induce a type-e-requiring node, which is decorated by *Mei* and fixed as a subject node by the nominative-case particle *ga*. The matrix verb *hiro-* (= ‘pick up’) then fleshes out the structure. It is **after** the parse of the **individual**-argument taking verb *hiro-* that U is saturated as the **individual** term a_b . In this way, a parser’s decision of term-copying may be delayed. The modified entry of *no* is presented in (4.85).

(4.85) Entry of the nominaliser *no* (modified)

```

IF      t
THEN    IF       $\varphi_E$ 
        THEN    make/go(<L-1>); put( $U_{\varphi_E[U]} : e$ )
        ELSE    ABORT
ELSE    ABORT

```

Given the lack of relevant experimental data, however, I shall continue to assume the original entry of *no* (4.83) in this thesis.

The second residual issue is raised by situation nominalisation in particular. In our analysis, the perception construction (4.86) and the factive construction (4.87) are characterised under the same umbrella ‘situation nominalisation.’

(4.86) Situation nominalisation (the perception construction)

```

Mei-wa [Nekobasu-ga hashiru no]-o mi-ta.
M-TOP  [N-NOM      run      NO]-ACC see-PAST
‘Mei saw an event where Nekobasu was running.’

```

(4.87) Situation nominalisation (the factive construction)

```

Mei-wa [Totoro-ga donguri-o atsumeteiru no]-o shitteiru.
M-TOP  [T-NOM  acorn-ACC collect      NO]-ACC know
‘Mei knows that Totoro collects acorns.’

```

Yet, there seem to be some differences between the two constructions. For instance, the perception construction is referentially **transparent** whereas the factive construction is referentially **opaque**; see Barwise (1981) for related observations in English. So, if the first two statements are true in the perception construction (4.88), the third statement is inevitably true.

(4.88) Situation nominalisation (the perception construction)

- a. Satsuki-wa [otokonoko-ga hashi-tteiru no]-o mi-ta.
 S-TOP [boy-NOM run-PROG NO]-ACC see-PAST
 ‘Satsuki saw an event where the boy was running.’

- b. Sono-otokonoko-wa Kanta da-tta.
 that-boy-TOP K COP-PAST
 ‘The boy was Kanta.’
- c. Satsuki-wa [Kanta-ga hashi-tteiru no]-o mi-ta.
 S-TOP [K-NOM run-PROG NO]-ACC see-PAST
 ‘Satsuki saw an event where Kanta was running.’

By contrast, even if the first two statements in the factive construction (4.89) are true, the truth of the third statement is not guaranteed.

(4.89) Situation nominalisation (the factive construction)

- a. Satsuki-wa [otokonoko-ga toumorokoshi-o mottekita no]-o shitteiru.
 S-TOP [boy-NOM maize-ACC brought NO]-ACC know
 ‘Satsuki knows that the boy brought maize with him.’
- b. Sono-otokonoko-wa Kanta da-tta.
 that-boy-TOP K COP-PAST
 ‘The boy was Kanta.’
- c. Satsuki-wa [Kanta-ga toumorokoshi-o mottekita no]-o shitteiru.
 S-TOP [K-NOM maize-ACC brought NO]-ACC know
 ‘Satsuki knows that Kanta brought maize with him.’

Kim (2009: 349) reports that the same contrast is observed for the perception and the factive constructions in Korean (both of which are instances of *kes*-nominalisation). Kim accounts for this contrast (and other sets of contrast) in terms of the type of situations: the embedded clause in the perception construction denotes a ‘minimal’ situation, while the embedded clause in the factive construction denotes a ‘maximal’ situation. In our analysis, too, the same form of analysis may be possible if the notion of DS situation terms is made more sophisticated. This task, however, goes well beyond the scope of the present thesis.

3.5 Summary

The two types of *no*-nominalisation are dynamically modelled in terms of how the parser constructs a semantic structure progressively as it processes a string word-by-word. The crucial stage is the timing where the nominaliser *no* is parsed, at which the parser decides what type-e term it copies in order to decorate a fresh node: copying of an individual term leads to participant nominalisation whereas copying of a situation term leads to situation nominalisation. This uniform analysis predicts a number of *no*-nominalisation characteristics, including the corpus findings in Chap. 2.

Given the unitary nature of our account, the reader might expect that no crucial asymmetry will be detected between the two types of *no*-nominalisation. In the next section, however, I shall point out that they behave differently once coordinate clauses are nominalised by the particle *no*.

4 Nominalisation and Coordination

4.1 Introduction

In the last section, language dynamics integrated the two types of *no*-nominalisation. This section extends this account to novel data. Sect. 4.2 observes that when the pre-*no* clauses are coordinated by the suffix *i*, the two types of *no*-nominalisation exhibit a clear contrast. It is suggested that previous studies could not handle these data without sacrificing uniformity in explanation. Sect. 4.3 demonstrates that the data in participant nominalisation follow if the coordinate suffix *i* is assigned an appropriate lexical entry. Sect. 4.4 contends that the data in situation nominalisation are accounted for if tense is represented by substantiating the situation predicate *S* in situation term. Last, Sect. 4.5 summarises the main results.

4.2 The Asymmetry

One way of coordinating clauses is to mark a verb in the non-final clauses with the suffix *i*.²⁰ In (4.90), the verb in the first clause, *yogosh-* (= ‘dirty’), is marked by the particle *i*. In this case, only two clauses are coordinated, but if there is a third clause, the verb in the second clause, *yabui-* (= ‘tear’), will be suffixed by *i*.

(4.90) Clause coordination

Kanta-wa hon-o yogosh-i, Mei-wa sore-o yabui-ta.
 K-TOP book-ACC dirty-LINK M-TOP that-ACC tear-PAST
 ‘Kanta dirtied a book and Mei tore it.’

Below, we shall see what will happen if the pre-*no* part is a complex of multiple clauses coordinated by the clause-linking suffix *i*.

Firstly, consider participant nominalisation. In (4.91), the verb in the initial pre-*no* clause, *yogosh-* (= ‘dirty’), is marked by the suffix *i*, and the coordinated clauses are then nominalised by the particle *no*.

(4.91) Participant nominalisation

- [Kanta-ga yogosh-i, Mei-ga yabui-ta no]-o Satsuki-ga yon-da.
 [K-NOM dirty-LINK M-NOM tear-PAST NO]-ACC S-NOM read-PAST
- ‘Kanta dirtied something_i and Mei tore it_i. Satsuki read it_i.’
 - ‘Kanta dirtied some objects_i and Mei tore them_i. Satsuki read them_i.’
 - *‘Kanta dirtied something_i and Mei tore something else_j. Satsuki read them_{i+j}.’

In (4.91)a, the *no*-part denotes an object (e.g., book) that is dirtied by Kanta and torn by Mei. As shown in (4.91)b, the number of objects may be multiple, but in this case, **each** object must satisfy the description of **every** pre-*no* clause. For instance, if there are two books, each book must be dirtied by Kanta and torn by Mei. This reading is accessible when the numeral quantifier *ni-satsu* (= ‘two-CL’) is placed before *Satsuki*. In other

²⁰ The suffix *i* may be followed by another suffix *te* (Kuno 1973a: 195). This thesis concentrates on the *i* form.

words, as shown in (4.91)c, the *no*-part cannot denote an object which Kanta dirtied (but to which Mei did nothing) and another object which Mei tore (but to which Kanta did nothing). Thus, in participant nominalisation, every entity denoted by the *no*-part must satisfy all aspects of the pre-*no* part.²¹

This pattern, however, does not hold of situation nominalisation. Consider (4.92), where the two clauses are coordinated by the suffix *i* within the pre-*no* part.

(4.92) Situation nominalisation

[Totoro-ga uta-i, Chibi-Totoro-ga arui-ta no]-o Mei-ga mi-ta.

[T-NOM sing-LINK CT-NOM walk-PAST NO]-ACC M-NOM see-PAST

- a. 'Totoro sang and Chibi-Totoro walked simultaneously. Mei saw this event.'
- b. 'Totoro sang and Chibi-Totoro walked simultaneously. Mei saw this event twice.'
- c. 'Totoro sang at one time and Chibi-Totoro walked at another time. Mei saw these events.'

First, as in (4.92)a, the *no*-part may denote an event that consists of 'Totoro's singing' and 'Chibi-Totoro's walking.' Second, as in (4.92)b, the *no*-part may denote multiple events **each** of which consists of 'Totoro's singing' and 'Chibi-Totoro's walking.' This reading is accessible when the numeral quantifier *ni-kai* (= 'twice') is placed before the matrix verb *mi-* (= 'see'). Finally, (4.92)c shows that in situation nominalisation, the *no*-part may denote **distinct** events one of which solely consists of 'Totoro's singing' and the other of which solely consists of 'Chibi-Totoro's walking.' This interpretation is accessible when the adverbial noun *arutoki* (= 'at one time') is placed before each embedded verb: *uta-* (= 'sing') and *arui-* (= 'walk'). (Crucially, even if *arutoki* is used in (4.91), the reading (4.91)c is still unavailable.)

²¹ There is a putative counterexample. It seems that (4.91) may be felicitously uttered in the situation where Satsuki read a two-volume book, the first volume of which has been dirtied by Kanta and the second volume of which has been torn by Mei. In this case, there may appear to be two distinct objects (i.e., two volumes), but what is denoted is a single object, a two-volume book, one **part** of which is dirtied by Kanta and the other **part** of which is torn by Mei.

The contrast is summed up as follows:

- (4.93) Contrast between the two types of *no*-nominalisation
- a. In participant nominalisation, every entity denoted by the *no*-part **must** satisfy all aspects of the content expressed by the pre-*no* clauses.
 - b. In situation nominalisation, every entity denoted by the *no*-part **does not have to** satisfy all aspects of the content expressed by the pre-*no* clauses.

This generalisation has not been noted in the literature, but one may argue that it would be handled by the standard syntactic/semantics machinery. For instance, the participant nominalisation data (4.91) are reminiscent of across-the-board movement (Ross 1967), where a null item in each clause in a coordinate structure is extracted all-encompassing. In situation nominalisation (4.92), the available readings would be described in event semantics notations (Parsons 1990: Sect. 4.3.3).

For this form of analysis to work, however, some issues need to be settled. First, for across-the-board movement to operate, it must be assumed that null items are moved from all of the gap positions in participant nominalisation. It must also be assumed that such null items are absent from situation nominalisation; otherwise, the distinct-event reading (4.92)c would be unavailable. Still, both of these assumptions are contentious. As for the first, it was initially assumed that Japanese relatives involve null operator movement (e.g., Hasegawa 1988), but this view has been widely criticised; for instance, Mihara (1994) argues against the movement analysis by pointing out island-insensitivity of Japanese relatives. As for the second assumption, some suggest that the event argument position in situation nominalisation is occupied by a null operator (e.g., Kuroda 2005: 173). Furthermore, additional stipulations are required. For instance, in situation nominalisation, we must prohibit the operation ‘sum pairing,’ a process to turn two one-place predicates into a single one-place predicate that denotes a set whose member is a group of the elements of the sets denoted by the two one-place predicates

(Grosu & Landman 2012: 193); otherwise, the distinct-object reading (4.91)c would be incorrectly ruled in.

The standard syntax/semantics machinery may nevertheless be used to capture the asymmetry (4.93) **non-uniformly** (e.g., applying across-the-board movement only to participant nominalisation), but Sect. 2 has motivated a **uniform** treatment. In the previous literature, a unitary account of *no*-nominalisation has been largely unexplored (Kitagawa 2005, Kitagawa & Ross 1982, Murasugi 1991, Shibatani 2009, Tonoike 1990), and accounts are sketchy. The most detailed work is provided in Kitagawa (2005), but I pointed out in Sect. 3.4.2 that her analysis made a wrong prediction about expressive connotation in participant nominalisation. In addition, her analysis is incompatible with across-the-board movement since she assumes that the gap in participant nominalisation is a small *pro*. Thus, she must stipulate a new device to secure the unavailability of the distinct-object reading (4.91)c.

Given that various proposals have been presented in Mainstream Generative Grammar, it may be possible to account for the asymmetry (4.93) in a uniform manner, but it would end up stipulating new mechanisms or assumptions. By contrast, as will be shown in the following two sub-sections, the asymmetry (4.93) naturally falls out from our analysis if the issues of coordination and tense are addressed within DS.

4.3 Participant Nominalisation

The urgent task is to assign an entry to the clause-linking suffix *i*. This is attained in Sect. 4.3.1. Then, Sect. 4.3.2 provides an account of participant nominalisation, and Sect. 4.3.3 points out further predictions of the analysis.

4.3.1 Clausal Coordination

In the previous DS analysis of clausal coordination (Cann et al. 2005: Sect. 5.4.1, Cann et al. 2005), the parser first develops a propositional structure for the initial clause, LINKs it to another propositional structure, and fleshes it out by the parse of the second

clause. In defining the entry of the suffix i , I shall follow this previous view. That is, the suffix i is a LINK-inducing item which maps a structure of the initial clause onto the structure of the second clause. Recall that LINK is a **term-sharing** relation (see Sect. 5, Chap. 3). In the previous analysis (Cann et al. 2005: Sect. 5.4.1, Cann et al. 2005), however, a requirement for a shared content was obscure. In this thesis, I shall explicitly encode a content-sharing requirement in the entry of the suffix i .

(4.94) Entry of the clause-linking suffix i

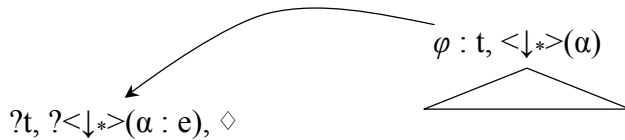
```

IF      t
THEN    IF       $\langle \downarrow_* \rangle(\alpha)$ 
        THEN    make/go( $\langle L \rangle$ ); put( $?t$ ,  $? \langle \downarrow_* \rangle(\alpha : e)$ )
        ELSE    ABORT
ELSE    ABORT

```

In plain English, if an active node is a type- t node and if there is a content α below the active node, the parser initiates a LINK relation to a new node and decorates it with $?t$ and $? \langle \downarrow_* \rangle(\alpha : e)$. The second requirement ensures that the paired trees will share the type- e content α . The attendant tree display is given in (4.95).

(4.95) Output structure of parsing the clause-linking suffix i

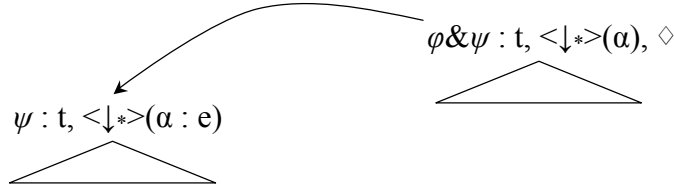


φ is the content of the initial clause, and $\langle \downarrow_* \rangle(\alpha)$ declares that there is a content α at a node within the tree. $? \langle \downarrow_* \rangle(\alpha : e)$ requires that the LINKed tree, to be developed by the second clause, will contain a type- e node decorated with the content α .

Once a LINKed tree is fully developed, the content of the LINKed tree is incorporated into the content of the main tree (Sect. 5, Chap. 3). In clausal coordination, this process takes the form of &-conjunction (Cann et al. 2005: Sect. 5.4.1, Cann et al.

2005), as formalised as the general action LINK ENRICHMENT. In (4.96), the proposition of the initial clause ϕ is conjoined with that of the second clause ψ , as in $\phi \& \psi$.

(4.96) LINK ENRICHMENT (for clausal coordination)



The lexical entry of the suffix *i* and the general action LINK ENRICHMENT will be illustrated in the next sub-section, where the participant nominalisation data with the coordinate pre-*no* clauses are analysed.

4.3.2 The Analysis

The entry of the suffix *i* (4.94) and the LINK ENRICHMENT process account for the available and the unavailable readings in participant nominalisation (4.91), repeated here as (4.97). The parse of (4.97) up to the suffix *i* generates the tree (4.98).

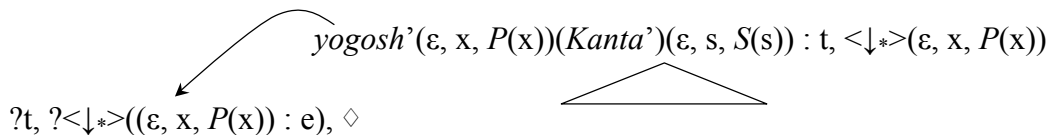
(4.97) Participant nominalisation

[Kanta-ga yogosh-i, Mei-ga yabui-ta no]-o Satsuki-ga yon-da.

[K-NOM dirty-LINK M-NOM tear-PAST NO]-ACC S-NOM read-PAST

- a. 'Kanta dirtied something_i and Mei tore it_i. Satsuki read it_i.'
- b. 'Kanta dirtied some objects_i and Mei tore them_i. Satsuki read them_i.'
- c. *'Kanta dirtied something_i and Mei tore something else_j. Satsuki read them_{i+j}.'

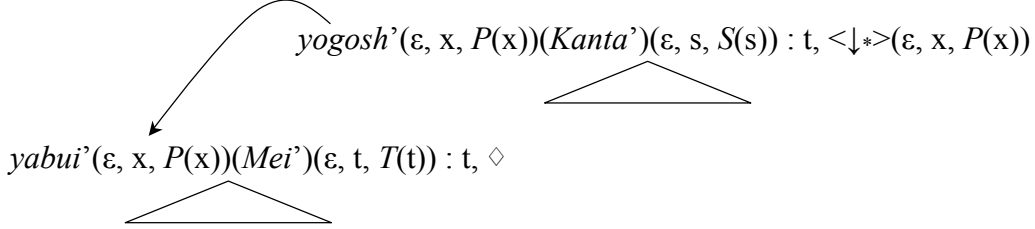
(4.98) Parsing *Kanta-ga yogosh-i*



In this display, α in (4.96) is instantiated as $(\epsilon, x, P(x))$, the content of the gap in the first pre-*no* clause. This ensures that the gap in the first pre-*no* clause will be **identical** to the

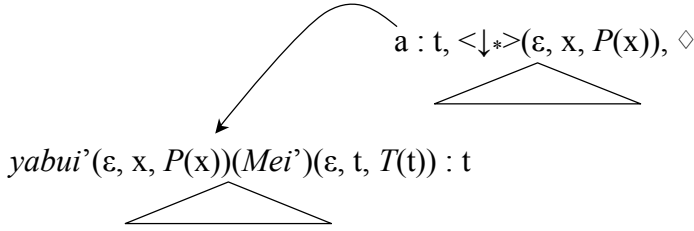
gap in the second pre-*no* clause. The LINKed node is developed by the second pre-*no* clause. In (4.99), $?<\downarrow^*>((\epsilon, x, P(x)) : e)$ has been met by the shared term $(\epsilon, x, P(x))$.

(4.99) Parsing *Kanta-ga yogosh-i, Mei-ga yabui-ta*



Since the LINKed tree is fully developed, LINK ENRICHMENT conjoins the proposition in the LINKed tree with the proposition in the main tree. In (4.100), the active node is decorated with the conjoined propositional formula a .

(4.100) LINK ENRICHMENT



where $a = \text{yogosh}'(\epsilon, x, P(x))(Kanta'(\epsilon, s, S(s)) \& \text{yabui}'(\epsilon, x, P(x))(Mei'(\epsilon, t, T(t)))$

The formula is then subject to QUANTIFIER EVALUATION (Q-EVALUATION). The term to be evaluated is $(\epsilon, x, P(x))$. Since this term is shared by the paired trees, Q-EVALUATION outputs a term picking out an arbitrary witness for **all** predicates in the conjoined propositions (Cann et al. 2005: 132). The evaluated term b denotes an object that is dirtied by Kanta and torn by Mei. In (4.101), the situation terms are dropped for simplicity's sake; for a full analysis of situation term, see Sect. 4.4.

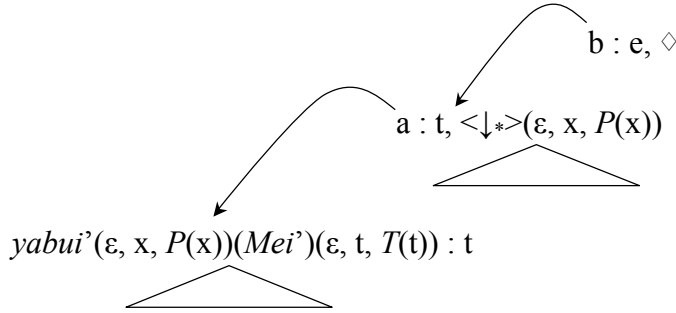
(4.101) Evaluating the conjoined proposition a in (4.100)

$$P(b) \& [\text{yogosh}'(b)(Kanta') \& \text{yabui}'(b)(Mei')]]$$

$$\text{where } b = (\epsilon, x, P(x) \& [\text{yogosh}'(x)(Kanta') \& \text{yabui}'(x)(Mei')])$$

What comes next is the nominaliser *no*, which inversely LINKs the current propositional node to a new node and decorates it with the evaluated term *b*.

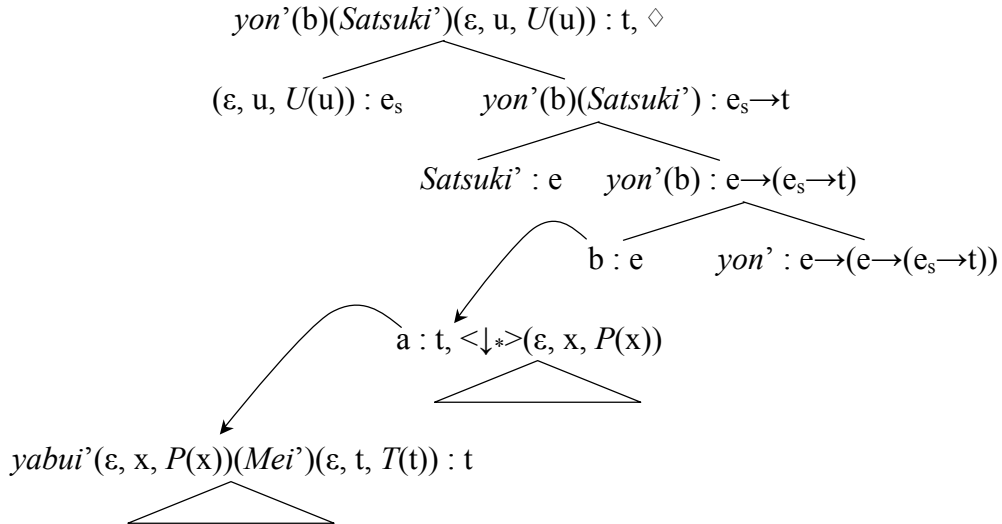
(4.102) Parsing *Kanta-ga yogosh-i, Mei-ga yabui-ta no*



where $a = yogosh'(\varepsilon, x, P(x))(Kanta')(\varepsilon, s, S(s)) \& yabui'(\varepsilon, x, P(x))(Mei')(\varepsilon, t, T(t))$
 $b = (\varepsilon, x, P(x) \& [yogosh'(x)(Kanta') \& yabui'(x)(Mei')])$

The rest of the process is as usual: the accusative-case particle *o* marks the current node as an object node; *Satsuki-ga* introduces a subject node; the matrix verb *yon-* (= 'read') projects a propositional structure. The final output is (4.103).

(4.103) Parsing the whole string (4.97)



where $a = yogosh'(\varepsilon, x, P(x))(Kanta')(\varepsilon, s, S(s)) \& yabui'(\varepsilon, x, P(x))(Mei')(\varepsilon, t, T(t))$
 $b = (\varepsilon, x, P(x) \& [yogosh'(x)(Kanta') \& yabui'(x)(Mei')])$

The evaluated individual term b denotes an object that is dirtied by Kanta and torn by Mei, modelling the reading (4.97)a. Since the term b is existentially quantified, it may denote multiple objects **each** of which is dirtied by Kanta and torn by Mei. Thus, it is also compatible with the reading (4.97)b. In Japanese, plurality is pragmatically inferred (Sect. 1, Chap. 2), and it is possible to use the operator ε_2 in constructing the term b , setting aside a full account of plurality (Cann et al. 2005: 284). The term with the operator ε_2 specifically models a case where Satsuki read two books each of which was dirtied by Kanta and torn by Mei.

The analysis also rules out the distinct-object reading (4.97)c. To model this reading, there must be **multiple** terms one of which denotes an object dirtied by Kanta and the other of which denotes another object torn by Mei. But the entry of *no* does **not** allow the parser to copy more than one term. One may wonder whether Q-EVALUATION could output a term that denotes a group of distinct objects, but such a term cannot be constructed because the term $(\varepsilon, x, P(x))$ must be **shared** by the two propositions, as required by the entry of the suffix *i*. Note that this requirement is not a stipulation but a general constraint imposed by LINK transitions: LINK is a formal pairing of discrete trees in virtue of a shared type-e element.

Thus, all and only available readings in participant nominalisation (4.97) are predicted. In particular, the absence of the distinct-object reading (4.97)c is a reflection of the nature of LINK as requiring a type-e content to be shared by the paired trees.

4.3.3 Further Predictions

Our analysis of coordination and participant nominalisation makes further predictions. First, consider (4.104), where the string is grammatical only with the distinct-object reading. This pattern is strikingly contrasted with that of (4.97), repeated as (4.105), where the grammaticality pattern is reversed. In (4.104), the suffix *i* is **absent** and the pre-*no* clauses lack a shared content. Therefore, for **each** pre-*no* clause, Q-EVALUATION outputs an evaluated individual term reflecting the content of a **single** pre-*no* clause;

these terms are turned into a group term by the NP coordinator *to*.²² Since this group term denotes a set of distinct objects, only the interpretation (4.104)c is engendered.

(4.104) Participant nominalisation

[[Kanta-ga yogoshi-ta no] to [Mei-ga yabui-ta no]]-o
 [[K-NOM dirty-PAST NO] and [M-NOM tear-PAST NO]]-ACC
 Satsuki-ga yon-da.
 S-NOM read-PAST

- a. *‘Kanta dirtied something_i and Mei tore it_i. Satsuki read it_i.’
- b. *‘Kanta dirtied some objects_i and Mei tore them_i. Satsuki read them_i.’
- c. ‘Kanta dirtied something_i and Mei tore something else_j. Satsuki read them_{i+j}.’

(4.105) Participant nominalisation

[Kanta-ga yogosh-i, Mei-ga yabui-ta no]-o Satsuki-ga yon-da.
 [K-NOM dirty-LINK M-NOM tear-PAST NO]-ACC S-NOM read-PAST

- a. ‘Kanta dirtied something_i and Mei tore it_i. Satsuki read it_i.’
- b. ‘Kanta dirtied some objects_i and Mei tore them_i. Satsuki read them_i.’
- c. *‘Kanta dirtied something_i and Mei tore something else_j. Satsuki read them_{i+j}.’

Second, our analysis predicts that **any** number of pre-*no* clauses may be coordinated. This is because any number of LINK relations may be introduced as long as each LINK transition is driven by the suffix *i*. In (4.106), three pre-*no* clauses are coordinated. As shown in this example, if more than two clauses are coordinated, a verb in each non-final pre-*no* clause **must** have the suffix *i*, while a verb in the final pre-*no* clause **cannot** have the suffix *i*. In each non-final pre-*no* clause, a LINK relation is induced every time the suffix *i* is parsed, and the LINKed node is then developed by the following pre-*no* clause. In the final pre-*no* clause, if the suffix *i* is present, the LINKed node introduced will not be developed because there is no following clause. Then, any requirements at the LINKed node will be left unsatisfied, and the string is deemed to be

²² The NP coordinator *to* may be seen as a LINK-inducing device, relating the type-e node to another type-e node. It is also possible to define the LINK ENRICHMENT rule, which takes multiple terms and outputs a group term consisting of these terms.

ungrammatical. The distribution of the suffix *i* in (4.106) is accounted for as an outcome of incremental tree-growth: every time a LINK relation is introduced by the suffix *i*, the relation must be elaborated by the immediately following clause during the course of left-to-right processing.

(4.106) Participant nominalisation

[Kanta-ga yogos-i, Mei-ga yabuk-i, Satsuki-ga

[K-NOM dirty-LINK M-NOM tear-LINK S-NOM

chigi-tta/*chigir-i no]-o Yasue-ga yon-da.

tear-PAST/tear-LINK NO]-ACC Y-NOM read-PAST

‘Kanta dirtied something_i, Mei tore it_i, and Satsuki tore it_i. Yasue read it_i.’

4.4 Situation Nominalisation

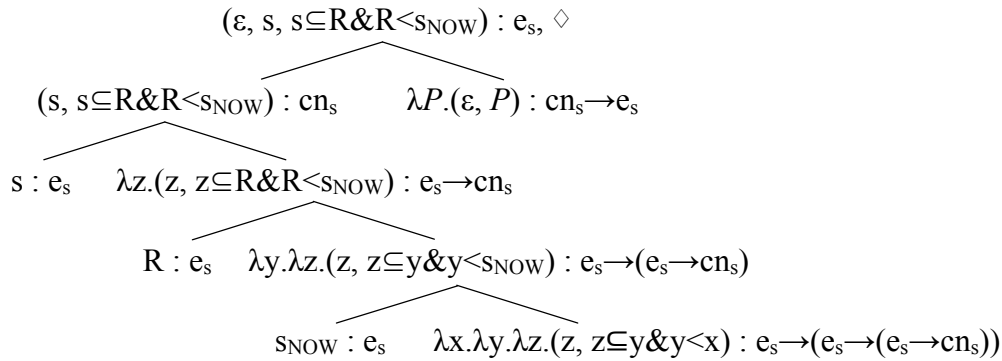
The entry of the clause-linking suffix *i* requires that a type-e element will be shared by paired trees. In participant nominalisation, a shared element is the content of a gap. In situation nominalisation, however, there is no gap, and a question arises of whether the entry of the suffix *i* should be modified. The answer is negative, if the issue of tense is handled within the present framework.

4.4.1 Representing Tense

So far, situation terms are notated as, say, $(\epsilon, s, S(s))$. S is a situation predicate, but its actual content has been ignored. Cann (2011) proposes to express tense by making a restrictor of situation terms more fine-grained. For instance, a situation term for a past event is $(\epsilon, s, s \subseteq R \& R < s_{\text{NOW}})$, with the internal structure (4.107).²³

²³ In this representation, the node for the utterance time constant s_{NOW} and the node for its functor are explicitly articulated. This complication is not addressed in Cann (2011).

(4.107) Situation term (modelling a past event)



Tense is expressed as restrictors on the situation variable s ; R is a time interval, s_{NOW} is an utterance time, $\subseteq$ is an inclusion relation, and $<$ is a precedence relation. R is **underspecified** and needs to be contextually saturated as, say, *yesterday*. In the case of a past event (4.107), the time interval R precedes the utterance time s_{NOW} and the event time is contained in R . This is a simplified account of tense (cf., Reichenbach 1947) but it is enough for current purposes.²⁴ This DS account of tense, together with the entry of the suffix *i*, captures various readings in situation nominalisation.

4.4.2 The Analysis

An example of situation nominalisation (4.92) is reproduced here as (4.108).

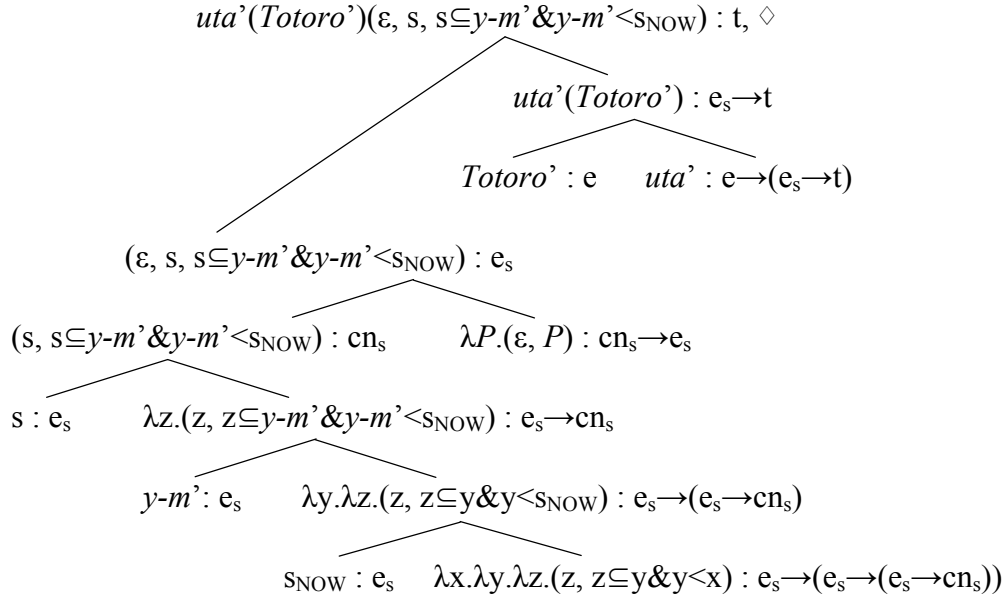
(4.108) Situation nominalisation

- [Totoro-ga uta-i, Chibi-Totoro-ga arui-ta no]-o Mei-ga mi-ta.
 [T-NOM sing-LINK CT-NOM walk-PAST NO]-ACC M-NOM see-PAST
- 'Totoro sang and Chibi-Totoro walked simultaneously. Mei saw this event.'
 - 'Totoro sang and Chibi-Totoro walked simultaneously. Mei saw this event twice.'
 - 'Totoro sang at one time and Chibi-Totoro walked at another time. Mei saw these events.'

²⁴ Ogihara (1999) shows that the past tense suffix *ta* is a 'relative tense' marker. Thus, I will ultimately have to modify the restrictor of situation term, but this issue can be sidestepped in the present analysis. This is because when both embedded and matrix predicates are in the past tense form, the tense behaves as if it were 'absolute.' In fact, some argue that tense is 'absolute' when predicates in the relative and the matrix clauses have the same tense form (Mihara 1992).

The parse of the initial clause up to the verb *uta-* (= ‘sing’) yields the tree (4.109). The term $(\epsilon, s, s \subseteq y\text{-}m' \& y\text{-}m' <_{\text{SNOW}})$ models a past event, where an underspecified time interval R has been pragmatically resolved as *yesterday-morning*, abbreviated as *y-m*. (The internal structure of the situation term is explicitly represented in (4.109).)

(4.109) Parsing *Totoro-ga uta-*



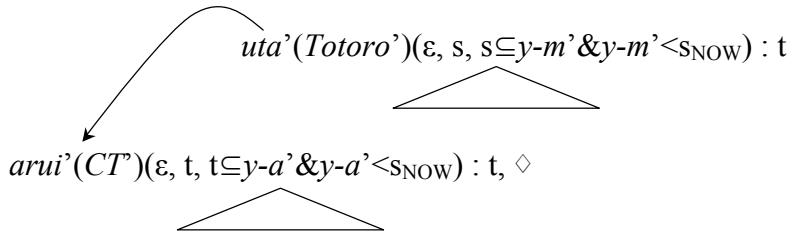
The next item to be parsed is the clause-linking suffix *i*. According to the entry (4.94), repeated here as (4.110), the paired trees must share a type- e element. That is, $\langle \downarrow_* \rangle(\alpha)$ declares that the current tree must contain a node which is decorated with a type- e element α , and $? \langle \downarrow_* \rangle(\alpha : e)$ declares that the LINKed tree will contain a node which is annotated with the type- e element α .

(4.110) Entry of the clause-linking suffix *i*

IF	t	
THEN	IF	$\langle \downarrow_* \rangle(\alpha)$
	THEN	make/go($\langle L \rangle$); put($?t, ? \langle \downarrow_* \rangle(\alpha : e)$)
	ELSE	ABORT
ELSE	ABORT	

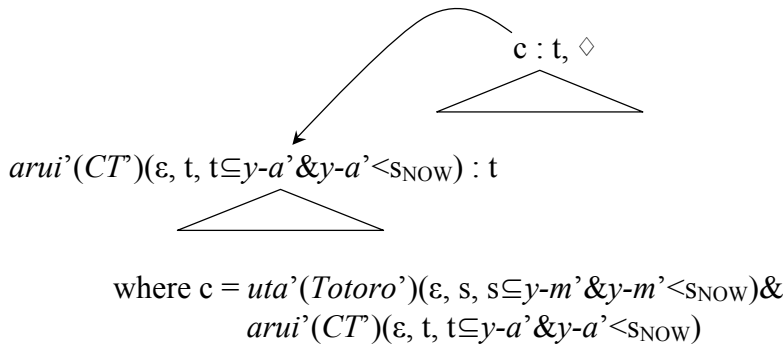
The question is what this shared type-e element would be. In the tree (4.109), note that the utterance time s_{NOW} is a **type-e** constant. I propose that it is the utterance time constant s_{NOW} that is shared by the paired trees in situation nominalisation.²⁵ This is conceptually intuitive because the coordinate clauses in (4.108) are uttered at **the same** point of time, regardless of whether an event described by the initial pre-*no* clause coincides with an event described by the second pre-*no* clause. Thus, (4.109) is updated into (4.111); the paired trees share the type-e constant s_{NOW} , and the underspecified time interval in the LINKed tree is saturated as *yesterday-afternoon*, abbreviated as $y-a'$.

(4.111) Parsing *Totoro-ga uta-i, Chibi-Totoro-ga arui-ta*



Since discrete trees have been paired, LINK ENRICHMENT conjoins the two propositions and represents the resulting formula at the top node of the main tree. In (4.112), the conjoined propositions are abbreviated as the formula c .

(4.112) LINK ENRICHMENT



²⁵ This idea is based on an exchange with Ruth Kempson. I am grateful for her suggestion.

The conjoined proposition is then subject to Q-EVALUATION. Suppose that the situation term of the initial *pre-no* clause out-scopes the situation term of the second *pre-no* clause. Then, the latter is first evaluated and the former is evaluated afterwards.

(4.113) Evaluating the conjoined proposition *c* in (4.112)

$$S(b) \& [T(a_b) \& [uta'(Totoro')(b) \& arui'(CT')(a_b)]]$$

$$\text{where } b = (\epsilon, s, S(s) \& [T(a_s) \& [uta'(Totoro')(s) \& arui'(CT')(a_s)]])$$

$$a_b = (\epsilon, t, T(t) \& [uta'(Totoro')(b) \& arui'(CT')(t)])$$

$$a_s = (\epsilon, t, T(t) \& [uta'(Totoro')(s) \& arui'(CT')(t)])$$

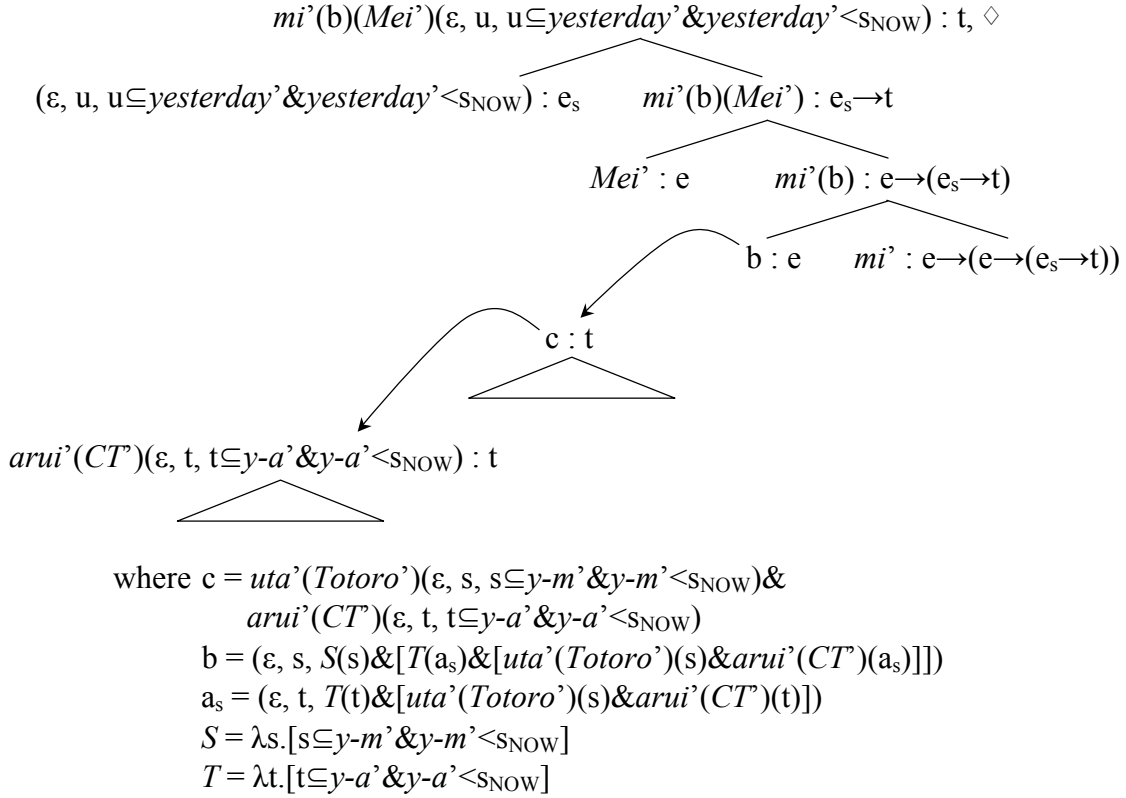
$$S = \lambda s. [s \subseteq y-m' \& y-m' <_{\text{NOW}}]$$

$$T = \lambda t. [t \subseteq y-a' \& y-a' <_{\text{NOW}}]$$

The representation is complicated, but what is essential is that the evaluated situation terms *b* and *a_b* reflect the content of the **whole** *pre-no* parts. Both of them denote a situation where Totoro sang during the interval *y-m'* (= *yesterday-morning'*) and Chibi-Totoro walked during the interval *y-a'* (= *yesterday-afternoon'*).

Next, the nominaliser *no* inversely LINKs the active node in (4.112) to a new node and decorates it with either the situation term *b* or *a_b*. The selection of the term does not lead to truth-conditional differences in the final output. Let us then suppose that the situation term *b* is selected. This node is resolved as an object node by the accusative-case particle *o* within a new structure; this structure is fleshed out by the rest of the string (4.108). A situation meta-variable in this main structure is pragmatically substituted with $(\epsilon, u, u \subseteq \text{yesterday}' \& \text{yesterday}' <_{\text{NOW}})$. (The internal structure of this situation term is neglected.) The final state is given in (4.114).

(4.114) Parsing the whole string (4.108)



As stated in Sect. 4.4.1, a time interval R is initially underspecified. I propose that the indeterminacy of R is a source of the multiple readings in (4.108). First, in (4.114), for each pre-*no* clause, an interval has been fixed **differently**: one is $y-m'$ (= *yesterday-morning'*) and the other is $y-a'$ (= *yesterday-afternoon'*). This models the reading (4.108)c, where the two events described by the pre-*no* clauses took place at different times. Second, suppose that these intervals have been fixed **identically**; for instance, both intervals could have been set as *yesterday-morning'*. This models the reading (4.108)a, where the two events described by the pre-*no* clauses took place at the same time. The upshot is that the readings (4.108)a and (4.108)c are reduced to two different ways of fixing underspecified intervals in the situation terms of the pre-*no* clauses (i.e., saturating them identically or differently).

Then, how about the reading (4.108)b? In this case, Mei saw two events each of which consists of Totoro's singing and Chibi-Totoro's walking. Despite there being

no formal specification here of plurality, nevertheless it is in principle clear how this reading is captured in our analysis. In the case of individual terms, we have simply notated the plural operator as, say, ε_2 (Cann et al. 2005: 284). I assume that plurality in events is expressed in the same manner. Thus, the situation term of the matrix clause may be expressed as $(\varepsilon_2, u, u \subseteq \text{yesterday}' \& \text{yesterday}' <_{\text{SNOW}})$. If the time intervals in the pre-*no* clauses are set identically as, say, *yesterday-morning'*, the interpretation (4.108)b follows: ‘Mei saw two events (possibly, at different times during the interval *yesterday-morning'*), each of which consists of Totoro’s singing and Chibi-Totoro’s walking.’²⁶

In a nutshell, all of the available readings in situation nominalisation (4.108) are predicted in our analysis once the situation predicate *S* is substantiated. In order to provide a comprehensive account, however, a DS theory of plurality must be developed.

4.4.3 Residual Issues

There are some remaining problems of the present account. First, it was argued that the suffix *i* required that the paired trees would share a type-e element. In participant nominalisation, it is the content of a gap, whereas in situation nominalisation, it is the utterance time constant. Yet, the utterance time constant is also present in participant nominalisation, and a question arises of how to prevent the utterance time constant from being shared by the paired trees in participant nominalisation. This blocking mechanism may be encoded as a further constraint in the entry of the suffix *i*.²⁷

Second, in situation nominalisation (4.108), we concentrated on the temporal relation between the two pre-*no* clauses. But coordinate clauses may express other kinds

²⁶ It is possible to saturate the time intervals in the pre-*no* clauses differently as, say, *yesterday-morning'* (for the first pre-*no* clause) and *yesterday-afternoon'* (for the second pre-*no* clause). In this case, Mei saw two pairs of events, each pair consisting of Totoro’s singing yesterday morning and Chibi-Totoro’s walking yesterday afternoon. This is a variant of (4.108)c.

²⁷ In fact, the entry of the suffix *i* presented in this thesis needs revision. Saiki (1985) observes that when multiple clauses are coordinated in participant nominalisation, there are prohibited combinations of the grammatical functions of gaps. Saiki (1985) considers the *te*-coordination alone (see footnote 20) and it should be clarified what pattern holds of the *i*-coordination. To account for whatever pattern is observed, the entry of the suffix *i* will have to be augmented with a further set of constraints.

of relation (e.g., causal). In the literature, it is widely assumed that when clauses are juxtaposed, their relation is pragmatically determined (Carston 2002: Chap. 3). I shall follow this general stance, presuming that non-temporal relations can be represented if restrictors in situation terms are enriched with underspecified relations (which are saturated pragmatically).

4.5 Summary

Participant and situation nominalisation behave differently when the pre-*no* part is coordinate clauses. In participant nominalisation, the entity denoted by the *no*-part must reflect the description of every pre-*no* part. This is because Q-EVALUATION outputs a term which denotes an object reflecting the content of the whole pre-*no* part. A group term denoting distinct objects cannot be created due to the requirement that the content of a gap in each pre-*no* clause must be identical, as required by the entry of the clause-linking suffix *i*. This requirement is not a stipulation but a general constraint imposed by LINK transitions (i.e., sharing of a type-e element). In situation nominalisation, the pre-*no* part does not have to reflect the description of every pre-*no* clause. All of the available interpretations in situation nominalisation are predicted on the basis of how underspecified time intervals are saturated.

5 Conclusion

This chapter has started by justifying a unitary treatment of the particle *no* in the two types of *no*-nominalisation from functional, diachronic, and cross-linguistic/dialectal standpoints. In the previous literature, some attempts have been made to theoretically unify the two types of *no*-nominalisation, but no detailed analysis has been presented as yet. Our claim is that language dynamics is the key to integrating the phenomena. In our unitary account, the particle *no* is assigned a single entry as a nominaliser; *no* encodes

the instruction to take the evaluated propositional content of the pre-*no* clause and to output a type-*e* term that denotes an entity reflecting the propositional content (Cann et al. 2005: 285). The two types of nominalising functions then boil down to a parser's decision of what type-*e* term it copies at the time of parsing *no*. That is, if the parser copies an individual term, participant nominalisation emerges, while if the parser copies a situation term, situation nominalisation emerges. A number of characteristics of *no*-nominalisation (including the corpus data in Chap. 2) follow from our unified analysis.

Moreover, we have observed that the two types of *no*-nominalisation exhibit a clear discrepancy when the pre-*no* part is coordinate clauses. The *no*-part in participant nominalisation must satisfy every description of the pre-*no* clauses, while the *no*-part in situation nominalisation does not have to satisfy every description of the pre-*no* clauses. It is suggested that this asymmetry would pose a problem for the standard syntactic and semantic analyses within Mainstream Generative Grammar. By contrast, the asymmetry naturally follow from the dynamic account if (i) the clause-linking suffix *i* is viewed as a LINK-inducing device to pair two propositional trees (cf., Cann et al. 2005) and (ii) if the restrictor of situation terms is substantiated so as to represent tense (Cann 2011). In participant nominalisation, the absence of a distinct-object interpretation is due to the general feature of a LINK relation as a tree-mapping based on a shared type-*e* term. In situation nominalisation, various interpretations may be reduced to a parser's choice of how it resolves an underspecified time interval in situation terms.

This chapter has explicated the lexical encoding of the nominaliser *no* with reference to language dynamics, and developed a unitary account of *no*-nominalisation. This is a solid basis for addressing various *no*-involving constructions, as we shall see in the rest of the thesis.

Chapter 5

The Dynamics of Cleft Constructions

1 Overview

The last chapter developed an account of *no*-nominalisation in tune with language dynamics, as formalised within the framework of Dynamic Syntax (DS) (Cann et al. 2005, Kempson et al. 2001, Kempson et al. 2011; see Chap. 3). This chapter bolsters the significance of language dynamics in linguistic theorising by examining another *no*-involving phenomenon: cleft constructions.

Clefts have been the subject of a number of studies cross-linguistically. This is because the data displayed by clefts range from syntax, semantics to pragmatics, as well as other branches of linguistics. Arguably, the most extensively studied language is English: for syntax, see Akmajian (1970), Chomsky (1977), and subsequent work; for semantics, see Atlas & Levinson (1981), Horn (1981), and subsequent work; for pragmatics, see Declerck (1988), Prince (1978), and subsequent work. However, a growing body of research has also been conducted for other languages; to take syntax as an example, see Kitagawa & Ross (1982) and Li (2008) for Mandarin Chinese, Harbour (2008) for Haitian, Kang (2006) and Kim & Lee (2008) for Korean, Kiss (1998) for Hungarian, Paul (2001) for Malagasy, Kempson et al. (2011) for SiSwati, etc.

This chapter concentrates on the syntactic aspects of clefts in Contemporary Japanese.¹ In the literature, it has been assumed that Japanese clefts are divided into two types, depending on whether a focus element has a case particle (Hoji 1990).

- (5.1) [Chihiro-ga e_i tabeta no]-wa onigiri mi-ttsu_i da.
 [C-NOM ate NO]-TOP rice.ball 3-CL COP
 ‘It is three rice balls that Chihiro ate.’

- (5.2) [Chihiro-ga e_i tabeta no]-wa onigiri-o mi-ttsu_i da.
 [C-NOM ate NO]-TOP rice.ball-ACC 3-CL COP
 ‘It is three rice balls that Chihiro ate.’

The embedded clause *Chihiro-ga tabeta* (= ‘Chihiro ate’) contains the gap *e_i*. This embedded part (often called ‘presuppositional clause’ (Kizu 2005)) is nominalised by the particle *no* and topicalised by the particle *wa*. With respect to this topic, the focus *onigiri* (= ‘rice ball’) is subsequently processed and the string ends with the copula *da*-. The difference between (5.1) and (5.2) is that the accusative-case particle *o* is attached to the focus only in (5.2). Clefts with a **case-less** focus as in (5.1) are called clefts_{-C}, while clefts with a **case-marked** focus as in (5.2) are called clefts_{+C}.²

It has been observed that there are some syntactic differences between the two types of clefts. One of the most marked distinctions is that multiple foci are licensed only in clefts_{+C} (Koizumi 1995). Another well-studied difference is that clefts_{+C} but not clefts_{-C} are sensitive to syntactic islands (Hoji 1990). Given these sets of asymmetries, previous studies have assigned different syntactic structures to clefts_{+C} and clefts_{-C}. In Sect. 5, Chap. 2, however, it was shown based on the corpus data that the case-marking of a focus was affected by pragmatic factors. It was then argued that an account of clefts

¹ For clefts in non-Contemporary Japanese, see Kondo (2000) and Wrona (2005).

² As pointed out in Sect. 5, Chap. 2, clefts with an accusative-case marked focus are degraded for some speakers. But they are acceptable when a numeral quantifier is present, as in *onigiri-o mi-ttsu* (= ‘3 rice balls’) or when there are multiple foci (Koizumi 1995).

must be **neutral** to the presence of case particles, contrary to what has been assumed in previous works.

There are also empirical puzzles that undermine previous studies. As briefly stated above, it has been widely observed that multiple foci are possible in clefts_{+C} but not in clefts_{-C} (e.g., Cho et al. 2008, Hiraiwa & Ishihara 2012, Koizumi 1995).

(5.3) Clefts_{+C}

[Haku-ga	e _i	e _j	age-ta	no]	-wa	Chihiro-ni _j	onigiri-o _i	da.
[H-NOM			give-PAST	NO]	-TOP	C-DAT	rice.ball-ACC	COP

Lit. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

(5.4) Clefts_{-C}

*[Haku-ga	e _i	e _j	age-ta	no]	-wa	Chihiro _j	onigiri _i	da.
[H-NOM			give-PAST	NO]	-TOP	C	rice.ball	COP

What has been overlooked is that, although case particles have been presumed to be obligatory in multiple foci, a second focus, but not a first focus, may be case-less:

(5.5) Clefts with partially case-marked foci

[Haku-ga	e _i	e _j	age-ta	no]	-wa	Chihiro-ni _j	onigiri _i	da.
[H-NOM			give-PAST	NO]	-TOP	C-DAT	rice.ball	COP

Lit. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

(5.6) Clefts with partially case-marked foci

*[Haku-ga	e _i	e _j	age-ta	no]	-wa	Chihiro _j	onigiri-o _i	da.
[H-NOM			give-PAST	NO]	-TOP	C	rice.ball-ACC	COP

These partially case-marked foci data have not been noted elsewhere aside from Hiraiwa & Ishihara (2012: 146), who make a passing remark but without analysis. These phenomena as well as other new data to be provided in later sections pose obstacles to previous studies within Mainstream Generative Grammar (see Sect. 2).

Thus, Japanese clefts raise two issues: the theoretical problem of how to unify the two types of clefts and the empirical problem of how to account for a range of data problematic for previous studies (e.g., partially case-marked foci). The aim of this chapter is to show that these issues are fruitfully addressed from the perspective of language dynamics.

Sect. 2 surveys previous studies. Sect. 3 develops a unified analysis of clefts and accounts for a number of cleft properties, including those exemplified by corpus data. It will be argued that the two types of clefts differ in how the content of a focus is incorporated into the main tree, with this difference arising from the surface presentation of case particles (Seraku 2011). Sect. 4 turns to the issue of multiple foci. In particular, partially case-marked foci are explained in terms of how a node for each focus may be resolved during the course of incremental parsing (Seraku 2012a, 2013b). Sect. 5 untangles the issue of long-distance dependencies and islands. Long-distance clefts are handled by the feeding relation of distinct actions to introduce an unfixed node. The analysis also accounts for why clefts_{+C} are sensitive to islands but clefts_{-C} are not; it is claimed that the focus may be parsed on the basis of LINK transitions, and this LINK-based tree update reaches a well-formed state only when the focus is case-less. Sect. 6 extends our account to ellipsis constructions. We observe and explain affinities between clefts, stripping, and sluicing with respect to partial case-marking. In addition, the island-sensitivity pattern of clefts repeats itself in stripping and sluicing (Fukaya 2007), and it is amenable to our uniform analysis. Finally, Sect. 7 concludes the chapter.

2 Previous Studies

2.1 Introduction

Japanese clefts have been analysed in Mainstream Generative Grammar (MGG). An alternative view is Gunji (2007), which is based on Discourse Representation Theory

(Kamp & Reyle 1993). The pioneer of research on Japanese clefts within MGG is Hoji (1990), though there were insightful works previously (Harada 1973, Inoue 1976, Kitagawa & Ross 1982, Nakau 1973, Okutsu 1974). After Hoji (1990), past studies have been divided into two strands. One line of research has elaborated Hoji's account (Hasegawa 1997, Kizu 2005, Matsuda 1997, 2000, Murasugi 1991, Yoshimura 2001, Yoshimura & Nishina 2004). Another line of research has explored clefts from different angles, often invoking different types of movement operations (Cho et al. 2008, Fukui & Sakai 2003, Hiraiwa & Ishihara 2002, 2012, Koizumi 1995, 2000, Kuwabara 1996, Saito 2004, Takano 2002, Yamada 2004). Influential work in the former camp is Kizu (2005) and that in the latter camp is Hiraiwa & Ishihara (2012). This section critically reviews Hoji (1990), Kizu (2005), and Hiraiwa & Ishihara (2012) in turn.

2.2 Hoji (1990)

Hoji (1990) is an important contribution to research on the syntax of Japanese clefts. His insights are twofold as follows.

Firstly, it is observed that the presence of a case particle attached to a focus affects island-sensitivity. Thus, the cleft_{+C} (5.7), where the focus item *kono-fuku* has the accusative-case particle *o*, is sensitive to syntactic islands, while the cleft_{-C} (5.8), where the case particle is absent, is **not** sensitive to syntactic islands. Here, the island is the complex NP *ki-teiru hito* (i.e., the NP modified by the relative clause).

(5.7) Clefts_{+C}³

*[[_{NP} [_{CP} e_i e_j ki-teiru] hito_i]-ga shiawase-souna no]-wa
 [[[wear-PRES] person]-NOM happy-look NO]-TOP
 kono-fuku-o_j da.
 this-cloth-ACC COP
 Int. ‘It is this cloth x_j that the person who wears x_j looks happy.’

(5.8) Clefts_{-C}

[[_{NP} [_{CP} e_i e_j ki-teiru] hito_i]-ga shiawase-souna no]-wa
 [[[wear-PRES] person]-NOM happy-look NO]-TOP
 kono-fuku_j da.
 this-cloth COP
 Lit. ‘It is this cloth x_j that the person who wears x_j looks happy.’

Later works (e.g., Hiraiwa & Ishihara 2012, Koizumi 1995, Kuroda 2005) have pointed out that the two types of clefts also differ with respect to multiple foci: thus, as shown by (5.3)-(5.4), multiple foci are possible only in clefts_{+C}.

Secondly, Hoji contends that the contrast concerning islands indicates that movement is involved **only** in clefts_{+C}. More specifically, clefts_{+C} are assigned the structure (5.9) while clefts_{-C} are assigned the structure (5.10).

(5.9) Structure for clefts_{+C}

[_{CP} OP_i [_{IP} ... t_i ... V] no]-wa NP-case_i da

(5.10) Structure for clefts_{-C}

[_{NP} [_{IP} ... pro_i ... V] no]-wa NP_i da

In (5.9), *no* is a complementiser and the null operator OP moves from the theta position t_i to Spec CP. NP-case is a focus and is co-indexed with OP (and ultimately, with the trace) through a ‘predication relation’ (Williams 1980). In this derivation, the movement

³ Clefts with an accusative-case marked focus are degraded for some speakers (see footnote 2). Crucially, the island-involving cleft_{+C} (5.7) is unacceptable even for those who judge clefts with an accusative-case marked focus to be acceptable.

of OP is responsible for island effects. By contrast, movement is not involved in (5.10), hence no island effects; *no* is a pronominal and the co-reference relation between *pro* and NP is licensed by an ‘aboutness condition’ (Kuno 1973a).

In essence, Hoji (1990) syntactically distinguishes between clefts_{+C} and clefts_{-C} in virtue of the distinct island effects. Hoji’s insights should certainly be acknowledged, but this structurally bifurcated account is not plausible in the light of corpus findings (Sect. 5, Chap. 2). Furthermore, it seems that his analysis fails to model partially case-marked foci (5.5)-(5.6), repeated here as (5.11)-(5.12).

(5.11) Clefts with partially case-marked foci

[Haku-ga e_i e_j age-ta no]-wa Chihiro-ni_j onigiri_i da.
 [H-NOM give-PAST NO]-TOP C-DAT rice.ball COP
 ‘It is rice balls_i to Chihiro_j that Haku gave e_i e_j.’

(5.12) Clefts with partially case-marked foci

*[Haku-ga e_i e_j age-ta no]-wa Chihiro_j onigiri-o_i da.
 [H-NOM give-PAST NO]-TOP C rice.ball-ACC COP

In each example, a case-marked focus item and a case-less focus item appear at the same time within a single structure. It is not obvious how *no* in (5.11)-(5.12) could be characterised according to Hoji (1990), who claims that *no* is a complementiser in clefts_{+C} and *no* is a pronominal in clefts_{-C}. Even if a stipulation is made that *no* in (5.11)-(5.12) has a dual function as complementiser and as pronominal, the question remains as to why it rules in (5.11) but not (5.12). Thus, Hoji (1990) cannot cope with partially case-marked foci.

2.3 Kizu (2005)

Building on Hoji (1990), Kizu (2005) claims that (i) **both** clefts_{+C} and clefts_{-C} involve the movement of a null operator and that (ii) *no* is **uniformly** seen as a complementiser.

First, clefts_{+C} are assigned the structure (5.9), repeated here as (5.13). Note that (5.13) involves syntactic movement, hence island-sensitivity of clefts_{+C}.

(5.13) Structure for clefts_{+C}

[_{CP} OP_i [_{IP} ... t_i ... V] no]-wa NP-case_i da

Second, the analysis of clefts_{-C} is based on Kuroda (1986) and especially Sakai (1994). It is widely assumed that Japanese relatives are not island-sensitive (Kuno 1973a), but Sakai (1994) argues that Japanese relatives are, indeed, sensitive to islands, and that the alleged island-insensitivity comes from the fact that Japanese exhibits a so-called major-subject construction, where there are multiple nominative-case-marked items in a single clause. Kizu (2005) applies this insight to Japanese clefts, and proposes that the derivational source of the cleft_{-C} (5.8), repeated here as (5.14), is (5.15), which is an instance of the major-subject construction.

(5.14) Clefts_{-C}

[[_{NP} [_{CP} e_i e_j ki-teiru] hito_i]-ga shiawase-souna no]-wa
 [[[wear-PRES] person]-NOM happy-look NO]-TOP
 kono-fuku_j da.
 this-cloth COP
 Lit. 'It is this cloth x_j that the person who wears x_j looks happy.'

(5.15) The major-subject construction: Derivational source of (5.14)

Kono-fuku_j-ga [_{NP} [_{CP} e_i e_j ki-teiru] hito_i]-ga shiawase-souda.
 this-cloth_j-NOM [[wear-PRES] person]-NOM happy-look
 Lit. 'This cloth x_j, the person who wears x_j looks happy.'

Based on (5.15), the cleft_{-C} (5.14) is assigned the structure (5.16). The null operator OP moves to Spec CP from t_j, a major-subject position **outside** the complex NP island, and the OP and the focus are co-indexed by a 'predication relation' (Williams 1980). In this

derivation, the movement of the OP does not cross any island boundaries, hence alleged island-insensitivity of clefts_{-C}.

(5.16) Kizu's analysis of the cleft_{-C} (5.14)

[_{CP} OP_j [_{IP} t_j [_{NP} [_{CP} e_i e_j ki-teiru] hito_i]-ga
 [[[[wear-PRES] person]-NOM
 shiawase-souna] no]-wa kono-fuku_j da.
 happy-look] NO]-TOP this-cloth COP

Lit. 'It is this cloth x_j that the person who wears x_j looks happy.'

Kizu goes on to claim that this major-subject-based derivation is not available for clefts_{+C}; if *kono-fuku* had the accusative-case particle *o*, the ill-formed constituent *kono-fuku-o-ga* would be formed. Therefore, this type of derivation cannot be a derivational source of clefts_{+C}, and thus island-sensitivity of clefts_{+C} follows.

What is significant in Kizu (2005) is that the two types of clefts are uniformly handled in the sense that both types of clefts involve movement and that *no* is unitarily viewed as a complementiser. But her analysis is not truly uniform because the two types of clefts have different derivational sources, contrary to the corpus indication that the two types of clefts are not discernable structurally (Sect. 5, Chap. 2). Moreover, as criticised in Hiraiwa & Ishihara (2012), Kizu (2005) cannot guarantee the case-matching effect between a verb in a presupposition clause and a focus because the predication relation has nothing to do with case-marking. The main problem for Kizu's (2005) analysis from the standpoint of the present thesis is that it cannot license multiple foci, let alone partially case-specified foci: Kizu (2005: 54) assumes that (i) the copula *da-* is a contracted form of the postposition *de* and the verb *aru-* (= 'exist') and that (ii) the postposition *de* selects only a single argument. Kizu states that her analysis models a grammar which disallows multiple foci (Inoue 1976: 101), but a growing body of research has indicated that multiple foci are possible in Japanese clefts (e.g., Cho et al. 2008, Hiraiwa & Ishihara 2012). Thus, Kizu (2005) has at best limited applicability.

2.4 Hiraiwa & Ishihara (2002, 2012)

Expanding on Hiraiwa & Ishihara (2002), Hiraiwa & Ishihara (2012) advocate an analysis of clefts_{+C} according to which *no* is a complementiser. As for clefts_{-C}, it is suggested that they are assigned a **distinct** structure, where *no* is a pronominal.

It is proposed that clefts_{+C} such as (5.17) are derived from *no-da* sentences (also called ‘in-situ focus’ sentences) such as (5.18).

(5.17) Clefts_{+C}

[Chihiro-ga e_i tabeta no]-wa onigiri-o mi-ttsu_i da.
 [C-NOM ate NO]-TOP rice.ball-ACC 3-CL COP
 ‘It is three rice balls that Chihiro ate.’

(5.18) *No-da* sentences

Chihiro-ga onigiri-o mi-ttsu tabeta no-da.
 C-NOM rice.ball-ACC 3-CL ate NO-COP
 ‘It is that Chihiro ate three rice balls.’

Based on Rizzi’s (1997) Split-CP Hypothesis, *no-da* sentences are assigned the structure (5.19), where *no* heads FinP and *da-* heads FocP. *Da-* is a focus particle; it is cross-linguistically attested that copulas are grammaticalised into focus particles (e.g., Li & Thompson 1989).

(5.19) Structure of *no-da* sentences

[_{TopP} [_{FocP} [_{FinP} [_{TP} ... NP-case ...] no] da]]

Clefts_{+C} are derived from (5.19) in two steps. First, the focus element NP-case moves from a theta position to Spec FocP by focus movement as in (5.20). Second, the remnant FinP undergoes topic movement to Spec TopP, where it is suffixed by the topic marker *wa*. This is illustrated in (5.21).

(5.20) Focus movement

[_{TopP} [_{FocP} NP-case_i [_{FinP} [_{TP} ... t_i ...] no] da]]

(5.21) Topic movement

$$[\text{TopP} [\text{FinP} [\text{TP} \dots t_i \dots] \text{no}]_j\text{-wa} [\text{FocP} \text{NP-case}_i t_j \text{da}]]$$

In this way, clefts_{+C} are derived from *no-da* sentences by focus movement, followed by topicalisation of the remnant.

This derivation accounts for several properties of clefts_{+C}. First, clefts_{+C} are sensitive to islands due to focus movement. Second, the case-matching problem in Kizu (2005) disappears because a focus originally resides at a theta position. Third, multiple foci result from multiple applications of focus movement. Finally, the analysis also accommodates some ellipsis constructions such as sluicing and stripping.

Despite these positive consequences, Hiraiwa & Ishihara's (2012) analysis suffers from some problems. A technical issue is that in the final output (5.21), the trace t_i is left unbound at this surface representation since it cannot be c-commanded by NP-case_i. This is in violation of the Proper Binding Condition (Fiengo 1977). Another, more substantial problem is that their analysis ends up with a non-uniform analysis of clefts_{+C} and clefts_{-C}. As stated above, the syntactic structural dichotomy of the two types of clefts is not compatible with the corpus data provided in Sect. 5.3, Chap. 2. Further, it cannot treat the full spectrum of multiple foci data. In particular, partially case-marked foci manifest the two types of clefts simultaneously within a single structure, as it were; thus, any non-uniform analysis that stipulates lexical ambiguity of *no*, like Hiraiwa & Ishihara (2012), would be unwarranted.

2.5 Summary

This section has assessed previous MGG studies on Japanese clefts. Overall, they propose a **non-uniform** account in speculating lexical ambiguity of the particle *no* depending on the presence of a case particle attached to a focus item. This is not consistent with the corpus finding in Sect. 5, Chap. 2 that case dropping seems to be controlled by pragmatic (but not structural) factors. In addition, they cannot handle

partially case-marked foci data because they instantiate the two types of clefts within a single structure. The only thorough attempt to model the particle *no* uniformly is Kizu (2005), but her analysis assigns radically discrete structures to the two types of clefts, and it cannot treat multiple foci, let alone partially case-marked foci. This situation may be indicative of the need for a shift in perspective. The remainder of this chapter shows that the shift of perspective from a static to a more dynamic theory of language is fruitful for accounting for a wide range of issues in clefts and related constructions such as stripping and sluicing.

3 A Dynamic Account

3.1 Introduction

This section proposes an account of Japanese clefts which directly reflects the flow of interpretation: the pre-*no* part (i.e., the presupposition clause) sets a context with respect to which the focus is subsequently interpreted within a fresh tree. Sect. 3.2 formalises this analysis by combining the entry of the nominaliser *no* (Chap. 4) and the entry of the topic marker *wa* (Chap. 3). Sect. 3.3 shows that the entry of the unit *no-wa* enables a uniform account of clefts_{+C} and clefts_{-C}, and Sect. 3.4 argues that this account captures a number of properties of clefts, including the corpus data in Chap. 2. Finally, Sect. 3.5 summarises the main results.

3.2 The Proposal

The key string in clefts is the succession of *no-wa*. These items have already been analysed in previous chapters. In this sub-section, I propose that the combination of the entry of *no* and that of *wa* leads to a simplified account of clefts.

Chap. 4 (Sect. 3.2) gave the entry of *no* (4.32), reproduced here as (5.22). It declares that if the current node is of type-*t* and if there is an evaluated proposition φ_E

containing a term α , then it is inversely LINKed to a type-e node which is decorated with α . This is schematically shown in (4.33), repeated here as (5.23).

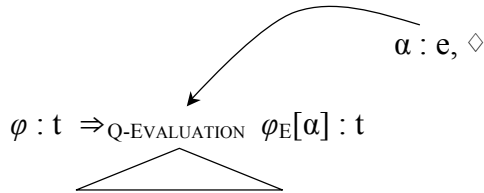
(5.22) Entry of the nominaliser *no*

```

IF      t
THEN    IF       $\varphi_E[\alpha]$ 
        THEN    make/go(<L-1>); put( $\alpha : e$ )
        ELSE    ABORT
ELSE    ABORT

```

(5.23) Output structure of parsing *no*



Chap. 3 (Sect. 5.3) gave the entry of *wa* (3.81), repeated here as (5.24). It declares that if the current node is decorated with a type-e content β , it is inversely LINKed to a type-t-requiring node which is decorated with $?<\downarrow D>(\beta)$, as in (5.25). This requirement ensures that the content β will be shared by the paired trees.

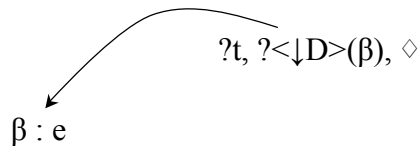
(5.24) Entry of the topic marker *wa* (final version)

```

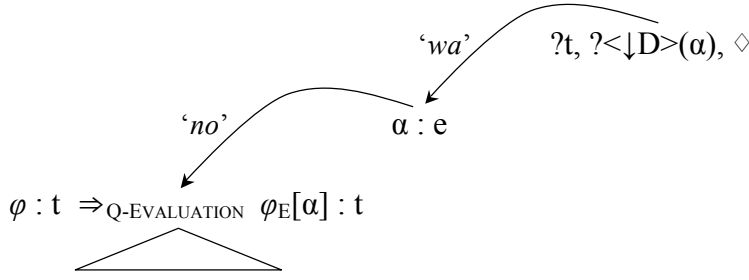
IF       $\beta : e$ 
THEN    make/go(<L-1>); put( $?t, ?<\downarrow D>(\beta)$ )
ELSE    ABORT

```

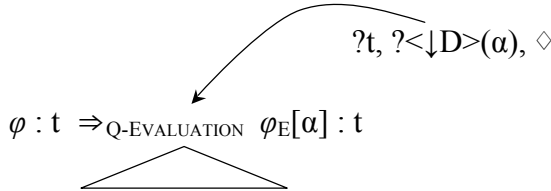
(5.25) Output structure of parsing *wa*



If these two items are parsed, the tree (5.26) is engendered. In this tree display, there are **two** inverse LINK relations, the first of which was induced by the nominaliser *no* and the second of which was induced by the topic marker *wa*.

(5.26) Output structure of parsing *no-wa*

It is possible to model Japanese clefts by inducing two inverse LINK relations. However, a simpler account presents itself if we assume that (i) *no* and *wa* have formed a **unit** through routinisation of minimising costs for introducing two LINK relations and that (ii) the unit *no-wa* **directly** connects two **propositional** nodes. That is, the unit *no-wa* inversely LINKs a type-*t* node to a type-*t*-requiring node, as in (5.27). The entry of *no-wa* is formulated accordingly by combining the entry of *no* (5.22) and the entry of *wa* (5.24), as in (5.28).

(5.27) Output structure of parsing the unit *no-wa* (preliminary version)(5.28) Entry of the unit *no-wa* (preliminary version)

```

IF      t
THEN    IF      φ_E[α]
        THEN    make/go(<L-1>); put(?t, ?<↓D>(α))
        ELSE    ABORT
ELSE    ABORT

```

The entry of *no-wa* may be further simplified. In (5.28), α is an arbitrary type-*e* element. Given that this entry is specifically used for clefts, α may be specified as the content of a gap. In Chap. 3 (Sect. 5.2), the content of a gap is notated as $(\varepsilon, x, P(x))$, where P is an abstract predicate (Kempson & Kurosawa 2009: 65). I propose that *no-wa*

makes use of the variable x in $(\epsilon, x, P(x))$ in writing a requirement on a shared content, as in $?<\downarrow D>(x)$.⁴ As we shall see, this variable x will appear in the content of the focus. This is informally put in (5.29), and formally presented in (5.30).

(5.29) Proposal

The unit *no-wa* encodes the procedures to perform these actions:

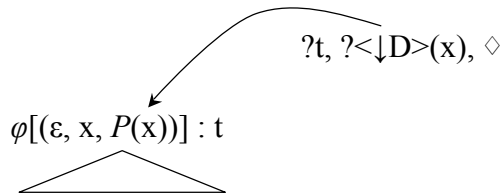
- a. to inversely LINK the tree of the preceding clause to a fresh node,
- b. to decorate the node with the requirement that it will be of type- t , and
- c. to decorate the node with the requirement that the term of the focus item will share a variable with the content of the gap.

(5.30) Entry of the unit *no-wa* (final version)

IF $\phi[(\epsilon, x, P(x))] : t$
 THEN **make/go**($<L^{-1}>$); **put**($?t, ?<\downarrow D>(x)$)
 ELSE ABORT

The entry (5.30) explicitly mentions $(\epsilon, x, P(x))$, which appears in the **pre**-evaluated proposition. (In (5.30), ϕ is a pre-evaluated proposition, and in (5.28), ϕ_E is an evaluated proposition.) So, in processing clefts, the parser may skip QUANTIFIER EVALUATION. That is, once the presupposition clause constructs a propositional tree, LINK transitions may occur **without** evaluating the proposition. This simplifies our account of clefts further more. The schematic tree display is given in (5.31).

(5.31) Output structure of parsing the unit *no-wa* (final version)



⁴ Alternatively, one could write the requirement as $?<\downarrow D>(\epsilon, x, P(x))$. In this case, the notion of ‘requirement satisfaction’ must be modified. Suppose that the focus constructs $(\epsilon, x, frog'(x))$. Then, we need to re-define ‘requirement satisfaction’ in such a way that $?<\downarrow D>(\epsilon, x, P(x))$ is satisfied by the node decorated with $(\epsilon, x, frog'(x))$.

The entry of the unit *no-wa* (5.30), together with its attendant tree display (5.31), will be illustrated with concrete examples in the next sub-section, where I shall put forward a uniform analysis of the two types of clefts.

3.3 The Analysis

The main tenet of the dynamic analysis is that tree growth of clefts_{+C}/clefts_{-C} follows the dynamics of left-to-right processing. To give an informal overview, I shall argue for the following tree update processes:

- The parser constructs a tree for the presupposition part (i.e., string prior to *no-wa*) and *no-wa* relates it to another propositional node.
- This node is developed with the presupposition tree as context, during which a focus is parsed by dint of structural underspecification.⁵
- Clefts_{+C} and clefts_{-C} differ **solely** in the mode of resolving this structural underspecification as driven by the case particles (clefts_{+C}), the other by the general action UNIFICATION (clefts_{-C}).
- Finally, **the same** tree output emerges in both types of clefts, ensuring their truth-conditional equivalence.

3.3.1 Clefts_{+C}

Let us take first the simple case of clefts_{+C} (5.32).⁶

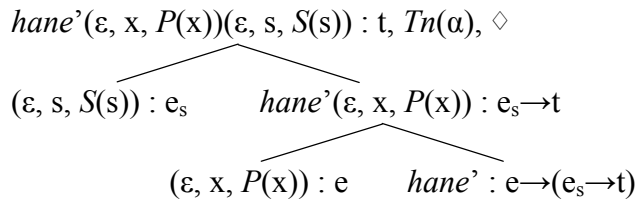
⁵ In Sect. 5, it will be pointed out that the focus item may also be parsed by inducing an inverse LINK relation. This LINK-based processing of the focus is available for both types of clefts, but a well-formed final state is attained only when the focus lacks a case particle. This accounts for the island-sensitivity asymmetry between the two types of clefts.

⁶ The focus, *susuwatari*, denotes wandering-soots, characters in the film *Spirited Away* (2001, Studio Ghibli). These black, fuzzy-haired ‘soot sprites’ work in Kamaji’s boiler room.

(5.32) Clefts_{+C}

[e_i hane-teiru no]-wa susuwatari-ga san-biki_i da.
 [hop-PROG NO]-TOP wandering-soot-NOM 3-CL COP
 ‘It is three wandering-soots that are hopping.’

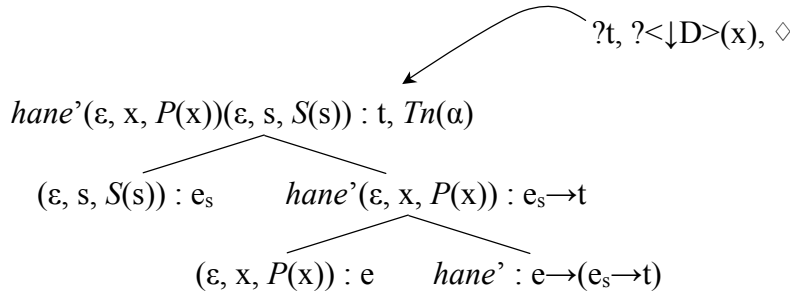
As always, the AXIOM introduces a type-*t*-requiring node, and the verb *hane-* (= ‘hop’) projects a propositional schema with a situation and a subject slot. The meta-variable at the situation node is substituted with $(\epsilon, s, S(s))$, where *S* is a temporary expedient to be replaced with a full-blown predicate modelling tense (see Sect. 4.4, Chap. 4). The meta-variable at the subject node is saturated as $(\epsilon, x, P(x))$, where *P* is an abstract predicate (Kempson & Kurosawa 2009: 65); see the last sub-section. After ELIMINATION (i.e., functional application, type deduction) fires, the aspect suffix *teiru* is scanned. (Tense and aspect are ignored throughout the chapter.)

(5.33) Parsing *Hane-teiru*

The next item is the unit *no-wa*. The entry of *no-wa* (5.30) is repeated here as (5.34). As argued in the last sub-section, this entry allows the tree transition to proceed without evaluating a proposition. Thus, the current type-*t* node is inversely LINKed to a fresh node that is decorated with ?*t* and ?<↓D>(x). When the latter requirement is met, the paired trees will share the variable *x*; this ensures that the variable within the term of the focus is identical to the variable *x* within the term of the gap, $(\epsilon, x, P(x))$.

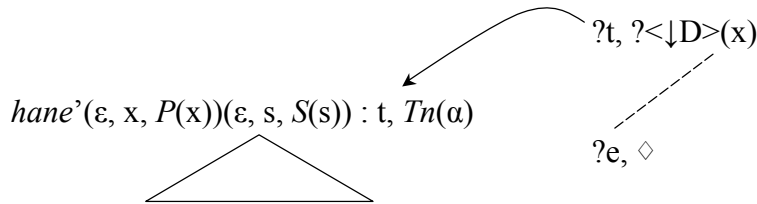
(5.34) Entry of the unit *no-wa* (final version)

IF $\phi[(\epsilon, x, P(x))] : t$
 THEN make/go(<L⁻¹>); put(?*t*, ?<↓D>(x))
 ELSE ABORT

(5.35) Parsing [*Hane-teiru no*]-*wa*

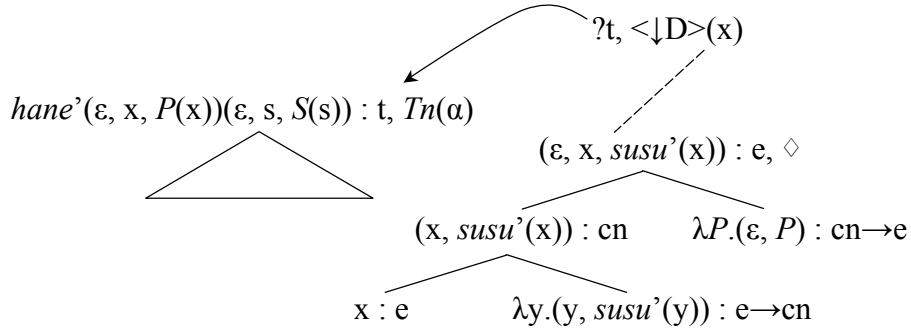
The next item is the focus *susuwatari* (= ‘wandering-soot’). What is required is a creation of a type-e-requiring node. This is achieved by LOCAL *ADJUNCTION, which induces a type-e-requiring unfixed node (Sect. 3.3, Chap. 3). In the tree (5.36) (and the subsequent trees), the structure of the presupposition clause is schematised by a triangle, and irrelevant tree decorations are glossed over.

(5.36) LOCAL *ADJUNCTION

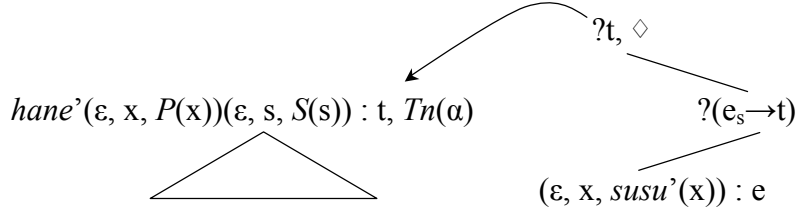


The current node is decorated by the focus *susuwatari*. In (5.37), the structure of the term $(\epsilon, x, \text{susu}'(x))$ is explicitly expressed, where there is a node for the variable x . Thus, the requirement $\text{?<\downarrow D>(x)}$ is **satisfied** at this point.⁷

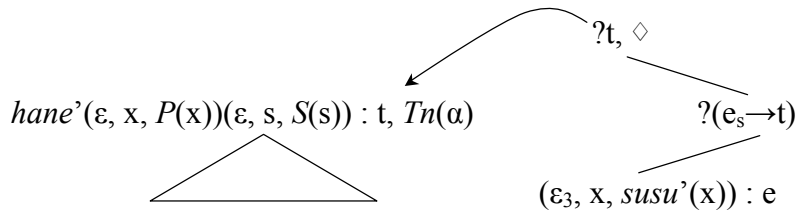
⁷ In Cann et al. (2005) and Kempson et al. (2001), every time a term is created, the variable in the term must be fresh. In the present case, however, the requirement $\text{?<\downarrow D>(x)}$ forces the parser to introduce the non-fresh variable x . Another environment where the non-fresh variable matters is appositive constructions (Ruth Kempson p.c.). Thus, in the string *A girl, a friend of Lin, works in Yubaba's bathhouse*, the content of *a girl* shares a variable with that of *a friend*.

(5.37) Parsing [*Hane-teiru no*]-wa *susuwatari*

Tree updates so far are neutral between clefts_{-C} and clefts_{+C}, but they start to differ since only the latter has a case particle. In the cleft_{+C} (5.32), the nominative-case particle *ga* resolves the unfixed node as a subject node; for the entry of *ga*, see Sect. 5.2, Chap. 3. (In (5.38) and the subsequent tree displays, the structure of the epsilon term is suppressed for brevity.)

(5.38) Parsing [*Hane-teiru no*]-wa *susuwatari-ga*

The next item is the numeral quantifier *san-biki* (= ‘3-CL’). Plurality has not been extensively addressed within DS, and this thesis simply notates the relevant binder as ϵ_3 (Cann et al. 2005: 284).

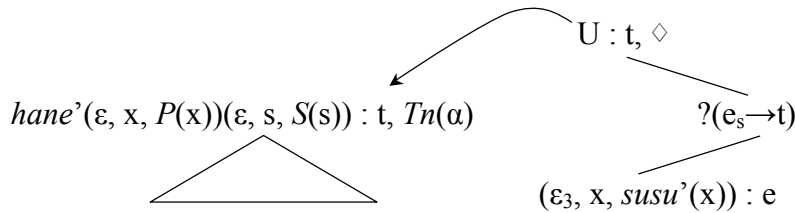
(5.39) Parsing [*Hane-teiru no*]-wa *susuwatari-ga san-biki*

Finally, the copula *da-*. Seraku (2013b) argues that *da-* is a **propositional** pro-form.⁸ More specifically, the copula *da-* decorates a type-*t*-requiring node with a propositional meta-variable *U* and the logical type *t*. A preliminary version of the entry of *da-* is given in (5.40); this entry will be amended in Sect. 5.3.

(5.40) Entry of the copula *da-* (preliminary version)

IF ?*t*
 THEN put(*U* : *t*)
 ELSE ABORT

(5.41) Parsing [*Hane-teiru no*]-*wa susuwatari-ga san-biki da*

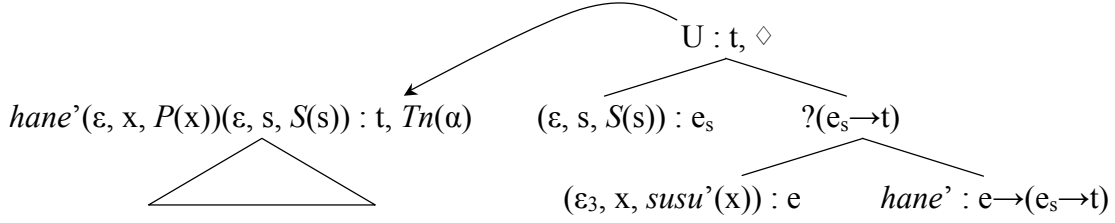


As a propositional meta-variable, *U* at the current node licenses REGENERATION, an instruction to re-run actions stored in context; recall that a DS notion of context consists of (i) words that have been parsed, (ii) structures that have been generated, and (iii) actions that have been run (Sect. 6, Chap. 3).⁹ In the present case, it licenses a repetition of the actions that led to the presupposition tree (i.e., the tree for *hane-teiru*). In particular, *hane-* (= ‘hop’) projects a propositional template where a situation and a subject node are annotated with a meta-variable. A meta-variable at the situation node is saturated as $(\epsilon, s, S(s))$. The subject node created by *hane-* harmlessly collapses with the pre-existing node of the focus since a meta-variable is commensurate with any specified formula such as $(\epsilon_3, x, susu'(x))$ (see Sect. 4.1, Chap. 3).

⁸ The conception of *da-* as a pro-form is also suggested in Okutsu (1974: 8-10) and Pustet (2003: 60-1). Further, the copula *be* in English is also viewed as a pro-form within DS in Cann (2006) and Cann et al. (2005: Chap. 8). Thanks to Ruth Kempson for a helpful exchange.

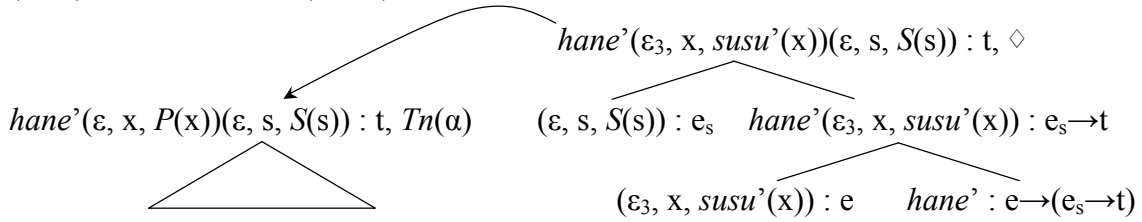
⁹ Another option is to re-use ‘structure’ (rather than ‘actions’). In the present case, the parser may copy the structure of the presupposition clause onto the current node. This tree update also leads to the same representation. This raises interesting implications for the strict/sloppy distinction in ellipsis constructions in Japanese; see footnote 20 of this chapter.

(5.42) REGENERATION



ELIMINATION (i.e., functional application, type deduction) fires twice, the first application computing the formula and its type at the intermediate functor node, and the second application computing the formula and its type at the top node. In the final state (5.43), the proposition at the top node represents the interpretation of the cleft_{+C} (5.32): ‘It is three wandering-soots that are hopping.’ (Q-EVALUATION of the proposition in the final state is ignored throughout this chapter.)¹⁰ This interpretation is truth-conditionally equivalent to the interpretation ‘Three wandering-soots are hopping.’

(5.43) ELIMINATION (twice)



The analysis is consonant with the stream of cleft understanding: the pre-*no-wa* part (i.e., the presupposition clause) builds up a propositional tree, relative to which the focus is interpreted within a fresh tree. The node of the focus is initially underspecified in the fresh tree and it is subsequently fixed by a case particle. By contrast, in clefts_{-C}, where a case particle is absent from the focus, the unfixed node of the focus is settled by a different means, as we shall see in the next sub-section.

¹⁰ The content of the focus is situated within the main structure in the final state. This predicts ‘reconstruction’ in clefts (Chomsky 1977, Heycock & Kroch 1999); see also Hoji (2003) for issues surrounding reconstruction in Japanese. As observed in Kizu (2005) and others, however, some items (e.g., the NPI *daremo* (= ‘anyone’)) are resistant to reconstruction in clefts; see also Percus (1997) for English data. I shall set aside the topic of reconstruction in this thesis.

3.3.2 Clefts_{-C}

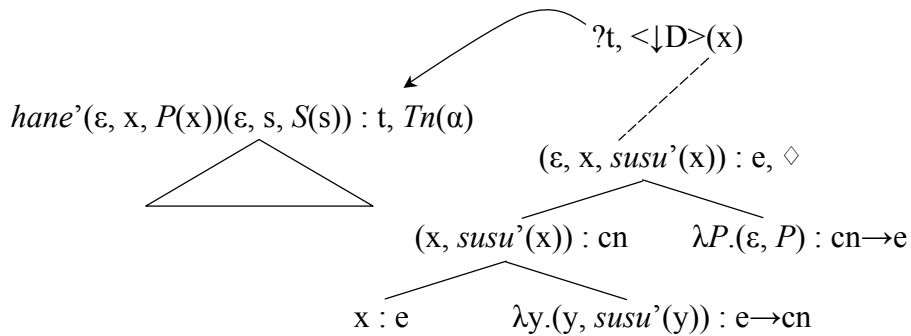
The dynamics of incremental build-up of interpretation for clefts_{+C} also applies to clefts_{-C} such as (5.44). This time, however, with no case particle, the unfixed node for the focus will be resolved by the general action UNIFICATION.

(5.44) Clefts_{-C}

[e_i hane-teiru no]-wa susuwatari san-biki_i da.
 [hop-PROG NO]-TOP wandering-soot 3-CL COP
 ‘It is three wandering-soots that are hopping.’

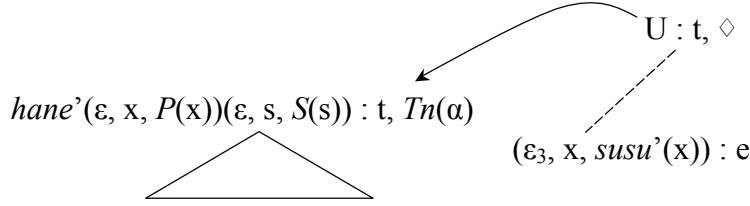
In line with the tree updates in the last sub-section, the parse of (5.44) up to the focus item *susuwatari* (= ‘wandering-soot’) generates (5.45). This tree is identical to (5.37).

(5.45) Parsing [*Hane-teiru no*]-wa *susuwatari*



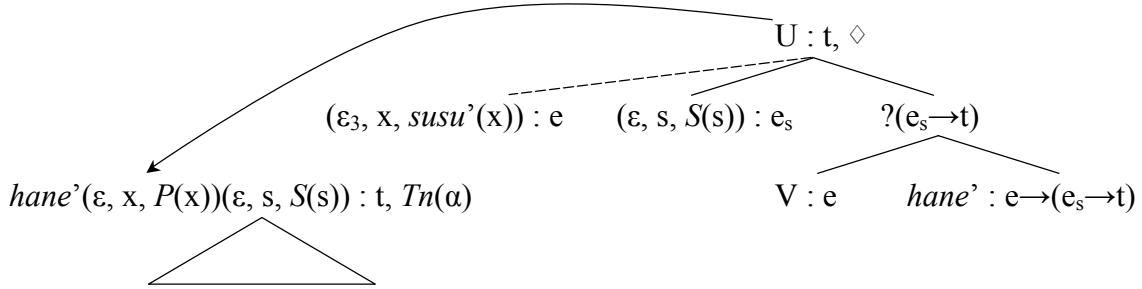
As in the previous example, the requirement $?<\downarrow D>(x)$ is met by the variable x within the internal structure of the term of the focus *susuwatari*.

The cleft_{-C} (5.44) lacks a case particle, and the tree update proceeds **without** resolving the unfixed node of the focus *susuwatari* at this stage. The numeral quantifier *san-biki* (= ‘3-CL’) updates the binder ε as ε_3 (Cann et al. 2005: 284), and the copula *da* annotates the type- t -requiring node with a propositional meta-variable U . (From the tree (5.46) on, the internal structure of the epsilon term is again omitted.)

(5.46) Parsing [*Hane-teiru no*]-wa *susuwatari san-biki da*

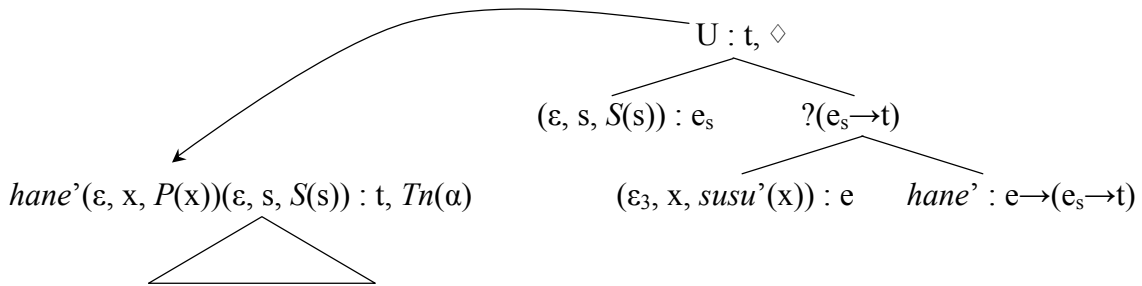
The propositional meta-variable U licenses REGENERATION, re-running actions to build up the presupposition tree, with *hane-* (= ‘hop’) projecting a situation and a subject node. In (5.47), the situation meta-variable is substituted with $(\epsilon, s, S(s))$, whereas the subject node is still decorated with a meta-variable V .

(5.47) REGENERATION



There is now a fixed node decorated with a meta-variable V of type- e , and it may be **merged** with the unfixed node decorated with $(\epsilon_3, x, susu'(x))$. This merger process is formalised as the general action UNIFICATION (Sect. 3.3, Chap. 3).

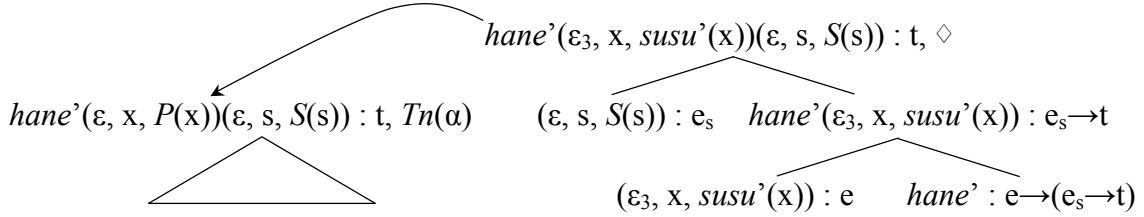
(5.48) UNIFICATION



Finally, ELIMINATION fires twice, the first application yielding a decoration at the intermediate functor node, the second application yielding a decoration at the top

node. In (5.49), the propositional formula at the top node represents the interpretation of the cleft_{-C} (5.44): ‘It is three wandering-soots that are hopping.’ This tree is **identical** to (5.43), the one for the cleft_{+C} (5.32). This tree identity indicates that these two cleft strings are truth-conditionally equivalent.

(5.49) ELIMINATION (twice)



3.3.3 A Realistic Model of Clefts

We have put forth a dynamic account of Japanese clefts. Tree updates for clefts_{+C} and clefts_{-C} proceed identically up to the parse of a focus, but they differ in the mode of resolving an unfixed node of the focus: the lexical actions of a case particle (clefts_{+C}) vs. the general action UNIFICATION (clefts_{-C}). Finally, the tree outputs for both types of clefts converge and their truth-conditional equivalence holds.

Unlike previous accounts, tree updates are surface-oriented in that they are sensitive to the unfolding presentation of the words in a surface string, even though what is constructed is a semantic tree, not a structure inhabited by words. In this surface sequence of words, neither empty categories nor lexical ambiguity of *no* are stipulated. This tree update fits well with the corpus data showing that particle drop is controlled pragmatically (Sect. 5.3, Chap. 2), which suggest that the two types of clefts are not distinguished structurally. In our analysis, the two types of clefts differ solely in the mode of resolving an unfixed node of the focus: if the surface sequence contains a case particle, the unfixed node is resolved by the instruction encoded in the case particle; otherwise, it is resolved by the general action UNIFICATION. Prior to the case particle, the update for the two types of clefts may proceed identically, and after the resolution of the unfixed node of the focus, the same output obtains.

The proposed account is also in harmony with the recent psycholinguistic data on Japanese clefts. Kahraman (2011) and Kahraman et al. (2011a) report that by the time the parser processes *no-wa*, it has started to form a dependency between a gap and a focus item that has not yet come. In our analysis, the unit *no-wa* creates a fresh type-*t*-requiring node decorated with the requirement $?<\downarrow D>(x)$. It was pointed out in Chap. 3 (Sect. 7) that the predictivity of DS tree update lies in a range of **requirements**. In our case, it is due to the requirement $?<\downarrow D>(x)$ that the parser may anticipate a dependency between a gap and an upcoming focus. $?<\downarrow D>(x)$ requires that the variable in the term of a gap will appear in the term of the focus that will be subsequently processed. Thus, the parser predicts a dependency between a gap and an upcoming focus by the time the unit *no-wa* is processed.

3.4 Consequences

A number of consequences are drawn from the dynamic analysis of clefts. This subsection discusses the corpus findings in Chap. 2 and the notion of topic and focus.

3.4.1 Category of Foci

Chap. 2 (Sect. 5.4) reported that the category of foci was diverse, as shown in Table 2.9, reproduced below.

Category	Occurrences (clefts _{+C})	Occurrences (clefts _{-C})
NP	8	261
Adv-NP	0	23
Clausal-Adv-NP	16	1
Dem	0	1
WhP	1	11
Total	25	297

Table 2.9. Category of a focus in clefts in the CSJ

Though the table displays a range of categories, note that all of these categories have a **type-e** status in DS (Marten 2002). According to the entry of *no-wa* (5.30), reproduced

here as (5.50), the IF-block specifies the input condition: the current node should be decorated with a proposition which contains the **type-e** term $(\varepsilon, x, P(x))$, or the content of a gap. Then, $?<\downarrow D>(x)$ requires that the variable x in the term will appear in the **type-e** content of the focus item. With the reasonable assumption that a nominal category corresponds to the logical type e , our analysis predicts that the focus may bear any of the categories in the table.

(5.50) Entry of the unit *no-wa* (final version)

IF $\varphi[(\varepsilon, x, P(x))] : t$
 THEN $\text{make/go}(<L^{-1}>); \text{put}(?t, ?<\downarrow D>(x))$
 ELSE ABORT

Some remark is in order for ‘adjunct’ categories like Adv-NP and Clausal-Adv-NP. For example, consider (2.59), repeated here as (5.51). The focus *mikkakan* (= ‘for three days’) is of the category Adv-NP.

(5.51) Clefts, extracted from the CSJ (S01F0217)

[Nishiomotejima-ni i-ta no]-wa jissai mikkakan nandesu...
 [N-in stay-PAST NO]-TOP actually for.three.days COP
 ‘It is actually for three days that I was in Nishiomotejima...’

A full account of adjuncts cannot be developed in this thesis, but some preliminary ideas may be presented. À la Marten (2002), who claims that adjuncts are treated as **type-e** items, I assume that Japanese verbs project a propositional template with all argument slots and optionally introduce type-e nodes for adjuncts (see also Sect. 3.4.6, Chap. 4). For instance, in (5.51), *i-* (= ‘stay’) projects a propositional schema with a situation, a subject, and also a temporal adjunct node. On the assumption that the adjunct node is a node for a gap, it is decorated with $(\varepsilon, x, P(x))$, where P is an abstract predicate (see Sect 3.3). Then, *no-wa* ensures that the term of the focus *mikkakan* (= ‘for three days’) will contain the variable x that appears in $(\varepsilon, x, P(x))$. This requirement is met when the term $(\varepsilon, x, \text{mikkakan}'(x))$ is constructed.

In some languages, a predicate may occupy a focus position in clefts (so-called ‘predicate clefts’). For instance, in Haitian clefts (5.52), the initial occurrence of the verb *kouri* (= ‘run’) is at the focus position, marked by the particle *se*.

- (5.52) Predicate clefts [Haitian]
 Se kouri Bouki ap kouri.
 SE run B PROG run
 ‘Bouki is running.’ (adapted from Harbour (2008: 853))

Thus, there is nothing conceptually wrong with focussing a predicate in clefts, but this type of structure in Japanese is barred by the type-e restriction encoded in *no-wa*.

3.4.2 Grammatical Function of Foci

Chap. 2 (Sect. 5.5) showed that foci might bear a wide variety of grammatical functions.

This was illustrated in Table 2.11 in Chap. 2, repeated below:

Type	Occurrences (clefts _{+C})	Occurrences (clefts _{-C})
subject	3	138
object	0	74
indirect object	0	3
function _{PLACE}	0	14
function _{TIME}	9	33
function _{REASON}	11	7
function _{THEME}	2	0
unclear	0	28
Total	25	297

Table 2.11. Grammatical function of a focus in clefts in the CSJ

These data are expected in our analysis. In the entry of *no-wa* (5.50), grammatical-functional constraints are **not** imposed on the focus expression. The sole condition is that the content of the focus should involve the variable *x* that is present in the term of the gap (ϵ , *x*, *P(x)*). Thus, the focus may have a variety of grammatical functions as shown in the table. As for non-core functions (e.g., function_{TIME}), the node of an adjunct gap is optionally constructed by the verb and decorated with (ϵ , *x*, *P(x)*) (see the last

sub-section). Finally, there are 28 instances of the ‘unclear’ category. These are the cases where it is not obvious whether the focus is an argument or an adjunct of the embedded verb. In (2.68), repeated here as (5.53), it is not clear whether the focus, the complex NP headed by *toko* (= ‘place’), is the argument or the adjunct of *iku-* (= ‘go’). I shall set aside the issue of valency in these examples; but no matter what grammatical function the focus in these cases bears, it does not contradict with the entry of *no-wa*, which is silent to the notion of grammatical function.

(5.53) Clefts, extracted from the CSJ (S00F0193)

[watashi-ga yoku iku no]-wa [Harajuku-ni aru Togo-jinja
 [I-NOM often go NO]-TOP [H-in exist T-shrine
 toiu toko] nandesu...
 TOIU place] COP
 ‘It is the Togo shrine, located in Harajuku, that I often go to...’

3.4.3 Information Status of Foci

The focus item usually conveys new information, but Chap. 2 (Sect. 5.6) pointed out that there were at least 7 examples where the focus conveyed old information. One such example is (2.64), repeated here as (5.54).

(5.54) Clefts, extracted from the CSJ (S03M0996)

[warui-hou-no yuumeina no]-wa sore desu-ga...
 [bad-side-GEN famous NO]-TOP that COP-but
 ‘It is that thing (= the river) that is notorious, but...’

These data are also consonant with the entry of *no-wa*. This is because the entry does **not** specify any condition on the information status of the focus. Thus, as long as the term of the focus contains a variable that is identical to the variable in the term of the

gap (as required by forced sharing of a type-e element in a LINK relation), the focus item may convey new or old information depending on context.¹¹

Various corpus data have been analysed so far; what has not been discussed is ‘long-distance dependencies and islands’ (Sect. 5.7, Chap. 2). This issue will be taken up in Sect. 5. In the next sub-section, I shall draw another implication of our analysis by contemplating the notions of topic and focus.

3.4.4 Topic and Focus

The notions of topic and focus are difficult to define as they pertain to various types of constructions (e.g., clefts, topicalisation, *wh*-questions) and these constructions behave distinctly across languages (Erteschik-Shir 2007: 26). ‘Topic’ is generally conceived of as old or given information that stands in an aboutness relation to the information expressed by the non-topicalised part of the string. ‘Focus’ is a more elusive concept: in the case of assertion, it is part of the propositional content of a string that assigns a value to an issue under discussion (Lambrecht 1996).

As stated in Chap. 3 (Sect. 5.3), DS does not postulate a primitive concept of topic; this is also true of the concept of focus. Rather, topic and focus effects are captured as a **by-product** of gradual growth of semantic representation (Cann et al. 2005: 183-4, Kempson et al. 2006). More specifically, topic effects arise when a previous structure provides a context against which an emerging structure is updated. Focus effects arise when the parser has constructed an open proposition and the provision of a term leads to a closed proposition.

The same form of explanation is applicable to Japanese cleft constructions. First, the presupposed part (i.e., the string prior to *no-wa*) builds up a propositional tree, and *no-wa* (i) induces an inverse LINK relation to an unfolding propositional node and

¹¹ The example (5.54) would be problematic for approaches to information structure which define the notion of focus in terms of ‘new information’ such as Falk (2001a: 59). Yet, cleft data with a focus conveying old information such as (5.54) do not favour my account unless the present thesis succeeds in accounting for the data covered by such approaches. The discussion of information structure within DS must await other occasions.

(ii) imposes $?<\downarrow D>(x)$, a requirement that the parser will develop the node with respect to the variable x in the presupposition tree. This requirement drives the tree update to build the current propositional tree with the presupposition tree as context, evoking what one regards as a topic effect in clefts. Second, the parser processes a pre-copula item in this emergent propositional tree. The copula *da-* then licenses the reiteration of actions to build the presupposition tree. In particular, the lexical actions encoded in a verb in the presupposition clause project an open proposition, which is saturated by the content of the pre-copula item (i.e., an issue under discussion). This captures the ‘specificational’ nature of clefts (Declerck 1988), provoking a focus effect. In this way, the notions of topic and focus in Japanese clefts are not expressed over the structure; rather, their effects are explained through incremental growth of semantic representation without any necessity to invoke either as a syntactic or semantic primitive.¹²

3.5 Summary

In Japanese clefts, the pre-*no* part (i.e., the presupposition clause) builds up a tree which serves as a context against which the focus expression is processed within a new tree. This account unifies the two types of clefts in that (i) there are no lexical ambiguities in the surface string of words, (ii) the presence of a case particle just affects the way the unfixed node of the focus is settled, and (iii) the same semantic representation finally obtains. This is contrasted with previous work that assigns distinct syntactic structures to the two types of clefts and often hypothesises lexical ambiguity of the particle *no* (Hiraiwa & Ishihara 2012, Hoji 1990, Kizu 2005; see Sect. 2). Our account is preferable to these previous works in virtue of the corpus finding in Chap. 2 that the presence of a case particle is controlled pragmatically (rather than structurally). Additional merits of

¹² Wei & Kempson (2011) put forward a DS analysis of Chinese clefts, arguing that topic and focus effects in Chinese clefts are also modelled in terms of semantic-tree growth. A cross-linguistic account of clefts within DS, however, must await other occasions.

our analysis include the treatment of other corpus data and the modelling of topic and focus effects as an epiphenomenon of progressive growth of an interpretation.

What has not been debated is how this uniform analysis deals with structural differences between the two types of clefts. These are addressed in the subsequent two sections: Sect. 4 on multiple foci and Sect. 5 on island-sensitivity.

4 Multiple Foci

4.1 Introduction

The first difference between clefts_{+C} and clefts_{-C} is concerned with multiple foci. That is, multiple foci are licensed only in clefts_{+C} (e.g., Cho et al. 2008, Hiraiwa & Ishihara 2012, Koizumi 1995, Kuroda 2005).

(5.55) Clefts_{+C}

[Haku-ga e_i e_j age-ta no]-wa Chihiro-ni_j onigiri-o_i da.
 [H-NOM give-PAST NO]-TOP C-DAT rice.ball-ACC COP
 Lit. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

(5.56) Clefts_{-C}

*[Haku-ga e_i e_j age-ta no]-wa Chihiro_j onigiri_i da.
 [H-NOM give-PAST NO]-TOP C rice.ball COP
 Int. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

What has been unnoticed in the literature is the mixed case: when there is more than one focus, the non-final focus must be case-marked but the **final** focus may be case-less. Consider the contrast between (5.57) with (5.58). (See Sect. 4.4 for clefts with more than two foci.)

(5.57) Clefts with partially case-marked foci

[Haku-ga e_i e_j age-ta no]-wa Chihiro-ni_j onigiri_i da.
 [H-NOM give-PAST NO]-TOP C-DAT rice.ball COP
 Lit. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

(5.58) Clefts with partially case-marked foci

*[Haku-ga e_i e_j age-ta no]-wa Chihiro_j onigiri-o_i da.
 [H-NOM give-PAST NO]-TOP C rice.ball-ACC COP
 Int. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

This mixed case is briefly noted in Hiraiwa & Ishihara (2012: 146), but no previous studies have presented an analysis. In fact, Sect. 2 has pointed out that these examples are recalcitrant for extant analyses. The aim of the present section is to explain an array of multiple foci data (plus other sets of new data) as a reflection of language dynamics.

Sect. 4.2 presents the Unique-unfixed-node Constraint, a corollary of the tree logic LOFT. With this constraint, Sect 4.3 explains the multiple-foci contrast between the two types of clefts, and Sect 4.4 makes additional predictions. Finally, Sect. 4.5 summarises the results.

4.2 A Formal Basis for Explanation

According to LOFT (Blackburn & Meyer-Viol 1994), a logical underpinning of DS (Sect. 2.1, Chap. 3), each node must be uniquely identified relative to the other nodes in a tree. This is definitional of a tree. It follows that if multiple nodes in a tree are unfixed with respect to **the same node** and by **the same LOFT relation**, they will be mutually indistinguishable and cannot be separately defined. Thus, if two supposedly distinct unfixed nodes have an identical tree-node label (i.e., unfixed relative to the same node and by the same LOFT relation), they will lead to inconsistency in the node description.

This formal restriction is stated as the Unique-unfixed-node Constraint. Note that this condition is **not** a mere stipulation but a general consequence of LOFT.¹³

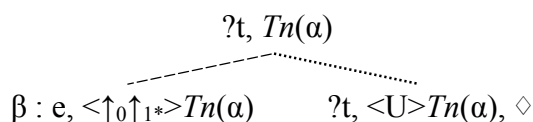
(5.59) The Unique-unfixed-node Constraint

If supposedly distinct unfixed nodes have the same tree-node label but non-commensurate labels, the node description becomes inconsistent and the tree update crashes.

Three clarifications are in order. First, two attempts to build a node with distinct labelling are legitimate only if the labelling is **commensurate**; in such a case, there will only be one such node. Indeed we have already encountered such cases: the parse of an argument NP and a case particle produces a fixed argument node, and this node harmlessly collapses with an argument node created by the verb because the node built by the verb is decorated with a meta-variable, a formula which is commensurate with any specified formula (see the discussion about (5.42)).

Second, if unfixed nodes are decorated with **distinct** locality specifications, they are distinguishable and a tree update may lead to a well-formed state. For instance, the parser may first perform LOCAL *ADJUNCTION and then GENERALISED ADJUNCTION. In this case, the two unfixed nodes may be separately identified since they are decorated with the distinct tree-node labels: $\langle \uparrow_0 \uparrow_1^* \rangle Tn(\alpha)$ vs. $\langle U \rangle Tn(\alpha)$.

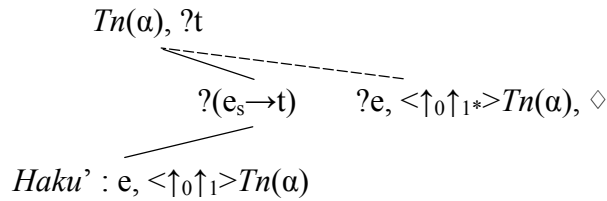
(5.60) LOCAL *ADJUNCTION + GENERALISED ADJUNCTION



¹³ Chap. 3 (Sect. 3.3) states that the DS mechanism of unfixed relations is formally similar to functional uncertainty of LFG, but it seems there is no LFG analogue of the Unique-unfixed-node Constraint. In analysing focus constructions in Russian, King (1997) claims that the attribute FOCUS may have multiple values if these values form a set. I am grateful to Mary Dalrymple for bringing my attention to King (1997).

Third, if an unfixed node that has been introduced is **resolved**, the parser may introduce another unfixed node with the same tree-node label without leading to inconsistency in the description. Consider the string *Haku-ga Chihiro-ni*, where *ga* is a nominative-case particle and *ni* is a dative-case particle. The parser may first run LOCAL *ADJUNCTION to induce a type-e-requiring unfixed node, to be decorated by *Haku*. At this stage, the nominative-case particle *ga* may fix this unfixed node as a subject node. As an unfixed node is no longer in place, the parser may run LOCAL *ADJUNCTION once again to induce a type-e-requiring unfixed node for *Chihiro*.

(5.61) Parsing *Haku-ga* + LOCAL *ADJUNCTION



That is, the successive applications of LOCAL *ADJUNCTION are legitimate only if each unfixed node is resolved before another unfixed node is induced.

I shall now show that the Unique-unfixed-node Constraint (5.59), together with the structural resolution process, explains the asymmetry between the two types of clefts with regard to multiple foci.

4.3 The Analysis

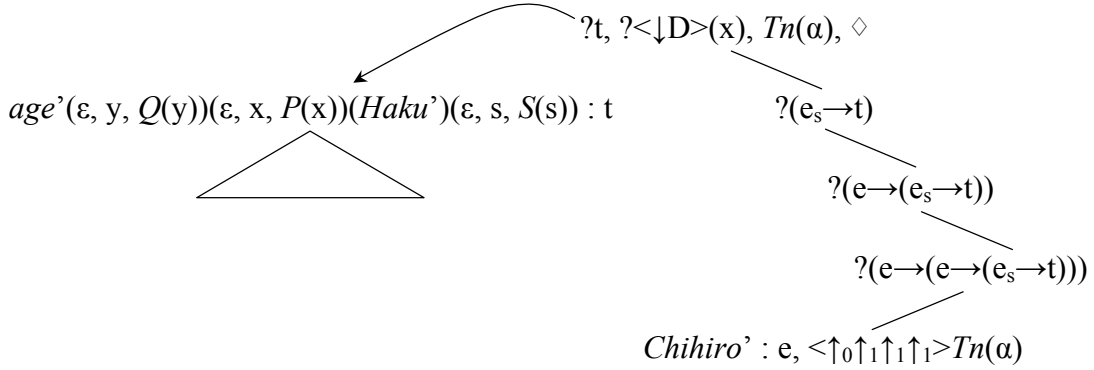
To begin with, multiple foci are licensed in clefts_{+C}. A pertinent example (5.55) is repeated here as (5.62).

(5.62) Clefts_{+C}

[Haku-ga	e_i	e_j	age-ta	no]-wa	Chihiro-ni _j	onigiri-o _i	da.
[H-NOM			give-PAST	NO]-TOP	C-DAT	rice.ball-ACC	COP
Lit. 'It is rice balls _i to Chihiro _j that Haku gave e_i e_j .'							

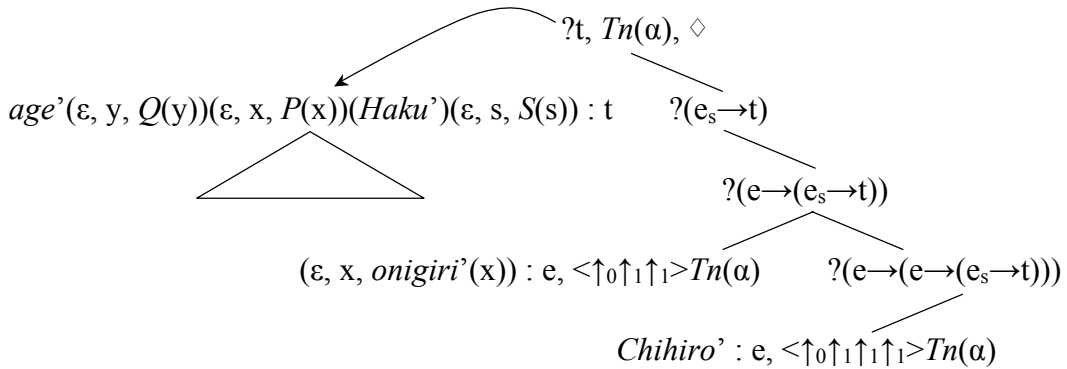
In (5.62), after the parse of *no-wa*, LOCAL *ADJUNCTION induces an unfixed node for the first focus *Chihiro*. This unfixed node is resolved immediately by the dative-case particle *ni*. In (5.63), $(\epsilon, y, Q(y))$ is the content of an indirect object gap, and $(\epsilon, x, P(x))$ is the content of an object gap.

(5.63) Parsing *Haku-ga age-ta no-wa Chihiro-ni*



The tree no longer has unfixed nodes, and the parser may run LOCAL *ADJUNCTION once again to induce an unfixed node for the second focus *onigiri* (= ‘rice ball’). This unfixed node is resolved as an indirect-object node by the dative-case particle *ni*. Note that the requirement $?<↓D>(x)$ has been satisfied by $(\epsilon, x, onigiri'(x))$.¹⁴

(5.64) Parsing *Haku-ga age-ta no-wa Chihiro-ni onigiri-o*



¹⁴ The reader might wonder whether *no-wa* could have posited $?<↓D>(y)$, instead of $?<↓D>(x)$. The answer is positive; in this case, however, proper names must be expressed in the epsilon calculus. For example, the content of *Chihiro* may be represented as the term $(\iota, y, Chihiro'(y))$, with the iota operator ι (Sect. 2.2, Chap. 3).

The copula *da-* then provides a type-*t* meta-variable, licensing the reiteration of actions stored in context (i.e., REGENERATION). In particular, the lexical actions encoded in *age-* (= ‘give’) develop the propositional structure, where the content of *Chihiro* and that of *onigiri* have been incorporated. Therefore, multiple foci are possible when each focus is marked by an appropriate case particle (i.e., clefts_{+C}).

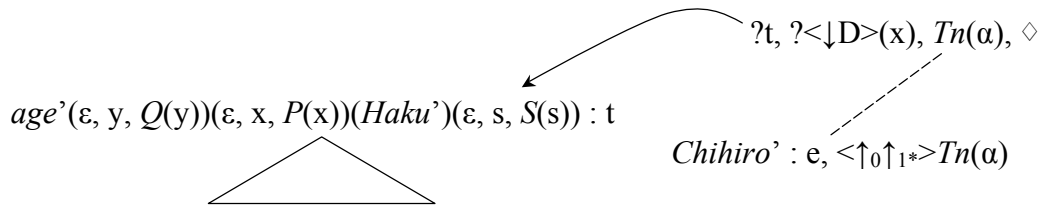
By contrast, multiple foci are not possible if they are not marked by a case particle. Consider the cleft_{-C} (5.56), reproduced here as (5.65).

(5.65) Clefts_{-C}

*[Haku-ga e_i e_j age-ta no]-wa Chihiro_j onigiri_i da.
 [H-NOM give-PAST NO]-TOP C rice.ball COP
 Int. ‘It is rice balls_i to Chihiro_j that Haku gave e_i e_j.’

As usual, LOCAL *ADJUNCTION introduces a type-*e*-requiring unfixed node for the first focus *Chihiro*, but this unfixed node cannot be resolved for the following reasons. First, case particles may resolve an unfixed node but *Chihiro* is case-less. Second, another means of fixing an unfixed relation is UNIFICATION. For UNIFICATION to fire, however, a propositional structure with fixed nodes must have been built up. Crucially, it is after the parse of the copula *da-* that such a structure arises. (Recall that *da-* provides a meta-variable, which licenses the re-run of actions to build a propositional structure.) Thus, the unfixed node of *Chihiro* remains unsettled.

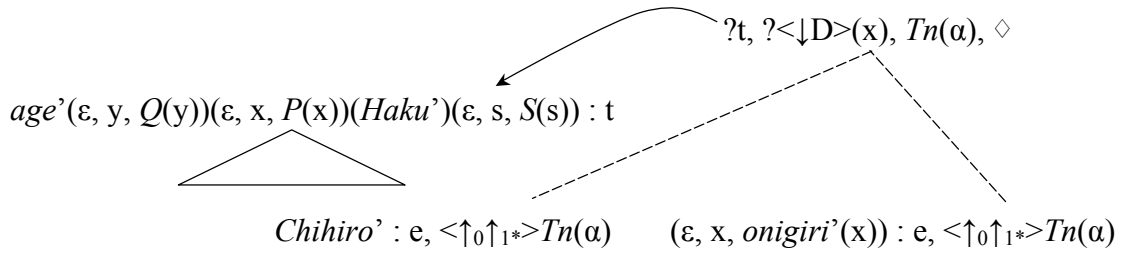
(5.66) Parsing *Haku-ga age-ta no-wa Chihiro*



The next item to be scanned is the second focus *onigiri* (= ‘rice ball’). If LOCAL *ADJUNCTION induces another unfixed node for *onigiri*, it is indistinguishable from the

node of the first focus *Chihiro*, and the node description gets inconsistent. In (5.67), there are two unfixed nodes with the same tree-node label $\langle \uparrow_0 \uparrow_1^* \rangle Tn(\alpha)$, which violates the Unique-unfixed-node Constraint. Hence, multiple foci are not possible when they are case-less (i.e., clefts_{-C}).

(5.67) Parsing *Haku-ga age-ta no-wa Chihiro onigiri*



To sum up, the asymmetry of clefts_{+C}/clefts_{-C} concerning multiple foci is an outcome of gradual tree growth: the asymmetry reflects whether an unfixed node for a focus is resolvable before the parse of the next focus. As a conjecture, let us finally point out that the dynamic analysis predicts not only the ungrammaticality of (5.65) but also the **timing** at which (5.65) gets ungrammatical. Tree growth proceeds successfully prior to the second focus *onigiri* (= ‘rice ball’), but it crashes once *onigiri* is scanned because the node description gets inconsistent at this stage: one and the same node is decorated with both the content of *Chihiro* and that of *onigiri*. Thus, it is expected that (5.65) becomes unacceptable at the timing where the second *onigiri* is parsed. Though this prediction conforms to my intuition, it needs to be tested experimentally. Whether it is verified, there are two essential points: (i) this type of predictions **cannot** be attained in any theories that do not pay due attention to language dynamics; and (ii) this renders the dynamic account **more** falsifiable than the static account.

4.4 Further Predictions

The analysis of multiple foci in clefts makes further predictions. First, the order of foci should not matter just as long as each focus has an appropriate case particle: whatever

the order within the focus cluster, the locally unfixed relation can be fixed before the next application of LOCAL *ADJUNCTION. So, the cleft_{+C} (5.68), where the order of foci is reversed from the one in (5.62), is equally grammatical.

(5.68) Clefts_{+C}

[Haku-ga e_i e_j age-ta no]-wa onigiri-o_i Chihiro-ni_j da.
 [H-NOM give-PAST NO]-TOP rice.ball-ACC C-DAT COP
 Lit. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

By contrast, clefts_{-C} are completely ungrammatical no matter what order the foci are presented, since an unfixed node for the first focus cannot be fixed before the parse of the second focus. Thus, the cleft_{-C} (5.69), where the order of case-less foci is swapped from the one in (5.65), is equally ungrammatical.

(5.69) Clefts_{-C}

*[Haku-ga e_i e_j age-ta no]-wa onigiri_i Chihiro_j da.
 [H-NOM give-PAST NO]-TOP rice.ball C COP
 Int. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

Second, the analysis explains partially case-marked foci, problematic data for previous studies. The example (5.57) is reproduced here as (5.70).

(5.70) Clefts with partially case-marked foci

[Haku-ga e_i e_j age-ta no]-wa Chihiro-ni_j onigiri_i da.
 [H-NOM give-PAST NO]-TOP C-DAT rice.ball COP
 Lit. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

The unfixed node of the first focus *Chihiro* may be resolved by the case particle *ni*, and another unfixed node may be introduced for the second focus *onigiri* (= 'rice ball'). Although *onigiri* is case-less, it is immediately followed by the copula *da-*, which eventually provides the appropriate argument node, with which the unfixed node for *onigiri* can be merged (i.e., UNIFICATION). Thus, *onigiri* may lack a case particle. As a

further confirmation, the acceptable example (5.70) is sharply contrasted with the wholly unacceptable example (5.58), repeated here as (5.71).

(5.71) Clefts with partially case-marked foci

*[Haku-ga e_i e_j age-ta no]-wa Chihiro_j onigiri-o_i da.
 [H-NOM give-PAST NO]-TOP C rice.ball-ACC COP
 Int. 'It is rice balls_i to Chihiro_j that Haku gave e_i e_j.'

In (5.71), the unfixed node of the first focus *Chihiro* cannot be resolved because (i) *Chihiro* is case-less and (ii) UNIFICATION cannot fire since the copula *da-* is not parsed until the end of the string. Therefore, if another unfixed node is induced for the second focus *onigiri*, the two supposedly distinct unfixed nodes are inseparable and result in inconsistency in the node description.

Third, as long as each focus comes with an appropriate case particle, clefts with more than two foci are possible, as in (5.72). For each focus, an unfixed node is resolved immediately by the case particle attached to the focus.

(5.72) Clefts_{+C} with three foci

[e_i e_j e_k age-ta no]-wa Haku-ga_i Chihiro-ni_k onigiri-o_j da.
 [give-PAST NO]-TOP H-NOM C-DAT rice.ball-ACC COP
 Lit. 'It is Haku_i rice balls_j to Chihiro_k that e_i gave e_j e_k.'

Finally, a cleft should be acceptable where there are three foci with only the final focus being case-less. This is because (i) UNIFICATION requires a propositional structure to have been built up; (ii) the construction of such a structure is licensed by the copula *da-*; and (iii) *da-* is processed after the final focus. In short, UNIFICATION is applicable only to the unfixed node of the **final** focus.

(5.73) Clefts with three foci

[e_i e_j e_k age-ta no]-wa Haku-ga_i Chihiro-ni_k onigiri_j da.
 [give-PAST NO]-TOP H-NOM C-DAT rice.ball COP
 Lit. ‘It is Haku_i rice balls_j to Chihiro_k that e_i gave e_j e_k.’

This string is contrasted with the other six possible variants, all of which are totally unacceptable: the examples where (i) only *Haku* is case-less, (ii) only *Chihiro* is case-less, (iii) only *Haku* and *Chihiro* are case-less, (iv) only *Haku* and *onigiri* are case-less, (v) only *Chihiro* and *onigiri* are case-less, and (vi) all foci are case-less. These are all fully ungrammatical since the tree transitions necessarily multiply unfixed nodes with the same node-label.

4.5 Summary

The dynamic account of clefts treats a variety of issues in multiple foci without losing uniformity. The heart of the analysis is the incremental updating of a semantic tree, especially ‘structural underspecification and resolution,’ together with the Unique-unfixed-node Constraint, a corollary of the tree logic LOFT (Blackburn & Meyer-Viol 1994). This result constitutes a clear advantage of our analysis because multiple foci in clefts (especially, partial case-marking) pose a challenge to previous studies (Sect. 2).

In our analysis, we maintain the structural uniformity of the two types of clefts: there is no ambiguity in the surface sequence, and the identical final output is derived. The presence/absence of a case particle at a focus position simply affects the way structural underspecification is resolved. This conforms to the corpus finding in Chap. 2 (Sect. 5.3) that the case particle dropping is affected pragmatically. This unified account has explained away the first set of evidence for a bifurcated view of Japanese clefts: multiple foci. The next section turns to another promissory note: the island-asymmetry.

5 Long-Distance Dependencies and Islands

5.1 Introduction

A primary piece of evidence for the syntactic dichotomy of clefts_{+C} and clefts_{-C} has been island-sensitivity: clefts_{+C} are sensitive to islands, while clefts_{-C} are not (Hoji 1990). The prevalent view in the literature is that clefts_{+C} involve syntactic movement, while clefts_{-C} do not (Sect. 2). The objective of this section is to defend our uniform analysis of clefts by explaining away the island-asymmetry in terms of how an interpretation is progressively built up.

For a start, Sect. 5.2 models long-distance clefts. To this end, we consider a possible combination of different actions to introduce structural underspecification. Sect. 5.3 examines the issue of island-sensitivity. Given the optionality of general actions, there is more than one way to process a focus item; the parser may induce an inverse LINK relation to process the focus (rather than inducing an unfixed node). This update is available for both types of clefts, but a well-formed tree arises only when the focus lacks a case particle. Finally, Sect. 5.4 concludes the present section.

5.2 Long-Distance Clefts

Chap. 2 (Sect. 5.7) reported that the Corpus of Spontaneous Japanese included 8 cases of long-distance clefts (7 clefts_{-C} and 1 cleft_{+C}). Consider (2.74), repeated as (5.74).

(5.74) Clefts, extracted from the CSJ (S00M0213)

[[Eikoku-to-Nihon-to ichiban chigau-na to] omo-tta
 [[UK-and-Japan-and most different-CFP COMP] think-PAST
 no]-wa terebi deshi-te...
 NO]-TOP TV COP-and

‘It is TV programmes that I thought were the most different between the UK and Japan, and...’

In our analysis of clefts (Sect. 3.3), the parser processes a focus item by running LOCAL *ADJUNCTION. Since an unfixed node induced by LOCAL *ADJUNCTION must be fixed within a **local** structure, it fails to deal with long-distance clefts. To take (5.74) as an example, LOCAL *ADJUNCTION disallows the focus *terebi* (= ‘TV’) to be the argument of the predicate *chigau-* (= ‘different’).

Kempson & Kiaer (2010) argue that long-distance scrambling is handled by the successive running of *ADJUNCTION and LOCAL *ADJUNCTION. This feeding relation of the two general actions, as will be argued, is applicable to long-distance clefts. To keep the illustration simple, let us analyse the simplex example (5.75).

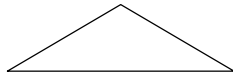
(5.75) Long-distance clefts_{-c}

[[_{e_i} hane-teiru to] Chihiro-ga i-tta no]-wa susuwatari_i da.
 [[hop-PROG COMP] C-NOM say-PAST NO]-TOP wandering-soot COP
 ‘It is wandering-soots that Chihiro said are hopping.’

Prior to *no-wa*, the parse of (5.75) gives rise to the tree (5.76). That is, *hane-teiru-* (= ‘hopping’) develops a propositional structure, and the complementiser *to* identifies the top node of this structure as an object node in a larger structure (see Sect. 4.2, Chap. 3 for clausal embedding). This larger structure is developed by *Chihiro-ga i-tta*. In (5.76), the propositional content of *hane-teiru-* is abbreviated as *a*.

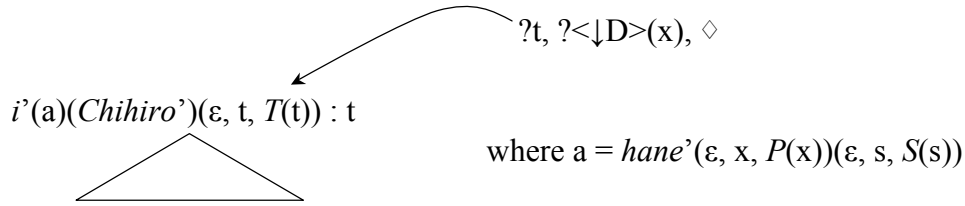
(5.76) Parsing [*Hane-teiru to*] *Chihiro-ga i-tta*

$i'(a)(\text{Chihiro}')(\epsilon, t, T(t)) : t, \diamond$



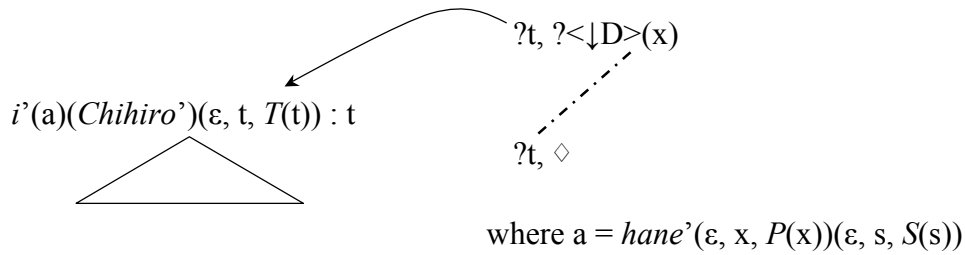
where $a = \text{hane}'(\epsilon, x, P(x))(\epsilon, s, S(s))$

Then, *no-wa* inversely LINKs the current node to a new type-*t*-requiring node. So far, the tree update process is as proposed in Sect. 3.

(5.77) Parsing [*Hane-teiru to*] *Chihiro-ga i-tta no-wa*

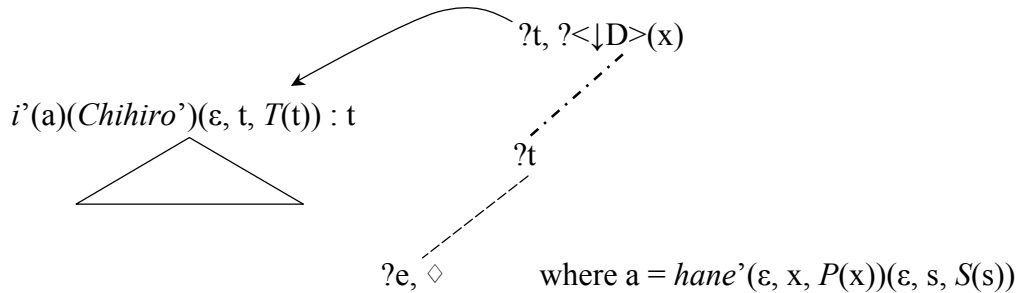
In Sect. 3, LOCAL *ADJUNCTION fired at this stage. But it forces the node of the focus to be resolved within the **local** structure, failing to participate in a long-distance dependency. This problem disappears if the parser first executes *ADJUNCTION at this point, inducing a type-t-requiring unfixed node that may be fixed **non-locally**.

(5.78) *ADJUNCTION



It is at this type-t-requiring unfixed node that LOCAL *ADJUNCTION is applied. That is, a type-e-requiring unfixed node is hung from the type-t-requiring unfixed node.

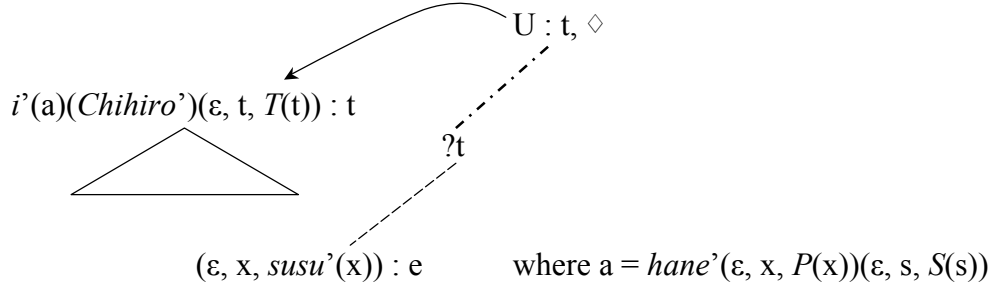
(5.79) LOCAL *ADJUNCTION



The rest of the process is as usual. The current node is decorated by the focus *susuwatari*. At this point, the requirement $?<\downarrow D>(x)$ is met by $(\varepsilon, x, \text{susu}'(x))$. (Note that the operator $<\downarrow D>$ declares that the variable x may be found at a node of arbitrary

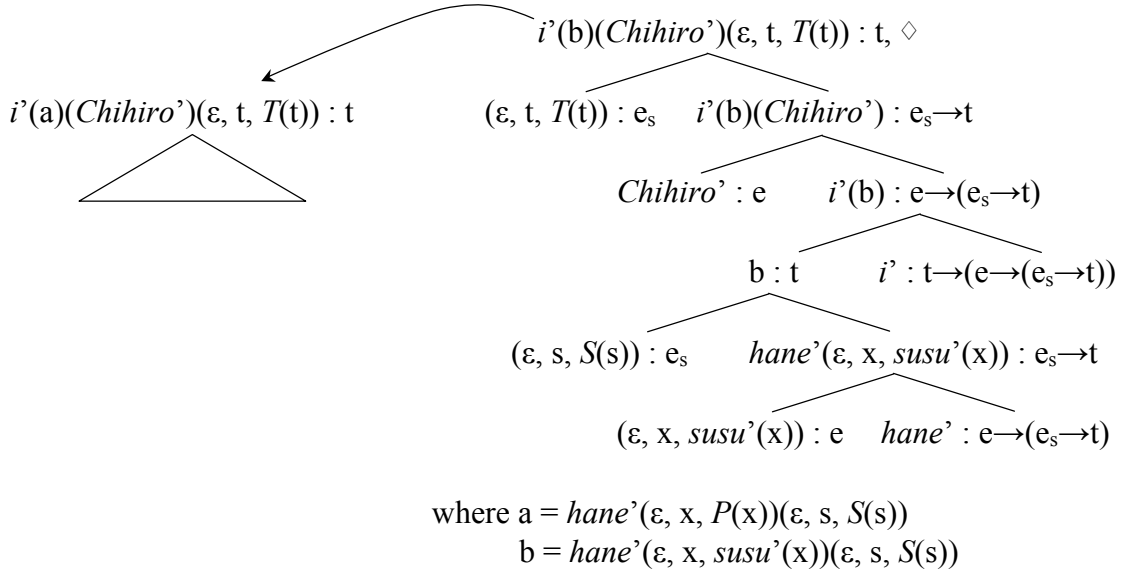
depth.) After the pointer $\diamond$ returns to the top node, the copula *da-* decorates the node with a meta-variable U and the logical type t .

(5.80) Parsing [*Hane-teiru to*] *Chihiro-ga i-tta no-wa susuwatari da*



The type- t meta-variable U licenses REGENERATION, re-running the actions that built the previous tree. In particular, the lexical actions of *hane-* (= ‘hop’) create a subject node, with which the node of the focus is merged (i.e., UNIFICATION). After ELIMINATION (i.e., functional application, type deduction), the final state (5.81) emerges.

(5.81) ELIMINATION (five times)



In this way, the applications of *ADJUNCTION and LOCAL *ADJUNCTION in this order handle long-distance clefts without giving up the incremental, monotonic tree-update (see Kempson & Kiaer 2010 for long-distance scrambling). Since the type- t -requiring node introduced by *ADJUNCTION may be fixed at **any** position in the tree (but

not across a LINK boundary; see Sect. 5.3), our analysis correctly predicts long-distance dependencies in clefts with clausal embedding of arbitrary depth.

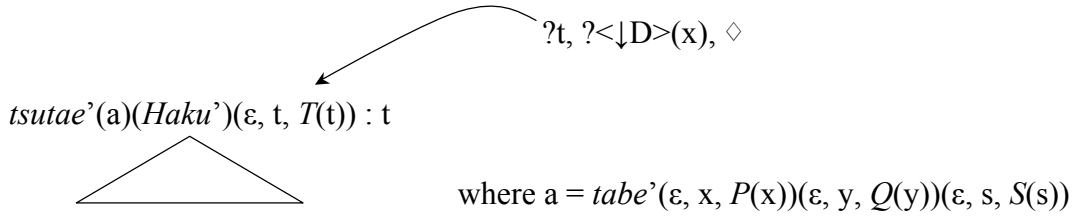
This account equally applies to multiple long-distance clefts as in (5.82).

(5.82) Multiple long-distance clefts

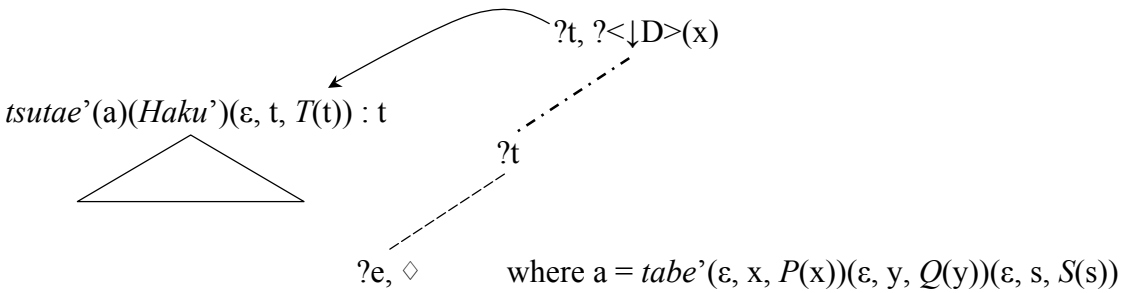
[[e_i e_j *tabe-ta to*] Haku-ga *tsutae-ta no*]-wa
 [[*eat-PAST COMP*] H-NOM *tell-PAST NO*]-TOP
Chihiro-ga_i onigiri-o_j da.
 C-NOM rice.ball-ACC COP
 Lit. ‘It is *Chihiro_i* rice balls_{*j*} that Haku told e_i ate e_j .’

The parse of (5.82) up to *no-wa* leads to the tree (5.83), where $(\epsilon, x, P(x))$ is the content of the object gap e_j and $(\epsilon, y, Q(y))$ is the content of the subject gap e_i . Then, the parser performs *ADJUNCTION and LOCAL *ADJUNCTION in this order.

(5.83) Parsing [*Tabe-ta to*] *Haku-ga tsutae-ta no-wa*



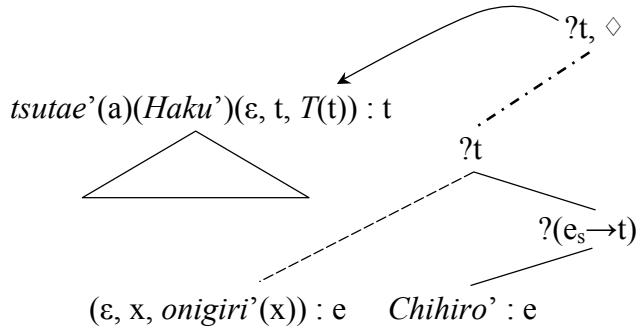
(5.84) *ADJUNCTION + LOCAL *ADJUNCTION



The current node is decorated by the first focus *Chihiro*, and the node is immediately resolved as a subject node by the nominative-case particle *ga*. Since the tree no longer contains an unfixed node, LOCAL *ADJUNCTION may fire once again to introduce an unfixed node for the second focus *onigiri* (= ‘rice ball’). This node, however, is just

marked (but not fixed) as an object node by the accusative-case particle *o*. This is because if this node were fixed at this point, the pointer $\diamond$ could not go up to the upper type-*t*-requiring node (Kempson & Kiaer 2010: 180); see the definition of the general action COMPLETION in the appendix.

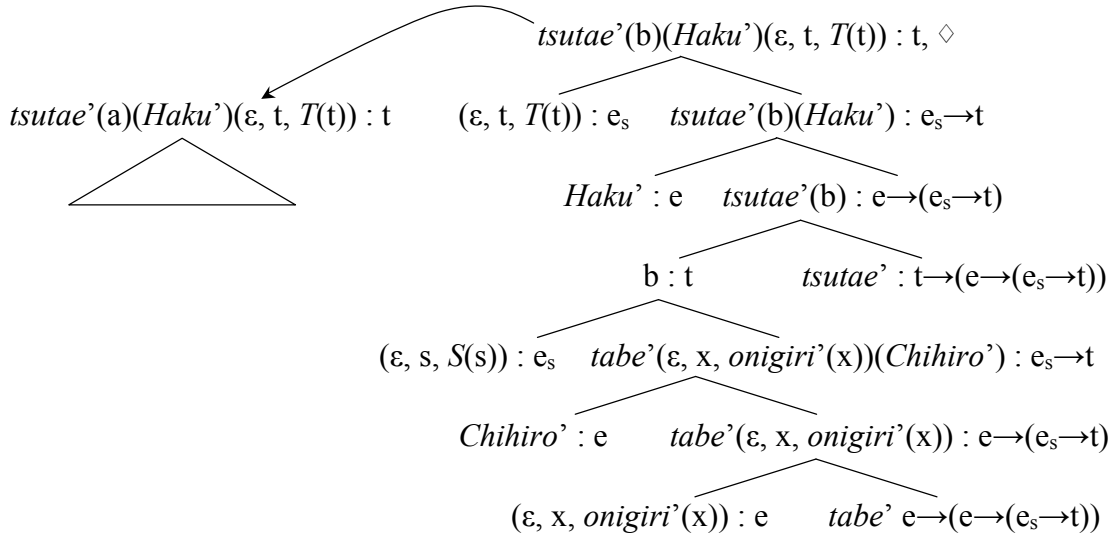
(5.85) Parsing [*Tabe-ta to*] *Haku-ga tsutae-ta no-wa Chihiro-ga onigiri-o*



where $a = \text{tabe}'(\epsilon, x, P(x))(\epsilon, y, Q(y))(\epsilon, s, S(s))$

Finally, the copula *da-* provides a propositional meta-variable; this licenses REGENERATION, repeating the actions that built the presupposition structure. During this reiteration, the type-*t*-requiring unfixed node is merged with the type-*t* node built by the re-run of the actions associated with the embedded clause *tabe-ta to* (i.e., UNIFICATION). Further, the re-run of the actions encoded in *tabe-* creates a subject node and an object node: the subject node collapses with the node of the focus *Chihiro*, and the object node is merged with the unfixed node of the focus *onigiri* (i.e., UNIFICATION). Finally, the tree is cleaned up by ELIMINATION.

(5.86) ELIMINATION (six times)



where $a = tabe'(\epsilon, x, P(x))(\epsilon, y, Q(y))(\epsilon, s, S(s))$
 $b = tabe'(\epsilon, x, onigiri'(x))(Chihiro')(\epsilon, s, S(s))$

This is exactly as expected, the application of *ADJUNCTION feeding that of LOCAL *ADJUNCTION.¹⁵ Given that the unfixed node of the final focus may be resolved by UNIFICATION (Sect. 4.4), it is predicted that the case particle *o* may be dropped off the final focus *onigiri* in (5.82). This prediction is borne out by (5.87). Some may find this string slightly degraded, but it is much better than the case where *Chihiro* is case-less, which is wholly unacceptable. This is also predicted; if *Chihiro* is case-less, the unfixed node introduced for *Chihiro* is not resolved and it cannot be separated from the unfixed node for *onigiri*, leading to inconsistency in the node description.

¹⁵ Koizumi (1995) notes that each focus must belong to the same local clause. Some studies account for this restriction structurally (Koizumi 2000, Takano 2002), but the restriction is controversial (Fukui & Sakai 2003: 339). Further, Hiraiwa & Ishihara (2002, 2012) report that the restriction is obviated when clefts are produced with certain intonation, suggesting that it would be better to treat the condition non-structurally. Our analysis predicts multiple foci may belong to distinct clauses: if a focus is parsed at the type-t-requiring node induced by *no-wa*, it belongs to a matrix clause; if it is parsed at the type-t-requiring node induced by *ADJUNCTION, it belongs to an embedded clause. A question is how to capture the strong clause-mate tendency. Of note is Kiaer (in press), who proposes a pragmatically-motivated performance constraint on DS structure building. The clause-mate tendency may be tackled from this viewpoint.

(5.87) Multiple long-distance clefts

[[e_i e_j tabe-ta to] Haku-ga tsutae-ta no]-wa
 [[eat-PAST COMP] H-NOM tell-PAST NO]-TOP
 Chihiro-ga_i onigiri_j da.
 C-NOM rice.ball COP
 Lit. 'It is Chihiro_i rice balls_j that Haku told e_i ate e_j.'

5.3 Island-Involving Clefts

The last sub-section demonstrated that our account predicts long-distance dependencies in both clefts_{+C} and clefts_{-C}. When it comes to island-sensitivity, however, the two types of clefts show a clear contrast. Since Hoji's (1990) seminal work, it has been widely held that clefts_{+C} are sensitive to islands while clefts_{-C} are not. Compare the cleft_{+C} (5.7) with the cleft_{-C} (5.8), repeated here as (5.88) and (5.89) respectively.

(5.88) Clefts_{+C}

*[[NP [CP e_i e_j ki-teiru] hito_i]-ga shiawase-souna no]-wa
 [[[wear-PRES] person]-NOM happy-look NO]-TOP
 kono-fuku-o_j da.
 this-cloth-ACC COP
 Int. 'It is this cloth x_j that the person who wears x_j looks happy.'

(5.89) Clefts_{-C}

[[NP [CP e_i e_j ki-teiru] hito_i]-ga shiawase-souna no]-wa
 [[[wear-PRES] person]-NOM happy-look NO]-TOP
 kono-fuku_j da.
 this-cloth COP
 Lit. 'It is this cloth x_j that the person who wears x_j looks happy.'

Chap. 2 (Sect. 5.7) pointed out that the Corpus of Spontaneous Japanese included one possible instance of island-involving clefts. The example (2.76) is reproduced here as (5.90). It is possible to assume that the focus *Asakusa* may be associated with the gap

located within the adjunct island *tabete* (= ‘eat’). Note that (5.90) is an instance of clefts_{-C}, not clefts_{+C}.

(5.90) Clefts, extracted from the CSJ (S03F0087)

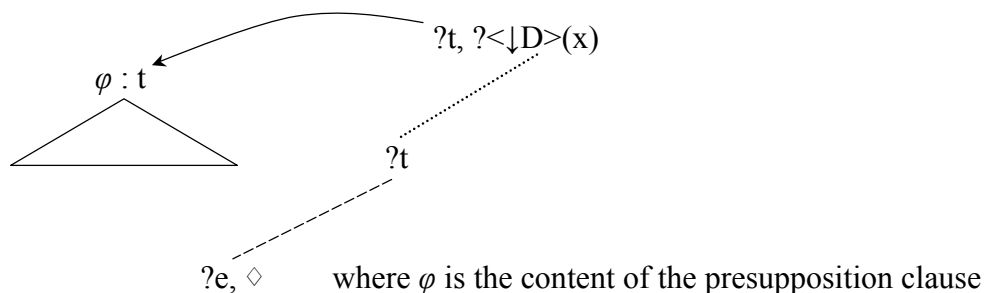
[[*tabete*] *tanoshii no*]-*wa* *yappari Asakusa desu...*

[[*eat*] *enjoyable NO*]-TOP *after.all Asakusa COP*

Lit. ‘It is after all Asakusa_i that people have fun when they eat in *e_i*...’

The feeding applications of *ADJUNCTION and LOCAL *ADJUNCTION (see the last sub-section) model long-distance dependencies, but not island-insensitivity. This is because the unfixed node introduced by *ADJUNCTION may be settled non-locally but **not** across a LINK boundary. Another option is to resort to the feeding applications of GENERALISED ADJUNCTION and LOCAL *ADJUNCTION.

(5.91) GENERALISED ADJUNCTION + LOCAL *ADJUNCTION



But this tree update does not work. First, after the focus is parsed, the pointer $\diamond$ cannot move up to the **higher** type-*t*-requiring node in order to process the copula *da-*; this is because the pointer movement across a globally unfixed relation is blocked by the constraint on pointer movement (Cann et al. 2005: 91).¹⁶ Second, the copula *da-* may be parsed at the **lower** type-*t*-requiring node, but this brings about inconsistency in the tree; I return to this point in Sect. 6.4.3 (see the discussion about (5.149)).

At first glance, this may seem to impose island-sensitivity on **both** types of clefts, in DS terms the update within a single tree (i.e., not across a LINK boundary). If

¹⁶ This constraint is encoded in the general action COMPLETION. As defined in the appendix, the rule formulation does not involve the operator $\langle U \rangle$.

this were the whole story, both (5.88) and (5.89) would be ungrammatical, the contrast being left as a mystery. But there is an alternative tree update: after *no-wa* initiates an inverse LINK relation, the parser may invoke a **second** inverse LINK relation to process a focus expression. In what follows, I shall illustrate how this alternative tree transition predicts the asymmetry between (5.88) and (5.89).

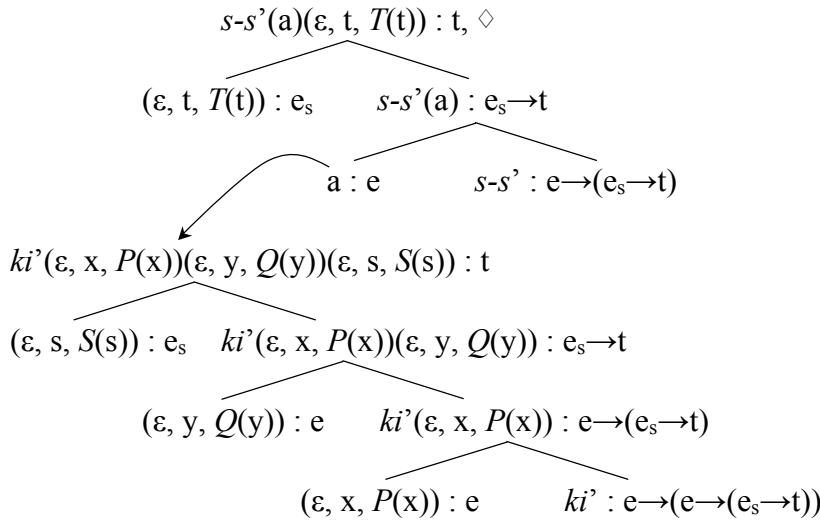
5.3.1 Clefts_C

Let us start with the cleft_C (5.89), repeated as (5.92). The parse of the presupposition clause builds the propositional tree (5.93), where *a* stands for the term of the complex NP, [*ki-teiru hito*]. (ϵ , y , $Q(y)$) is the content of the gap in the relative clause *ki-teiru*, and (ϵ , x , $P(x)$) is the content of the gap in the cleft. The tree update is as set out in Chap. 3 (Sect. 5.2). To recapitulate the main points, (i) the relative clause *ki-teiru* yields a propositional tree, and LINK INTRODUCTION introduces an inverse LINK relation to a type- ϵ -requiring node; (ii) this node is decorated by the head noun *hito* (= ‘person’); (iii) LINK ENRICHMENT incorporates the content of the relative clause into the content of the head noun, as a result of which the composite term *a* is constructed; (iv) this type- ϵ node is identified as a subject node within a new tree by the nominative-case particle *ga*; (v) this new tree is then fleshed out by the predicate *shiawase-souna*- (= ‘look happy’).

(5.92) Clefts_C

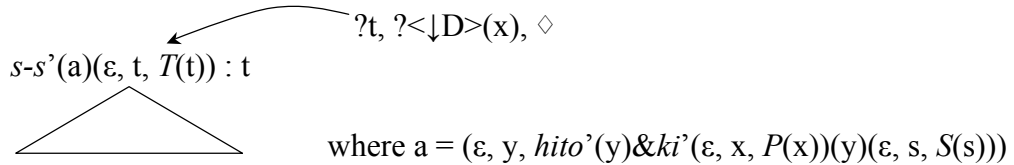
[[NP	[CP	e_i	e_j	ki-teiru]	hito _i]-ga	shiawase-souna	no]-wa
[[	[			wear-PRES]	person]-NOM	happy-look	NO]-TOP
kono-fuku _j				da.			
this-cloth				COP			

Lit. ‘It is this cloth x_j that the person who wears x_j looks happy.’

(5.93) Parsing [*Ki-teiru hito*]-ga shiawase-souna

where $a = (\varepsilon, y, hito'(y) \& ki'(\varepsilon, x, P(x))(y)(\varepsilon, s, S(s)))$

Then, *no-wa* initiates an inverse LINK relation from the current type-*t* node to a fresh type-*t*-requiring node. So far, the tree transition is as claimed in Sect. 3. (The whole tree in (5.93) is schematised by a triangle in (5.94).)

(5.94) Parsing [[*Ki-teiru hito*]-ga shiawase-souna no]-wa

What the parser does at the next step, according to Sect. 3, is to execute LOCAL *ADJUNCTION to introduce a type-*e*-requiring unfixed node for the upcoming focus. But there is **another** possible tree update. Recall that general actions are **optional**; that is, as long as the input condition (as defined in the IF-block) is met, the parser may run any general action at any point. I propose that the focus may be parsed by LINK transitions. To this end, I shall define the pair of LINK INTRODUCTION and LINK ENRICHMENT.¹⁷

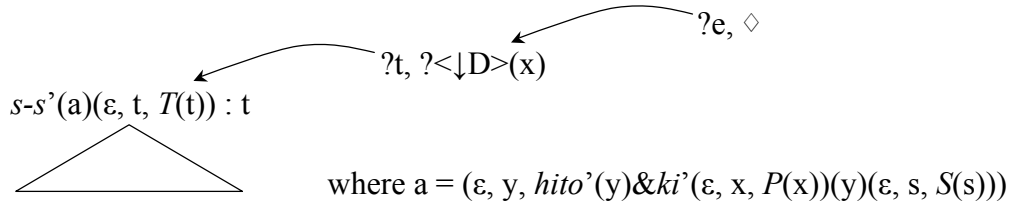
¹⁷ As we shall see below in the text, these two rules are formulated quite generally. Thus, they are used not only for clefts but also for stripping and sluicing (Sect. 6.4). Further applications of the rules to other constructions will be explored in future research.

First, the parser may run the general action LINK INTRODUCTION, as defined in (5.95), to initiate an inverse LINK relation from a type-t-requiring node to a new type-e-requiring node. This general action updates the tree (5.94) into (5.96).

(5.95) LINK INTRODUCTION (rule formulation)

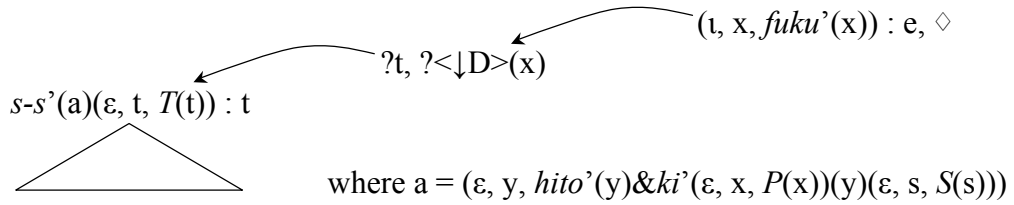
IF ?t
 THEN make/go(<L⁻¹>); put(?e)
 ELSE ABORT

(5.96) LINK INTRODUCTION



It is at this type-e-requiring node that the focus item *kono-fuku* is scanned.

(5.97) Parsing [[*Ki-teiru hito*]-ga *shiawase-souna no*]-wa *kono-fuku*

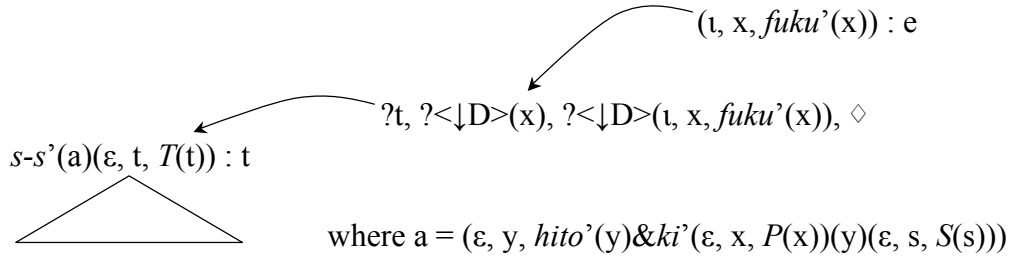


Second, LINK ENRICHMENT, as defined in (5.98), returns the pointer $\diamond$ to the type-t-requiring node and provides $?<\downarrow D>(\iota, x, fuku'(x))$ in virtue of having decorated the type-e node with this term. Notice that this requirement involves the operator $<D>$, which governs the **weakest** structural relation. Thus, the requirement may be satisfied by a node within an island (i.e., a node within a LINKed tree).

(5.98) LINK ENRICHMENT (rule formulation)

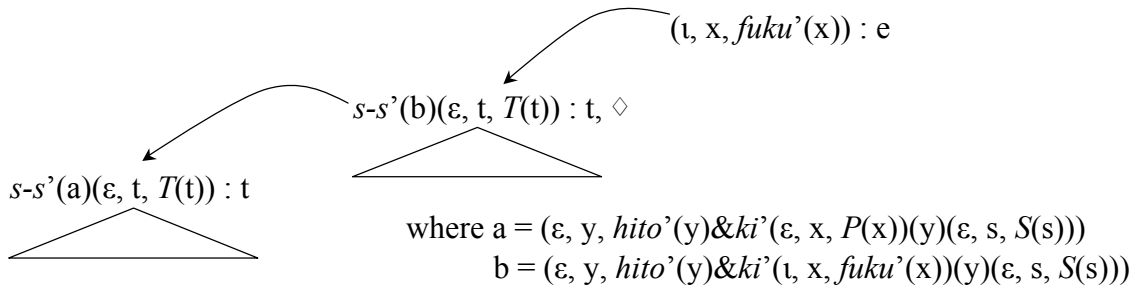
IF $\alpha : e$
 THEN IF $<L>?t$
 THEN go(<L>); put($?<\downarrow D>(\alpha)$)
 ELSE ABORT
 ELSE ABORT

(5.99) LINK ENRICHMENT



The rest of the process is as usual. The copula *da-* provides a propositional meta-variable and licenses REGENERATION, re-running the actions that constructed the presupposition tree. In particular, the lexical actions encoded in *ki-* (= ‘wear’) introduce an object node with a meta-variable, to be substituted with $(\iota, x, fuku'(x))$. This process satisfies the requirements $?<\downarrow D>(\iota, x, fuku'(x))$ and $?<\downarrow D>(x)$ **simultaneously**. After ELIMINATION is performed, the final tree state (5.100) is engendered.

(5.100) ELIMINATION (five times)



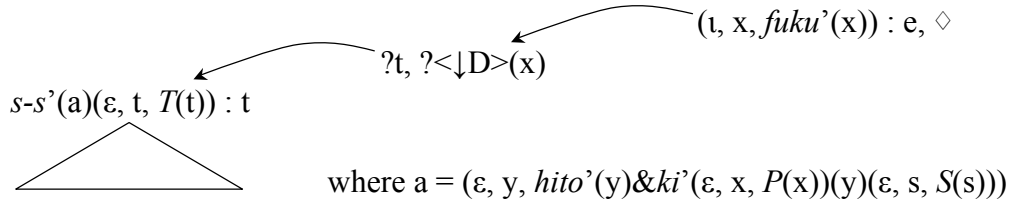
This LINK-transition based update predicts that island-involving clefts_{-C} are grammatical, characterising island-insensitivity of clefts_{-C}. As will be illuminated in the next sub-section, this alternative tree update does not lead to a well-formed tree state in clefts_{+C}, characterising island-sensitivity of clefts_{+C}.

5.3.2 Clefts_{+C}

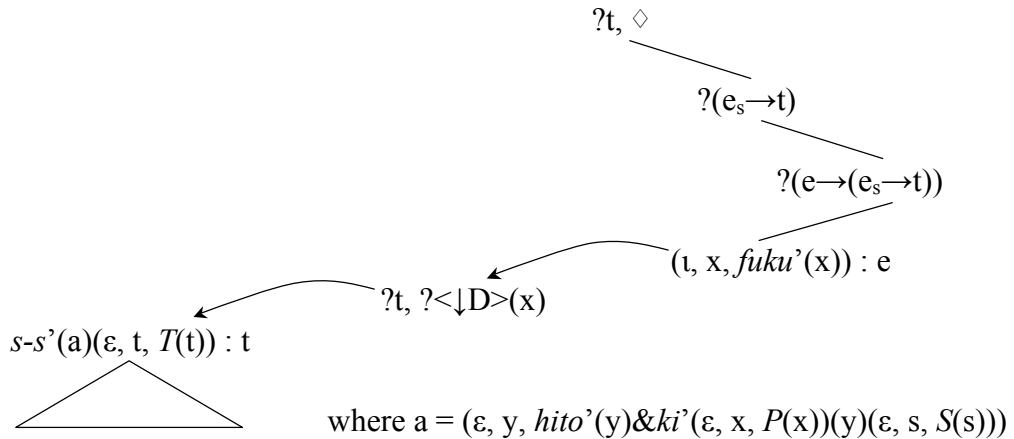
The example of island-involving clefts_{+C} (5.88) is reproduced here as (5.101). The parse of the string up to the focus *kono-fuku* (= ‘this cloth’) yields the tree (5.102). (This tree is identical to (5.97).)

(5.101) Clefts_{+C}

*[[_{NP} [_{CP} e_i e_j *ki-teiru*] *hito*]_i]-ga *shiwase-souna no*]-wa
 [[[_{CP} [_{VP} *wear-PRES*] *person*]-NOM *happy-look* NO]-TOP
kono-fuku-o da.
 this-cloth-ACC COP
 Int. 'It is this cloth x_j that the person who wears x_j looks happy.'

(5.102) Parsing [[*Ki-teiru hito*]-ga *shiwase-souna no*]-wa *kono-fuku*

In (5.102), the parser could execute LINK ENRICHMENT, moving the pointer $\diamond$ to the type-t-requiring node. This node, however, is not suited for processing the accusative-case particle *o*. So, the parser has to skip LINK ENRICHMENT and parses the accusative-case particle *o* at the current type-e node. The case particle builds a partial structure bottom-up, as in (5.103) (see Sect. 5.2, Chap. 3).

(5.103) Parsing [[*Ki-teiru hito*]-ga *shiwase-souna no*]-wa *kono-fuku-o*

In (5.103), if the copula *da-* is scanned at the current node, the lower type-t-requiring node remains undeveloped. On the other hand, if *da-* is scanned at the lower type-t-requiring node (after the pointer movement), the higher type-t-requiring node is not

developed. Either way, outstanding requirements are left in the tree and the well-formed state **cannot** emerge. Therefore, island-involving clefts_{+C} are ungrammatical, which, in turn, models island-sensitivity of clefts_{+C}.

In a nutshell, the optional nature of general actions allows the parser to run the pair of LINK INTRODUCTION and LINK ENRICHMENT in parsing the focus. This update leads to a well-formed state only in clefts_{-C} because the parse of a case particle at the inversely LINKed node ends up with outstanding requirements. It thus follows that clefts_{+C} are island-sensitive and clefts_{-C} are not. The LINK-based parsing of the focus is available for **both** clefts_{+C} and clefts_{-C}; it is simply that if the surface sequence contains a case particle attached to the focus (i.e., clefts_{+C}), a well-formed tree will **not** obtain. Unlike previous studies (Hiraiwa & Ishihara 2012, Hoji 1990, Kizu 2005), the dynamic account solves the island puzzle **without** stipulating two distinct structures for the two types of clefts and hypothesising ambiguity in the cleft string (especially, ambiguity of the particle *no*). This uniform analysis accords with the corpus indication that the two types of clefts should not be structurally distinguished (Sect. 5.3, Chap. 2).

5.3.3 The Copula Revisited

The very ‘optionality’ of general actions, however, threatens our analysis of multiple foci. As has been argued, there are two methods of processing a focus: (i) structural underspecification (i.e., LOCAL *ADJUNCTION) and (ii) LINK transitions (i.e., LINK INTRODUCTION plus LINK ENRICHMENT). But if these strategies were jointly utilised, it would wrongly predict that clefts_{-C} with multiple foci are grammatical. Consider (5.65), repeated here as (5.104).

(5.104) Clefts_{-C}

*[Haku-ga	e _i	e _j	age-ta	no]-wa	Chihiro _j	onigiri _i	da.
[H-NOM			give-PAST	NO]-TOP	C	rice.ball	COP

Int. ‘It is rice balls_i to Chihiro_j that Haku gave e_i e_j.’

If *Chihiro* were processed in terms of structural underspecification and *onigiri* were processed in terms of LINK transitions (or vice versa), there would be only a single unfixed node, and the Unique-unfixed-node Constraint would not be violated. Thus, (5.104) would be incorrectly predicted to be grammatical.

A possible solution is to enrich the entry of the copula *da-*. The original entry (5.40) is repeated here as (5.105). I amend the original version as in (5.106).

(5.105) Entry of the copula *da-* (preliminary version)

```
IF      ?t
THEN    put(U : t)
ELSE    ABORT
```

(5.106) Entry of the copula *da-* (final version)

```
IF      ?t
THEN    IF      <↓+>T
          THEN    put(U : t, <L-1>⊥)
          ELSE    put(U : t)
ELSE    ABORT
```

In (5.106), $\langle \downarrow_+ \rangle T$ declares that there is a dominated node.¹⁸ This is satisfied if a focus has been parsed by means of structural underspecification. In this case, *da-* provides a meta-variable *U*, the type *t*, and the label $\langle L^{-1} \rangle \perp$. This label declares that no node is inversely LINKed from the current node; this prohibits a focus from being parsed by means of the **combination** of structural underspecification and LINK transitions. Thus, (5.104) is correctly ruled out. On the other hand, if a focus has been processed **only** at an inversely LINKed node as in (5.97), *da-* simply provides a meta-variable *U* and the type *t*. The modified entry (5.106)¹⁹ also plays a role in predicting the island-contrast in stripping and sluicing (see Sect. 6.4).

¹⁸ The operator $\langle \downarrow_+ \rangle$ involves the Kleene plus $+$. Formally, $\langle \downarrow_+ \rangle$ is defined as $\langle \downarrow \downarrow^* \rangle$.

¹⁹ The copula *da-* has diverse functions (Imada 2009, Nishiyama 2003), and even (5.106) has to comprise further constraints. For instance, it is not immediately obvious how the entry (5.106) accommodates ‘predicational’ copula sentences. Yet, I shall adopt (5.106) for the purpose of the present thesis.

5.4 Summary

We have inquired into the issues of long-distance dependencies and island-sensitivity. Starting with long-distance clefts, despite the initial impression that they might be resistant to incremental and monotonic modelling, they are dealt with by considering a possible combination of *ADJUNCTION and LOCAL *ADJUNCTION.

Turning to island-sensitivity, the optional nature of general actions reveals that there is more than one strategy to process the focus. That is, instead of introducing an unfixed node, the parser may invoke an inversely LINKed node to parse the focus. This alternative tree update is available for both types of clefts, but the tree update converges only when the focus is case-less. This solves the island puzzle: clefts_{+C} are sensitive to islands whereas clefts_{-C} are not.

So far, we have developed a uniform analysis of clefts and accounted for the multiple foci phenomenon and the island asymmetry without abandoning uniformity in explanation. In the next section, this unitary treatment is further corroborated by its applicability elsewhere: stripping and sluicing.

6 Implications for Ellipsis

6.1 Introduction

Some previous works have handled stripping and sluicing on a par with clefts (Fukui & Sakai 2003, Hiraiwa & Ishihara 2012, Kizu 2005). In addition, Fukaya (2003, 2007, 2012) notes that case-marking affects island-sensitivity of these ellipsis constructions, arguing that (i) there are two types of stripping/sluicing and that (ii) they are to be assigned distinct structures (i.e., one which involves movement and another which does not). This situation reminds us of the prevailing, bifurcated view of clefts. As a further affinity, we shall see the same case-marking pattern in stripping and sluicing. These parallelisms between clefts, stripping, and sluicing provide us with another promising

test case. In this section, we shall make use of our analysis of clefts for articulating a uniform account of stripping/slucing and solving the case-marking and island puzzles without sacrificing uniformity.

Sect. 6.2 proposes a unitary account of stripping/slucing. Sect. 6.3 shows that the same case-marking facts of stripping/slucing are seen with clefts. This is predicted by the mechanism of ‘structural underspecification and resolution,’ as conditioned by the Unique-unfixed-node Constraint, a corollary of the tree logic LOFT. Sect. 6.4 turns to long-distance dependencies and islands. First, long-distance stripping/slucing are treated by the feeding relation of *ADJUNCTION and LOCAL *ADJUNCTION, as in long-distance clefts. Second, the island puzzle is solved on the same grounds as clefts: (i) the focussed NP may be parsed by means of LINK transitions and (ii) this LINK-based parsing leads to a well-formed state only if the focussed NP is not case-marked. Finally, Section 6.5 summarises the present section.

6.2 The Analysis

Like clefts, stripping and sluicing involve the copula *da-*. In Sect. 3.3, *da-* was regarded as a type-*t* pro-form, licensing the general action REGENERATION (i.e., re-run of a set of actions to build a propositional tree). We find that this pro-form analysis of *da-* and the consequent re-running of actions equally extend to these ellipsis cases. What will be achieved is a unitary account of stripping and sluicing, ‘unitary’ in the sense that (i) **both** stripping and sluicing are analysed by the same machinery and that (ii) for **each** of stripping and sluicing, the analysis remains the same regardless of whether the focus expression is case-marked or not.

6.2.1 Stripping

Stripping in Japanese is an ellipsis construction consisting of an NP and the copula *da-* (Fukaya, 2003, 2007, 2012, Fukaya & Hoji 1999, Sakai 2000; see Hankamer & Sag 1976 for English data). Consider (5.107).

(5.107) Stripping

- A: Sen-ga wara-tta nodayone?
 S-NOM smile-PAST Q
 ‘Sen smiled, right?’
- B: Iya, Haku da.
 no H COP
 ‘No, Haku.’ (= ‘No, Haku smiled.’)

The stripping (5.107)B is similar to the cleft (5.108) **except** that the presupposed information is conveyed by the pre-*no-wa* clause in clefts, whereas it is provided by the antecedent string in stripping.

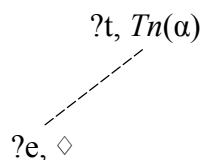
(5.108) Clefts

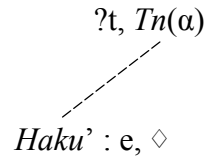
- [Wara-tta no]-wa Haku da.
 [smile-PAST NO]-TOP H COP
 ‘It is Haku that smiled.’

In clefts, the pre-*no-wa* part sets a context relative to which the focus is interpreted. I shall argue that in stripping, the context is established by re-running the previous actions which built the structure of the antecedent string.

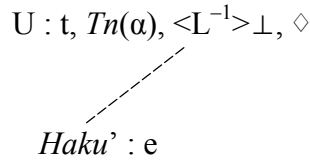
The tree update for the stripping (5.107)B proceeds as follows. The AXIOM sets out an initial node, and LOCAL *ADJUNCTION induces a type-e-requiring unfixed node. This node is decorated by the focussed NP *Haku*. (The particle *iya* (= ‘no’) is ignored.)

(5.109) LOCAL *ADJUNCTION



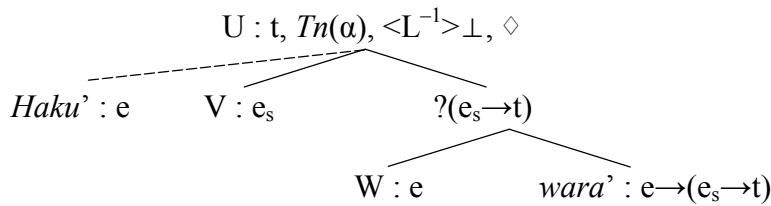
(5.110) Parsing *Haku*

The pointer $\diamond$ returning to the top node, the copula *da-* annotates the type-t-requiring node with a meta-variable U , the type t , and the label $\langle L^{-1} \rangle \perp$ (see the revised entry of the copula *da-* in (5.106)).

(5.111) Parsing *Haku da*

This type-t meta-variable licenses REGENERATION, the re-run of a set of actions to build up a propositional structure (Sect. 6, Chap. 3).²⁰ In the present case, such a set of actions is the lexical actions encoded in *wara-* (= ‘smile’) in the antecedent string (5.107)A. The re-run of the actions of *wara-* builds a propositional schema, where a situation node and a subject node are decorated with a type-e meta-variable.

(5.112) REGENERATION

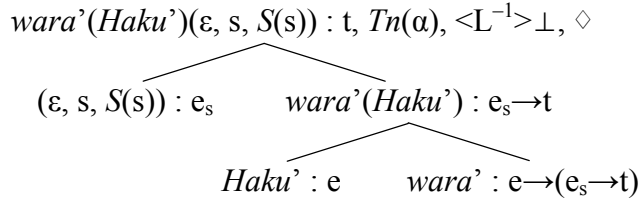


There is a single unfixed node of type-e and there is also a fixed node of type-e. The descriptions of these two nodes are merged by the general action UNIFICATION. The

²⁰ Alternatively, the parser may re-use the structure of the antecedent clause. As mentioned in Sect. 6, Chap. 3, the choice of re-using ‘structure’ or ‘actions’ leads to the strict/sloppy divide in ellipsis construal. The issues of strict/sloppy readings in stripping and sluicing in Japanese have been addressed in Fukaya (2007). I shall leave these issues for future work.

situation meta-variable V is then substituted with $(\epsilon, s, S(s))$. After ELIMINATION (i.e., functional application, type deduction) is run, the final tree state (5.113) emerges. The proposition at the top node represents the interpretation of stripping (5.107)B relative to the antecedent (5.107)A: ‘No, Haku smiled.’

(5.113) ELIMINATION (twice)



Finally, a prediction is made on the case-marking of stripping. In the above analysis, the unfixed node of *Haku* was resolved by UNIFICATION. But it could have been resolved lexically if an appropriate case particle were attached to the focus *Haku*. This predicts that stripping may be case-marked, as in (5.114)B.

(5.114) Case-marked stripping

- A: Chihiro-ga wara-tta nodayone?
 C-NOM smile-PAST Q
 ‘Chihiro smiled, right?’
- B: Iya, Haku-ga da.
 no H-NOM COP
 ‘No, Haku.’ (= ‘No, Haku smiled.’)

Thus, our analysis provides a unitary treatment of stripping regardless of whether a case particle is present. This unified analysis fares better than the previous study (Fukaya 2003, 2007), which posits distinct structures for case-marked and case-less stripping. A primary motivation for this non-unitary view is that the case-marking affects island-sensitivity in stripping. In Sect. 6.4, however, our uniform analysis addresses the island puzzle without relinquishing uniformity.

6.2.2 Sluicing

Sluicing is a construction where a matrix clause licenses the elliptical clause which consists of a *wh*-word, the copula *da*-,²¹ and the question marker *ka* (Takahashi 1994, Nishiyama et al. 1996, and subsequent works; see also Chung et al. (1995), Merchant (2001), and Ross (1969) for English data). Consider the example (5.115).

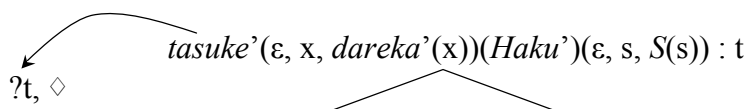
(5.115) Sluicing

Haku-wa	mukashi	kawa-de	dareka-o	tasuke-ta	ga,
H-TOP	in.the.past	river-in	someone-ACC	save-PAST	but
dare	(da)	ka	mattaku	omoidase-nai.	
who	(COP)	Q	at.all	remember-NEG	

‘Haku saved someone in the river in the past, but he cannot remember who.’

The initial clause builds a propositional tree, and the coordination marker *ga* LINKs it to a type-*t*-requiring node (Cann et al. 2005: 208); see also Sect. 4.3, Chap. 4. (The topic marker *wa* attached to *Haku* is treated as a nominative-case marker, and the adjuncts *mukashi* (= ‘in the past’) and *kawa-de* (= ‘in the river’) are ignored.)

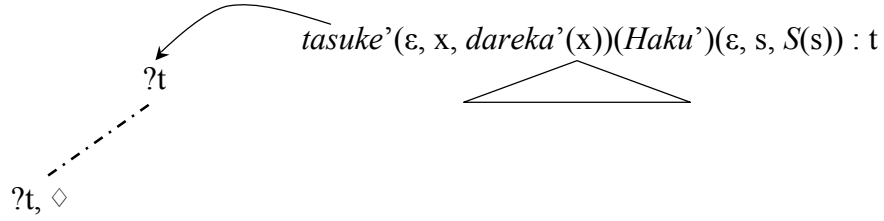
(5.116) Parsing *Haku-wa mukashi kawa-de dareka-o tasuke-ta ga*



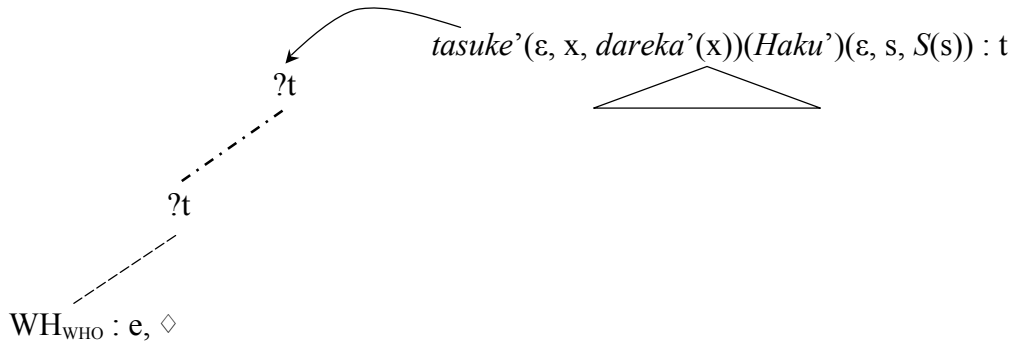
The ellipsis part *dare (da) ka* is in a **subordinate** structure, embedded under the clause *mattaku omoidase-nai*. Thus, *ADJUNCTION introduces a type-*t*-requiring unfixed node, as in (5.117).

²¹ The copula *da*- in sluicing may be dropped (Nishiyama et al. 1996: 341) but *da*- in stripping is difficult to omit when it is embedded. I have no decisive explanation for the relative easiness of omitting *da*- in sluicing, but it may be due to the question marker *ka*. Notably, if *ka* is present, *da*- in stripping can be dropped even if it is embedded; see Hiraiwa & Ishihara (2012: footnote 28) for essentially the same observation.

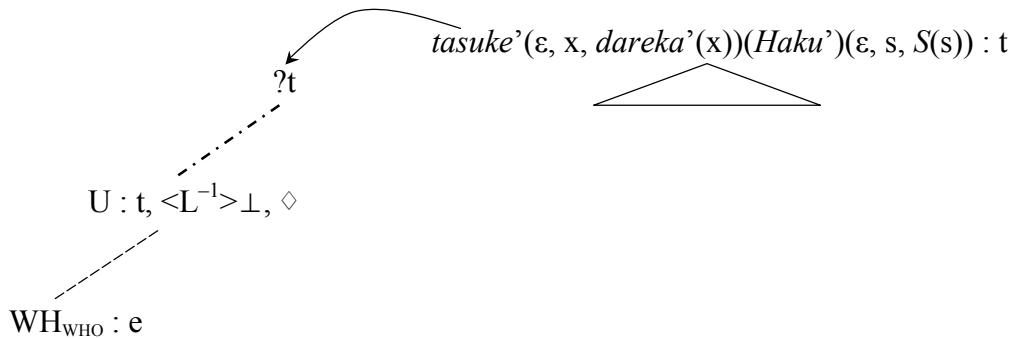
(5.117) *ADJUNCTION



From this stage on, the analysis of sluicing is essentially the same as that of stripping. LOCAL *ADJUNCTION introduces an unfixed node for *dare* (= ‘who’). The content of *dare* is a *wh*-meta-variable WH_{WHO} (Kempson et al. 2001: Chap. 5).

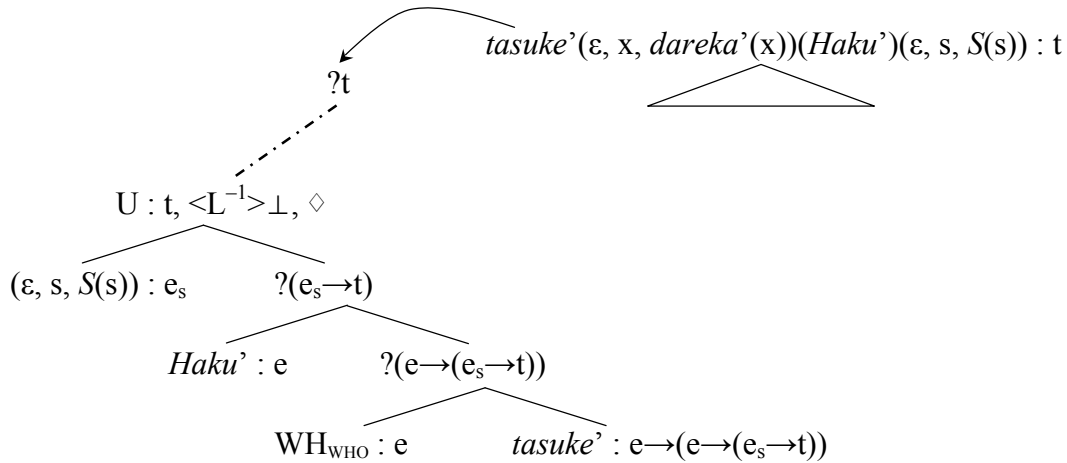
(5.118) Parsing *Haku-wa mukashi kawa-de dareka-o tasuke-ta ga, dare*

Then, the copula *da-* provides a meta-variable U , the type t , and the label $\langle L^{-1} \rangle_{\perp}$ at the lower type- t -requiring node.

(5.119) Parsing *Haku-wa mukashi kawa-de dareka-o tasuke-ta ga, dare da*

The propositional meta-variable U licenses REGENERATION, repeating a set of previous actions that constructed the structure of the initial clause (see footnote 20). In particular, *tasuke-* (= ‘save’) creates an object node, and this node is merged with the unfixed node decorated with WH_{WHO} (i.e., UNIFICATION).

(5.120) UNIFICATION



After ELIMINATION is performed, the question marker *ka* marks the accruing proposition as an interrogative, and the pointer $\diamond$ goes up to the upper type- t -requiring node, where the matrix clause *mattaku omoidase-nai* is scanned. (The tree displays of these tree transitions are omitted.)

Sluicing is accounted for by the same mechanisms as those used for stripping and clefts: (i) the apparatus of ‘structural underspecification and resolution,’ (ii) the copula *da-* as a propositional pro-form, and (iii) the general action REGENERATION. In the tree update proposed, the unfixed node of *dare* (= ‘who’) was settled by the general action UNIFICATION. But if an appropriate case particle is attached to the *wh*-word *dare*, the node may be lexically settled. That is to say, it is predicted that case-marked sluicing is possible, as ascertained in (5.121).

(5.121) Case-marked sluicing

Haku-wa mukashi kawa-de dareka-o tasuke-ta ga,
 H-TOP in.the.past river-in someone-ACC save-PAST but
 dare-o (da) ka mattaku omoidase-nai.
 who-ACC (COP) Q at.all remember-NEG
 ‘Haku saved someone in the river in the past, but he cannot remember who.’

Our account deals with sluicing in a unitary fashion irrespective of the case-marking, unlike previous accounts (Fukaya 2003, 2007, 2012; see also Takahashi 1994), which stipulate that case-marked sluicing structurally differs from case-less sluicing in a substantive way. Fukaya (2003, 2007, 2012) observes that the case-marking in sluicing is correlated with island-sensitivity. In Sect. 6.4, however, these island data are shown to follow from our uniform analysis.

6.3 Multiple Foci

In this sub-section, we find the strikingly similar case-marking pattern of stripping and sluicing as with clefts. Notably, when there is more than one focussed NP, only the final focus may be case-less. This partial case-marking fact is hard to capture in the previous studies since different structures are postulated depending on whether the focussed NP is case-marked or not (Fukaya 2003, 2007, 2012; see also Takahashi 1994).

6.3.1 Multiple Stripping

According to Sect. 6.2.1, stripping and clefts are analysed essentially in the same way; the chief difference is that REGENERATION in clefts re-uses the set of actions that built the presupposition tree while REGENERATION in stripping re-uses the set of actions that built the antecedent tree. This difference comes in, however, **after** the copula *da-* is scanned. Thus, the parsing process of focussed NPs in stripping is exactly the same as that in clefts. This identical process makes a number of predictions on stripping.

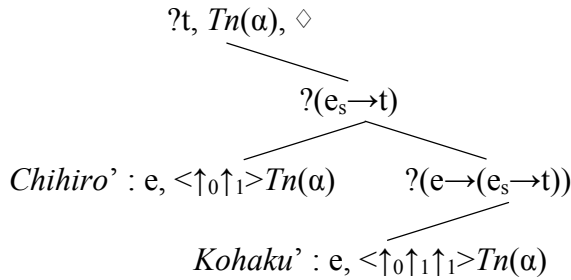
First, multiple stripping should be possible as long as each item is marked by an appropriate case particle.

(5.122) Stripping

- A: Sen-ga Haku-o aishi-teiru nodayone?
 S-NOM H-ACC love-PRES Q
 ‘Sen loves Haku, right?’
- B: Iya, Chihiro-ga Kohaku-o da.
 no C-NOM K-ACC COP
 ‘No, Chihiro loves Kohaku.’

In (5.122)B, LOCAL *ADJUNCTION introduces a type-e-requiring unfixed node for the initial NP *Chihiro*. Since this node is immediately **resolved** as a subject node by the nominative-case particle *ga*, LOCAL *ADJUNCTION can be applied once again for the second NP *Kohaku*. The unfixed node of *Kohaku* is then resolved as an object node by the accusative-case particle *o*.

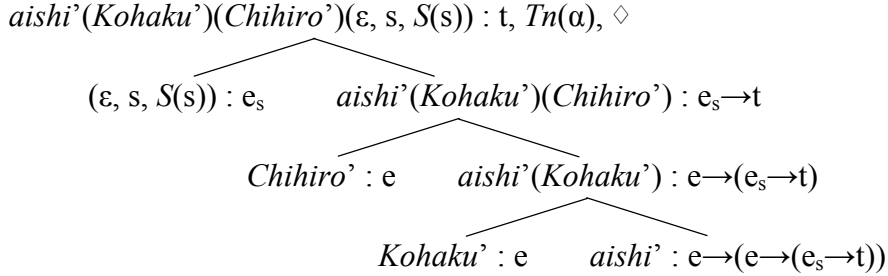
(5.123) Parsing *Chihiro-ga Kohaku-o* (ignoring *iya* (= ‘no’))



The copula *da-* posits a meta-variable and licenses REGENERATION, which reiterates the previous actions that built the tree of the antecedent string (5.122)A. In particular, *aishi-* (= ‘love’) projects a propositional schema with a situation node, a subject node, and an object node. The subject and object nodes collapse with the two pre-existing nodes in (5.123). A situation meta-variable is substituted with (ϵ , s , $S(s)$). Finally, ELIMINATION (i.e., functional application, type-deduction) yields the tree (5.124). The proposition at

the top node represents the interpretation of (5.122)B relative to the antecedent string (5.122)A: ‘No, Chihiro loves Kohaku.’

(5.124) ELIMINATION (three times)

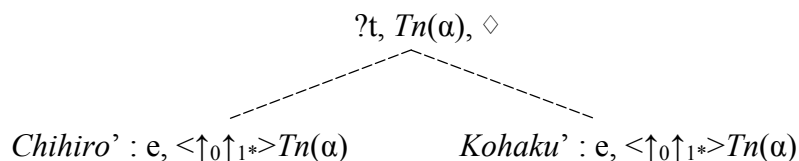


Second, if the focussed NPs in stripping are not case-marked, the string should be ungrammatical, as in (5.125)B.

(5.125) Stripping

- A: Sen-ga Haku-o aishi-teiru nodayone?
 S-NOM H-ACC love-PRES Q
 ‘Sen loves Haku, right?’
- B: *Iya, Chihiro Kohaku da.
 no C K COP
 Int. ‘No, Chihiro loves Kohaku.’

In (5.125)B, the initial NP *Chihiro* is parsed at a type-*e*-requiring unfixed node that was induced by LOCAL *ADJUNCTION. This unfixed node, however, is **not** resolvable since (i) *Chihiro* is case-less and (ii) the copula *da-* will not be parsed until the second focus *Kohaku* is parsed. So, the parser has to run LOCAL *ADJUNCTION once again in order to parse *Kohaku* **without** fixing the initial unfixed node. In (5.126), the tree ends up with multiple unfixed nodes with the same tree-node label $\langle \uparrow_0 \uparrow_1^* \rangle Tn(\alpha)$. This tree state gives rise to inconsistency in the node description; recall the Unique-unfixed-node Constraint (5.59), reproduced here as (5.127). Hence, ungrammaticality of (5.125)B. It should be emphasised here once again that the Unique-unfixed-node Constraint is not a stipulation but a general consequence of the tree logic LOFT.

(5.126) Parsing *Chihiro Kohaku*

(5.127) The Unique-unfixed-node Constraint

If supposedly distinct unfixed nodes have the same tree-node label but non-commensurate labels, the node description becomes inconsistent and the tree update crashes.

Third, the two NPs in (5.122)B may be swapped, for the order of these NPs does **not** affect the process of structural underspecification and resolution, provided that they are case-marked. In (5.128), the focus cluster *Kohaku-o Chihiro-ga* is in a reverse order. Note that the sequence *Haku-o Sen-ga* in the antecedent part is also in a reverse order to make the string fully acceptable; if the sequence in the antecedent part were still *Sen-ga Haku-o* as in (5.122), the stripping part (5.128)B seems slightly degraded, though it would be far from unacceptable.

(5.128) Stripping

A:	Haku-o	Sen-ga	aishi-teiru	nodayone?
	H-ACC	S-NOM	love-PRES	Q
	'Sen loves Haku, right?'			
B:	Iya,	Kohaku-o	Chihiro-ga	da.
	no	K-ACC	C-NOM	COP
	'No, Chihiro loves Kohaku.'			

In (5.128)B, LOCAL *ADJUNCTION induces an unfixed node for the first NP *Kohaku*, and it is immediately **resolved** as an object node by the accusative-case particle *o*. Since an unfixed node no longer exists in the tree, LOCAL *ADJUNCTION may be run once again to induce an unfixed node for the second NP *Chihiro*. This unfixed node is immediately resolved as a subject node by the nominative-case particle *ga*. The output is identical to

the tree (5.124), which makes sure that (5.122)B and (5.128)B are truth-conditionally equivalent with respect to the antecedent (5.122)A and (5.128)A respectively.

Fourth, partially case-marked multiple stripping should be possible. That is, the **final** NP (and only the final NP) in multiple stripping may lack a case particle since the unfixed node for the **final** NP (and only the final NP) can be resolved by UNIFICATION. Thus, compare the contrast between (5.129)B and (5.129)B'.

(5.129) Stripping

- A: Sen-ga Haku-o aishi-teiru nodayone?
 S-NOM H-ACC love-PRES Q
 'Sen loves Haku, right?'
- B: Iya, Chihiro-ga Kohaku da.
 no C-NOM K COP
 'No, Chihiro loves Kohaku.'
- B': *Iya, Chihiro Kohaku-o da.
 no C K-ACC COP

In (5.129)B, LOCAL *ADJUNCTION induces an unfixed node for the initial NP *Chihiro*, and this node is immediately fixed as a subject node by the nominative-case particle *ga*. Since there is no unfixed node in the tree, LOCAL *ADJUNCTION may induce an unfixed node for the second NP *Kohaku*. This unfixed node is resolved by UNIFICATION after the copula *da-* licenses REGENERATION. By contrast, in (5.129)B', a problem arises when LOCAL *ADJUNCTION introduces an unfixed node for the first NP *Chihiro*. The unfixed node for *Chihiro* cannot be fixed in any way: (i) it cannot be fixed lexically for *Chihiro* is case-less; (ii) UNIFICATION cannot fire since the copula *da-* will come only after the next focus is scanned. So, the parser has to run LOCAL *ADJUNCTION for the second NP *Kohaku* without resolving the unfixed node of the initial NP *Chihiro*. This violates the Unique-unfixed-node Constraint; there are two unfixed nodes with the same tree-node label but non-commensurate formulae (i.e., the content of *Chihiro* and that of *Kohaku*).

Finally, multiple stripping with more than two focussed NPs should be possible if structural underspecification for each NP is fixed by case particles in an incremental fashion. In (5.130)B, noteworthy are the parentheses attached to the accusative-case particle *o*. This shows that the **final** NP (and only the final NP) may lack a case particle. This is because UNIFICATION, which is eventually licensed by the copula *da-*, may apply only to this **final** NP.

(5.130) Stripping

- A: Haku-ga Sen-ni kin-o age-ta nodayone?
 H-NOM S-DAT gold-ACC give-PAST Q
 ‘Haku gave gold to Sen, right?’
- B: Iya, Sen-ga Haku-ni dango(-o) da.
 no S-NOM H-DAT dumpling(-ACC) COP
 ‘No, Sen gave a dumpling to Haku.’

Whether the final NP *dango* (= ‘dumpling’) is case-marked, (5.130)B is much more acceptable than the other six possible cases, all of which are fully unacceptable: namely, the examples where (i) only *Sen* is case-less, (ii) only *Haku* is case-less, (iii) only *Sen* and *Haku* are case-less, (iv) only *Sen* and *dango* are case-less, (v) only *Haku* and *dango* are case-less, and (vi) all NPs are case-less.

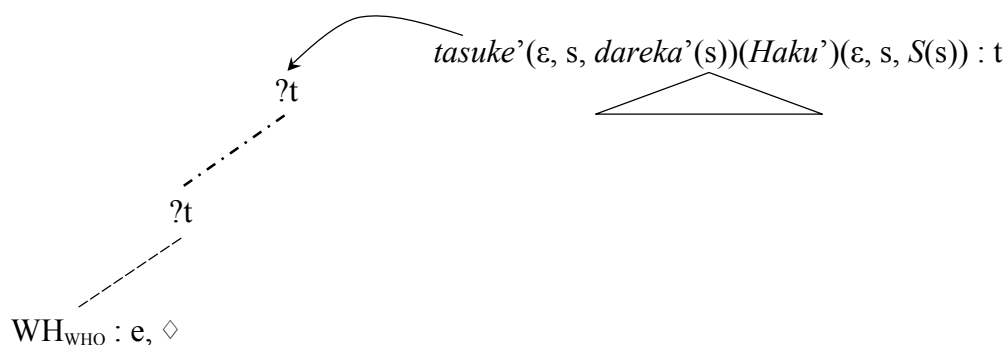
Stripping and clefts thus share the same case-marking pattern (including partial case-marking, which raises a problem for previous accounts). A further parallelism is revealed in the other ellipsis construction: sluicing.

6.3.2 Multiple Sluicing

According to Sect. 6.2.2, the *wh*-word in sluicing is processed by means of structural underspecification. To take as an example (5.115), repeated here as (5.131), the *wh*-word *dare* (= ‘who’) is parsed at an unfixed node induced by LOCAL *ADJUNCTION, as shown in (5.118), reproduced here as (5.132).

(5.131) Sluicing

Haku-wa mukashi kawa-de dareka-o tasuke-ta ga,
 H-TOP in.the.past river-in someone-ACC save-PAST but
 dare (da) ka mattaku omoidase-nai.
 who (COP) Q at.all remember-NEG
 ‘Haku saved someone in the river in the past, but he cannot remember who.’

(5.132) Parsing *Haku-wa mukashi kawa-de dareka-o tasuke-ta ga, dare*

As we saw in Sect. 6.2.2, the unfixed node of the *wh*-word in sluicing is resolved by (i) the parse of a case particle or (ii) the application of UNIFICATION as long as the copula *da-* provides a type-*t* meta-variable, licensing REGENERATION to project a propositional template. That is, the central mechanisms to analyse sluicing are the same as those used in clefts and stripping: (i) ‘structural underspecification,’ (ii) the copula *da-* as a type-*t* pro-form, and (iii) REGENERATION. Based on the Unique-unfixed-node Constraint, it is anticipated that the case-marking pattern in clefts and stripping repeats itself in sluicing.

First, multiple sluicing should be possible if each *wh*-word is case-marked, as evinced in (5.133).

(5.133) Sluicing

Haku-wa	mukashi	dokoka-de	dareka-o	tasuke-ta	ga,
H-TOP	in.the.past	somewhere-in	someone-ACC	save-PAST	but
doko-de	dare-o	(da) ka	mattaku	omoidase-nai.	
where-in	who-ACC	(COP) Q	at.all	remember-NEG	

Lit. 'Haku saved someone somewhere in the past, but he cannot remember who and where.'

LOCAL *ADJUNCTION induces an unfixed node for the initial *wh*-word *doko* (= 'where'). This is immediately **resolved** by the locative-case particle *de*. Since the tree no longer encompasses an unfixed node, LOCAL *ADJUNCTION may fire once again for the second *wh*-word *dare* (= 'who'). This unfixed node is immediately fixed by the case particle *o*.

Second, multiple sluicing should be ungrammatical if each *wh*-word lacks a case particle, as shown in (5.134).

(5.134) Sluicing

*Haku-wa	mukashi	dokoka-de	dareka-o	tasuke-ta	ga,
H-TOP	in.the.past	somewhere-in	someone-ACC	save-PAST	but
doko	dare	(da) ka	mattaku	omoidase-nai.	
where	who	(COP) Q	at.all	remember-NEG	

Int. 'Haku saved someone somewhere in the past, but he cannot remember who and where.'

The initial *wh*-word *doko* (= 'where') is scanned at an unfixed node induced by LOCAL *ADJUNCTION. This unfixed node, however, cannot be settled since (i) *doko* (= 'where') is case-less and (ii) the copula *da-* is not processed until the second *wh*-word *dare* (= 'who') is parsed. Thus, the parser has to induce another unfixed node **without** resolving the first unfixed node. This conflicts with the Unique-unfixed-node Constraint.

Third, the *wh*-words in (5.133) may be reversed as in (5.135), since the order makes no difference to the mechanism of resolving structural underspecification as long as each *wh*-word is marked by an appropriate case particle.

(5.135) Sluicing

Haku-wa mukashi dareka-o dokoka-de tasuke-ta ga,
 H-TOP in.the.past someone-ACC somewhere-in save-PAST but
 dare-o doko-de (da) ka mattaku omoidase-nai.
 who-ACC where-in (COP) Q at.all remember-NEG

Lit. 'Haku saved someone somewhere in the past, but he cannot remember who and where.'

In (5.135), the unfixed node for the initial *wh*-word *dare* (= 'who') is resolved by the accusative-case particle *o* before an unfixed node is introduced for the second *wh*-word *doko* (= 'where'). Thus, there is only a single unfixed node at a time during the course of incremental structure building.

Fourth, partially case-marked sluicing should be possible. That is, the **final** *wh*-word (and only the final *wh*-word) may be case-less since the unfixed node for the **final** *wh*-word is resolvable by UNIFICATION. Thus, compare (5.136)A with (5.136)B.²²

(5.136) Sluicing

Haku-wa mukashi dokoka-de dareka-o tasuke-ta ga,
 H-TOP in.the.past somewhere-in someone-ACC save-PAST but
 A: doko-de dare (da-tta) ka mattaku omoidase-nai.
 where-in who (COP-PAST) Q at.all remember-NEG
 B: *doko dare-o (da-tta) ka mattaku omoidase-nai.
 where who-ACC (COP-PAST) Q at.all remember-NEG

Int. 'Haku saved someone somewhere in the past, but he cannot remember who and where.'

Finally, multiple sluicing should be possible with more than two *wh*-words, provided that an unfixed node for each *wh*-word is resolved by a case particle, as in (5.137). In addition, the parentheses attached to the accusative-case particle *o* show that

²² Note that the copula *da-* in (5.136)A and (5.136)B is in a past tense form, as in *da-tta*. For unknown reasons, if the non-past form *da-* is used, the string (5.136)A is degraded to my ear. But even if the non-past form *da-* is used in both (5.136)A and (5.136)B, (5.136)A is much more acceptable than (5.136)B, which is fully unacceptable.

the case particle may be dropped off this **final** *wh*-word (and only this final *wh*-word). This is because UNIFICATION may apply to the **final** *wh*-word alone.

(5.137) Sluicing

Dareka-ga mukashi dokoka-de onnanoko-o tasuke-ta ga,
 someone-NOM in.the.past somewhere-in girl-ACC save-PAST but
 dare-ga doko-de dono-onnanoko-(o) (da-tta) ka omoidase-nai.
 who-NOM where-in which-girl(-ACC) (COP-PAST) Q remember-NEG
 Lit. ‘Someone saved a girl somewhere in the past, but I cannot remember who,
 where, and which girl.’

The usual comparison applies. That is, (5.137) is clearly contrasted with the other six possible variants, all of which are wholly unacceptable: namely, the examples where (i) only *dare* is case-less, (ii) only *doko* is case-less, (iii) only *dare* and *doko* are case-less, (iv) only *dare* and *dono-onnanoko* are case-less, (v) only *doko* and *dono-onnanoko* are case-less, and (vi) all *wh*-words are case-less.

We have discovered and explained the same grammaticality patterns between clefts, stripping, and sluicing. Most importantly, the partial case-marking facts in these three constructions have not been reported in the literature, and they pose a challenge to any previous accounts which posit different structures depending on whether focussed NPs are case-marked (e.g., Fukaya 2007, Hiraiwa & Ishihara 2012, Kizu 2005). In our dynamic account, these phenomena reflect the time-linear process of fixing structural underspecification in an incremental fashion.

6.4 Long-Distance Dependencies and Islands

In our system, stripping and sluicing are analysed on a par with clefts. In this subsection, we show that the account of long-distance clefts and island-involving clefts carries over to stripping and sluicing. First, long-distance stripping and sluicing are treated by the feeding relation of *ADJUNCTION and LOCAL *ADJUNCTION. Second, we

envisage the same island-sensitivity pattern of stripping and sluicing as seen with clefts. Fukaya (2003, 2007, 2012) has observed this expected pattern and stipulated completely distinct structures depending on whether the focussed NP is case-marked or not. In our dynamic account, this stipulation is dispensable.

6.4.1 Long-Distance Stripping

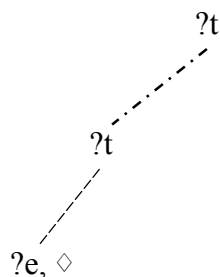
Consider long-distance stripping (5.138)B. Since the tree update is essentially the same as presented in Sect 5.2, a sketchy illustration would suffice.

(5.138) Long-distance stripping

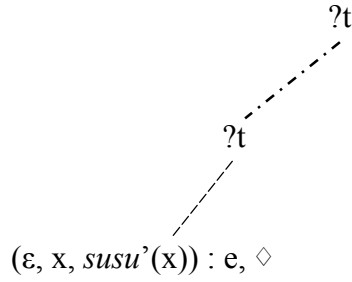
- A: [Kashira-ga hane-teiru to] Chihiro-ga i-tta nodayone?
 [K-NOM hop-PROG COMP] C-NOM say-PAST Q
 ‘Chihiro said that Kashira was hopping, right?’
- B: Iya, susuwatari da.
 no wandering-soot COP
 ‘No, Chihiro said that wandering-soots were hopping.’

As in the case of clefts, what matters is the **feeding** applications of *ADJUNCTION and LOCAL *ADJUNCTION. Thus, after the AXIOM sets out an initial type-t-requiring node, the parser performs *ADJUNCTION and LOCAL *ADJUNCTION in this order.

(5.139) *ADJUNCTION + LOCAL *ADJUNCTION

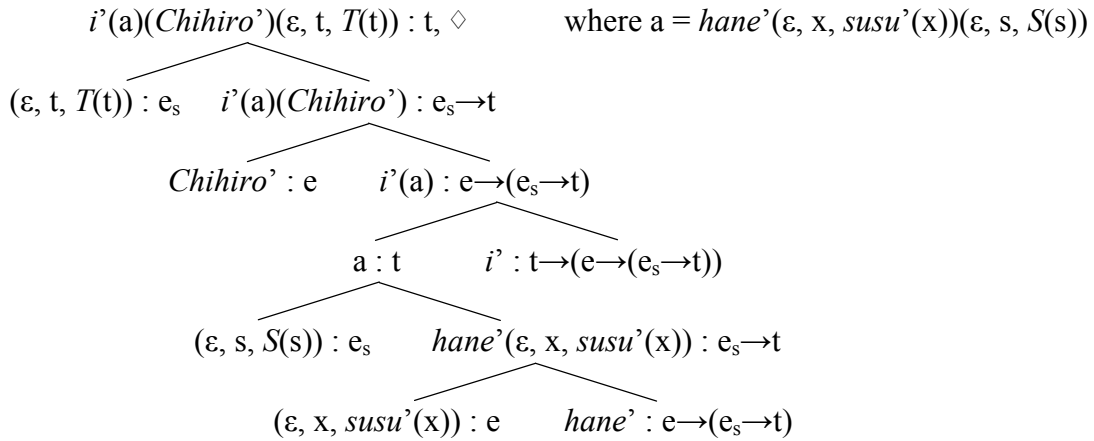


Ignoring the particle *iya* (= ‘no’), the focussed NP *susuwatari* (= ‘wandering-soot’) is scanned at the current type-e-requiring node.

(5.140) Parsing *Susuwatari*

The pointer $\diamond$ returning to the higher type- t -requiring node, the copula *da-* imposes a propositional meta-variable. This licenses the re-use of previous actions that built the structure of the antecedent string (5.138)A. In particular, *hane-* (= ‘hop’) introduces a subject node, with which the node of *susuwatari* is merged (i.e., UNIFICATION). Finally, ELIMINATION outputs the final state (5.141).

(5.141) ELIMINATION (five times)



Multiple long-distance stripping (5.142) is also predicted to be grammatical. This is because LOCAL *ADJUNCTION may introduce a type- e -requiring unfixed node for each focussed NP, *Chihiro* and *onigiri* (= ‘rice balls’), under the type- t -requiring node introduced by *ADJUNCTION.

(5.142) Multiple long-distance stripping

- A: [Lin-ga okashi-o tabe-ta to] Haku-ga i-tta nodayone?
 [L-NOM sweets-ACC eat-PAST COMP] H-NOM say-PAST Q
 ‘Haku said that Lin ate sweets, right?’
- B: Iya, Chihiro*(-ga) onigiri(-o) da.
 no C(-NOM) rice.ball(-ACC) COP
 ‘No, Haku said that Chihiro ate rice balls.’

Since the unfixed node of the **final** NP alone may be resolved by UNIFICATION, the **final** NP (in the present example, *onigiri*) may lack a case particle. This is designated by the parentheses attached to the accusative-case particle *o*. It is also predicted that any non-final NPs may not be case-less because (i) UNIFICATION is eventually licensed by the copula *da-* and (ii) *da-* is scanned only after the final NP. Thus, the nominative-case particle *ga* cannot be dropped, as shown by the asterisk.

6.4.2 Long-Distance Sluicing

Consider long-distance sluicing (5.143). This time, too, the illustration is concise since the tree update proceeds essentially the same as clefts and stripping.

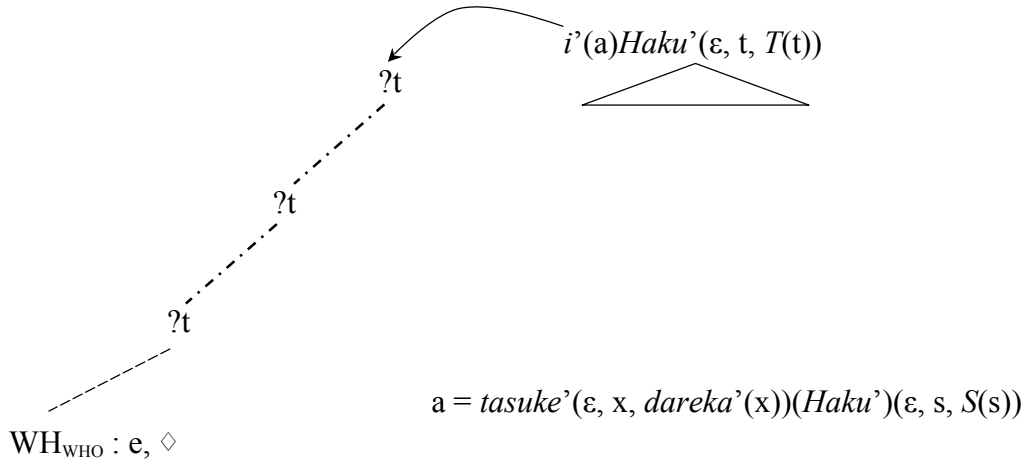
(5.143) Long-distance sluicing

- Haku-wa [mukashi kawa-de dareka-o tasuke-ta to]
 H-TOP [in.the.past river-in someone-ACC save-PAST COMP]
 i-tta ga, dare (da) ka mattaku wakara-nai.
 say-past but who (COP) Q at.all know-NEG
 ‘Haku said that he saved someone in the river in the past, but I do not know at all who he said that he saved.’

First of all, the AXIOM specifies an initial type-t-requiring node, and this is developed by the initial clause (glossing over the adjuncts). Once the propositional tree is built up, it is LINKed to a new type-t-requiring node by *ga* (= ‘but’). In (5.143), the ellipsis part *dare (da) ka* is **doubly** embedded: it is embedded under the clause *Haku-wa ... i-tta*,

and it itself is embedded under the clause *mattaku wakara-nai*. Thus, *ADJUNCTION is run **twice** and LOCAL *ADJUNCTION is then executed in this order. The *wh*-word *dare* (= ‘who’) is parsed at the type-e-requiring unfixed node induced by LOCAL *ADJUNCTION.

(5.144) Parsing *Haku-wa [mukashi kawa-de dareka-o tasuke-ta to] i-tta ga, dare*



The rest of the process is as usual. The pointer $\diamond$ moves up to the intermediate type-t-requiring node, and annotates the node with a propositional meta-variable. This then licenses the reiteration of the previous actions that built the tree of the antecedent part. In particular, the verb *tasuke-* (= ‘save’) creates an object node, with which the node of the *wh*-word *dare* is merged (i.e., UNIFICATION).

This analysis also handles multiple long-distance sluicing (5.145). For each *wh*-word *doko* (= ‘where’) and *dare* (= ‘who’), LOCAL *ADJUNCTION induces a type-e unfixed node with respect to the type-t-requiring node introduced by *ADJUNCTION. As signified by the parentheses and the asterisk, only the **final** *wh*-word *dare* may be caseless because UNIFICATION may apply only to the **final** *wh*-word.

(5.145) Multiple long-distance sluicing

Haku-wa [mukashi dokoka-de dareka-o tasuke-ta to]
 H-TOP [in.the.past somewhere-in someone-ACC save-PAST COMP]
 i-tta ga, doko*(-de) dare(-o) da ka mattaku wakara-nai.
 say-PAST but where(-in) who(-ACC) COP Q at.all know-NEG
 Lit. ‘Haku said that he saved someone somewhere in the past, but I do not
 know who and where at all.’

We thus present a uniform account of clefts, stripping, and sluicing in **both** short-distance and long-distance settings. This parallelism finds its way around the issue of island-sensitivity as well, to which we shall now turn.

6.4.3 Island-Involving Stripping

Fukaya (2003, 2007) points out that case-marked stripping is sensitive to islands while case-less stripping is not. Consider (5.146), which is constructed on the basis of Fukaya (2007: 159-60). The point is that the island-crossing reading is available only if the case particle *o* is absent from the focus *furansu-ryouri* (= ‘French cuisine’) in (5.146)B.

(5.146) Island-involving stripping

- A: [Itariya-ryouri-o tsukuru hito]-ga yoku kono-mise-ni kuru
 [Italian-cuisine-ACC make person]-NOM often this-shop-to come
 rashii yo.
 seem SFP
 ‘I hear that those who make Italian cuisine often come to this shop.’
- B: Furansu-ryouri(*-o)-mo da yo.
 French-cuisine(-ACC)-also COP SFP
 ‘French cuisine is also such that those who make it often come to this shop.’

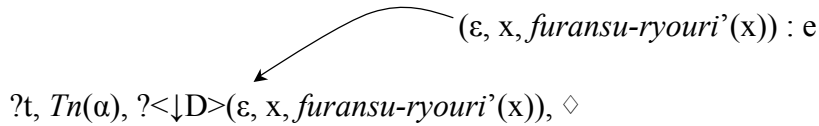
This island-sensitivity pattern is predicted by the LINK-based parsing of a focus item in clefts (Sect. 5.3). That is, after the AXIOM sets out an initial node, LINK INTRODUCTION initiates an inverse LINK relation to a new type-e-requiring node.

(5.147) LINK INTRODUCTION



After this node is decorated by the focus *furansu-ryouri* (= ‘French cuisine’), LINK ENRICHMENT returns the pointer $\diamond$ to the type-t-requiring node and annotates it with the requirement $\text{?<}\downarrow\text{D>}(\epsilon, x, \textit{furansu-ryouri}'(x))$. This states that $(\epsilon, x, \textit{furansu-ryouri}'(x))$ will be shared by the paired trees. Note that the operator <D> governs the **weakest** tree relation; thus, the requirement may be satisfied by a node inside an island, in DS terms, a node within a LINKed tree.

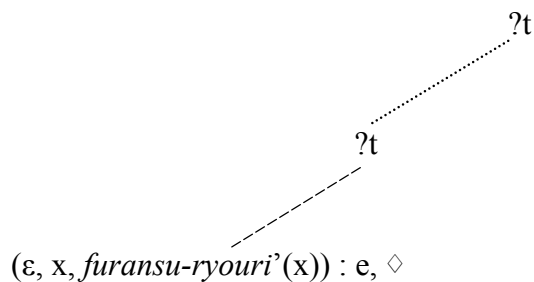
(5.148) LINK ENRICHMENT



Ignoring the additive particle *mo*, the rest of the tree update is as usual. The copula *da*-posits a meta-variable, licensing REGENERATION: the parser re-runs the previous actions that built the tree of the antecedent string. Of note is that the actions encoded in *tsukuru*- (= ‘make’) create an object node and that the meta-variable at this node is substituted with $(\epsilon, x, \textit{furansu-ryouri}'(x))$, satisfying the requirement. It thus follows that case-less stripping is not sensitive to islands. If the focussed item *furansu-ryouri* is case-marked, the LINK-based processing fails to reach a well-formed final state since the parse of a case particle at an inversely LINKed node leaves outstanding requirements (Sect. 5.3.2). Therefore, case-marked stripping is sensitive to islands.

One may wonder whether *furansu-ryouri* may be processed after GENERALISED ADJUNCTION and LOCAL *ADJUNCTION are applied in this order, as in (5.149). (Recall that GENERALISED ADJUNCTION induces an unfixed node that may be resolved across an island boundary, or in DS terms, across a LINK relation.)

(5.149) GENERALISED ADJUNCTION + LOCAL *ADJUNCTION + Parsing *furansu-ryouri*



But this tree-update does not work. In order to parse the copula *da-*, the pointer $\diamond$ must move to a type-t-requiring node. Suppose that the pointer $\diamond$ moves to the **lower** type-t-requiring node. If *da-* is parsed here, the node is decorated with the meta-variable U , the type t , and the label $\langle L^{-1} \rangle \perp$. The entry of *da-* (5.106) is repeated here as (5.150).

(5.150) Entry of the copula *da-* (final version)

IF	$?t$	
THEN	IF	$\langle \downarrow_+ \rangle \top$
	THEN	$\text{put}(U : t, \langle L^{-1} \rangle \perp)$
	ELSE	$\text{put}(U : t)$
ELSE	ABORT	

If this type-t meta-variable licenses REGENERATION (i.e., the re-run of the actions that constructed the structure of the antecedent part in (5.146)A), an inverse LINK relation is induced. But this contradicts with $\langle L^{-1} \rangle \perp$, which declares that there is no such inverse LINK relation. One may wonder whether the pointer $\diamond$ may move to the **higher** type-t-requiring node before *da-* is parsed, but the pointer movement across a globally unfixed relation is blocked by the constraint on pointer movement (see footnote 16).

Overall, the only successful tree update is the LINK-based processing of the focussed NP, as shown in (5.147)-(5.148). This analysis correctly predicts the island-sensitivity asymmetry between case-marked and case-less stripping.

6.4.4 Island-Involving Sluicing

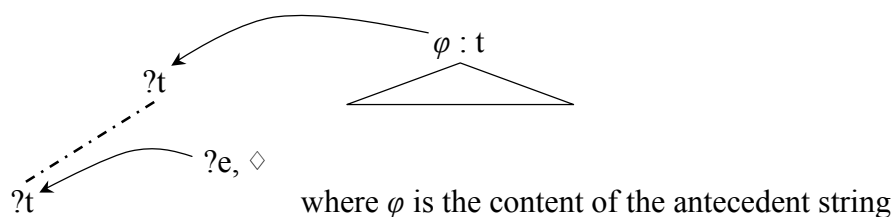
Fukaya (2003, 2007, 2012) points out that case-marked sluicing is sensitive to islands whereas case-less sluicing is not. In (5.151), the free translation, which is an island-crossing interpretation, is available only when the particle *ni* is dropped off the *wh*-phrase *hokano-dono-giin* (= ‘which other Representative’).

(5.151) Island-involving sluicing

Boku-wa [keisatsu-ga [Tanaka-giin-ni wairo-o oku-tta otoko]-o
 I-TOP [police-NOM [T-rep.-to bribe-ACC give-PAST man]-ACC
 taihoshita no]-wa shitteru ga, [hokano-dono-giin(*-ni)ka]-wa shira-nai.
 arrested NO]-TOP know but [other-which-rep.(-to) Q]-TOP know-NEG
 ‘I know that the police arrested the man who gave a bribe to Representative
 Tanaka, but I do not know which other Representative is such that the police
 arrested the man who gave a bribe to him.’ (adapted from Fukaya (2007: 110))

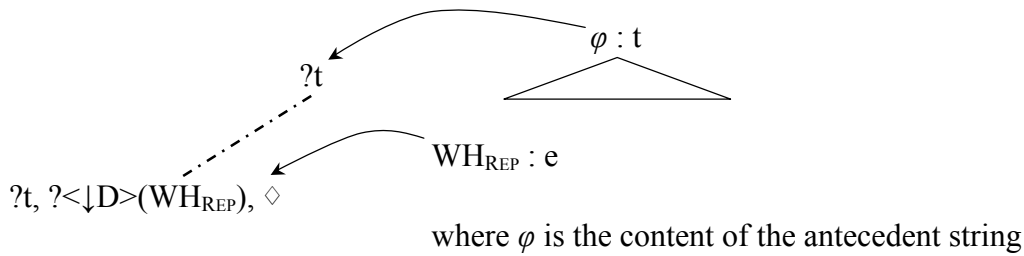
A possible option is the feeding applications of GENERALISED ADJUNCTION and LOCAL *ADJUNCTION, but this tree update does not reach a well-formed tree state (see the discussion around (5.149)). Still, there is another option available: the LINK-based processing of the focussed *wh*-word. First, the antecedent part generates a propositional tree and it is LINKed to a type-*t*-requiring node by the connective *ga*. Second, since the ellipsis part is subordinate to the embedding clause *shira-nai*, *ADJUNCTION introduces a type-*t*-requiring unfixed node. Now, LINK INTRODUCTION induces an inverse LINK relation to a new type-*e*-requiring node.

(5.152) LINK INTRODUCTION



The current node is then decorated with WH_{REP} , the content of the *wh*-word *hokano-dono-giin* (= ‘which other Representative’). LINK ENRICHMENT returns the pointer $\diamond$ to the LINKed type-*t*-requiring node and annotates the node with $?<\downarrow D>(WH_{REP})$, stating that a node **somewhere below** will be decorated with WH_{REP} . So, the decoration WH_{REP} may appear at a node inside the island (i.e., a node within a LINKed tree).

(5.153) LINK ENRICHMENT



The rest of the process is as usual. The copula *da-* eventually licenses the repetition of the previous actions that constructed the antecedent structure (i.e., REGENERATION). In particular, the lexical actions encoded in *oku-* (= ‘send’) create an indirect object node. The meta-variable at this node is substituted with WH_{WHO} , which meets the requirement. This captures island-insensitivity of case-less sluicing. In addition, this LINK-based processing of the *wh*-word does not lead to a well-formed state if the *wh*-word is case-marked because outstanding requirements will remain in the tree (Sect. 5.3.2). From this analysis, island-sensitivity of case-marked sluicing falls out.

The upshot is that our account of long-distance clefts and island-involving clefts is extendable to stripping and sluicing. Most importantly, the same island-contrast is detected in these ellipsis constructions. These data have been noted in Fukaya (2003, 2007, 2012), but his analysis ends up postulating radically distinct structures depending on whether the focussed item is case-marked or not. By contrast, the dynamic analysis uniformly characterises island-(in)sensitivity of clefts, stripping, and sluicing, whether case-marking is involved or not.

6.5 Summary

Our analysis of clefts is uniformly applicable to stripping and sluicing, ‘uniformly’ in that (i) all of these constructions are analysed by the same machinery and that (ii) the analysis is neutral to the case-marking of the focus. The analysis is not only uniform but also accounts for the discrepancies regarding multiple foci and islands. In particular, we have explained the novel data of partial case-marking in stripping and sluicing. These data invalidate any previous studies of stripping and sluicing that stipulate different structures depending on the case-marking of the focus (Fukaya 2003, 2007, 2012; see also Takahashi 1994).

7 Conclusion

Japanese clefts vividly manifest the flow of language understanding: the pre-*no-wa* part (i.e., the presupposition clause) constructs a propositional tree with respect to which the focus item is subsequently parsed within a new tree. This tree transition is triggered by the unit *no-wa*, whose encoding is derived by combining the entries of the nominaliser *no* (Sect. 3.2, Chap. 4) and the topic particle *wa* (Sect. 5.3, Chap. 3).

This dynamic account does **not** structurally distinguish between the two types of clefts. Clefts_{+C} and clefts_{-C} are mapped onto **the same** structure and the presence of a case particle simply affects the way the content of a focus is reflected into the main tree. Firstly, the focus may be parsed by dint of **structural underspecification**. In this case, if the focus has a case particle, the unfixed node is resolved lexically; otherwise, it is resolved by the general action UNIFICATION. Secondly, the focus may be parsed in terms of **LINK transitions**. In this case, if the focus has a case particle, the tree update crashes; otherwise, it may reach a well-formed state.

This uniform treatment of clefts is preferable over previous accounts by the corpus indication that clefts should not be structurally distinguished depending on

whether the focus has a case particle. The analysis is also endorsed by experimental data that the parser forms a dependency between the gap and the upcoming focus by the time *no-wa* is scanned (Kahraman 2011, Kahraman et al. 2011a). Furthermore, an array of corpus data in Chap. 2 is accounted for, the topic and focus effects are modelled as a by-product of progressive tree growth, and long-distance clefts are handled incrementally and monotonically.

The dynamic account integrates the two types of clefts but still succeeds in explaining away the two sets of contrasts: multiple foci and island-sensitivity. First, the issue of multiple foci. The Unique-unfixed-node Constraint, a consequence of LOFT, explains, among other things, (i) why the two types of clefts exhibit a contrast regarding multiple foci and (ii) why only the final focus may lack a case particle. This is a clear advantage of our analysis since these data are problematic for previous works (Sect. 2).

Second, the issue of island-sensitivity. As stated above, there are two ways the focus may be processed. First, structural underspecification may be induced by LOCAL *ADJUNCTION. This type of tree update fails to model island-involving clefts altogether, since the unfixed node induced must be settled **locally**. Second, the focus may be parsed in terms of LINK transitions and a requirement is posited that the main tree will contain a node decorated with the content of the focus **somewhere below**. This requirement may be met by a node inside an island (i.e., a node within a LINKed tree). This second type of tree update leads to a well-formed state only when the focus is case-less. This predicts that clefts_{+C} are sensitive to islands while clefts_{-C} are not.

The dynamic account of clefts is vindicated by its striking applicability to stripping and sluicing. These ellipsis constructions, whether case-marked or not, are modelled uniformly on the same grounds as clefts. Further, the same case-marking pattern is observed in clefts, stripping, and sluicing, and they are all explained in terms of how structural underspecification may be resolved during the course of incremental parsing. Moreover, the island-asymmetry in clefts carries over to stripping and sluicing,

and they are all accounted for in terms of whether the LINK-based processing of the focus converges or not.

In this thesis, we have so far articulated an account of *no*-nominalisation and cleft constructions. The thrust of the analysis, language dynamics, will be further confirmed by looking at its extension to a broader set of *no*-involving constructions in the next chapter: free relatives, head-internal relatives, and change relatives.

Chapter 6

The Dynamics of Relative Constructions

1 Overview

We have scrutinised *no*-nominalisation (Chap. 4) and cleft constructions (Chap. 5). As we shall see, the accounts given in these chapters extend naturally to the various types of *no*-involving relative clause constructions.

The familiar type of relative in Japanese is the head-external relative as in (6.1), as discussed in Chap. 3 (Sect. 5.2).

(6.1) Head-External Relatives (HERs)

Kiki-wa [tsukue-no-ue-ni oite-atta ringo]-o tabeta.

K-TOP [table-GEN-top-at place-existed apple]-ACC ate

‘Kiki ate an apple that was on a table.’

In addition to the familiar type (6.1), Japanese displays other kinds of relative clause constructions: change relatives (Tonosaki 1998), free relatives (Hoshi 1995), gapless relatives (Inoue 1976), *half*-relatives (Ishii 1990), head-internal relatives (Kuroda 1992), *no/de*-introduced relatives (Hiraiwa 2012, Kuroda 1992), *tokoro*-relatives (Harada 1973, Kuroda 1999), and so on. This chapter concerns free relatives (6.2), head-internal relatives (6.3), and change relatives (6.4), all of which involve the particle *no*. As we shall see, these three types of relatives are accounted for straightforwardly by the entry

of the nominaliser *no* (see Chap. 4) or the combined entry of the nominaliser *no* and a case particle.

(6.2) Free Relatives (FRs)

Kiki-wa [tsukue-no-ue-ni oite-atta no]-o tabeta.
 K-TOP [table-GEN-top-at place-existed NO]-ACC ate
 ‘Kiki ate what was on a table.’

(6.3) Head-Internal Relatives (HIRs)

Kiki-wa [ringo-ga tsukue-no-ue-ni oite-atta no]-o tabeta.
 K-TOP [apple-NOM table-GEN-top-at place-existed NO]-ACC ate
 ‘An apple was on a table and Kiki ate it.’

(6.4) Change Relatives (CRs)

[Otamajakushi-ga kaeru-ni natta no]-ga niwa-o haneteiru.
 [tadpole-NOM frog-COP became NO]-NOM garden-in is.hopping
 ‘A frog which is the result of changing from a tadpole is hopping in the garden.’ (Tonosaki 1998: 144)

‘Free relatives’ is another name for ‘participant nominalisation’ (Chap. 4). Thus, we already have a Dynamic Syntax (DS) account of FRs: the nominaliser *no* maps the proposition of the pre-*no* clause onto a term that denotes an individual reflecting the proposition (Sect. 3, Chap. 4). The objectives of this chapter are twofold: (i) to uncover further properties of FRs and (ii) to develop an analysis of HIRs and CRs by coalescing the analysis of *no*-nominalisation with that of clefts (Seraku in press b).

Sects. 2-4 are devoted to HIRs: Sect. 2 introduces foundational data, Sect. 3 surveys previous studies, and Sect. 4 proposes a DS account. HIRs look quite similar to FRs, but HIRs display **both** nominal and non-nominal properties and cannot be handled by the nominaliser *no*. I shall amalgamate the analysis of *no*-nominalisation (Chap. 4) with that of clefts (Chap. 5), and contend that HIRs are a construction in which the HIR clause and the case particle **jointly** determine a rich structure to be developed by the

immediately embedding clause. More specifically, (i) the particle *no* and a case particle form a unit and (ii) this unit maps the propositional tree of the HIR clause onto another propositional tree that has been partially articulated when it is induced. Sect. 5 shifts our attention to CRs. It will be pointed out that like FRs but unlike HIRs, CRs have **only** nominal properties (Tonosaki 1996, 1998). This motivates a uniform account of FRs and CRs in virtue of the nominaliser *no*. The overall picture is portrayed in Figure 6.1. Finally, Sect. 6 concludes the chapter.

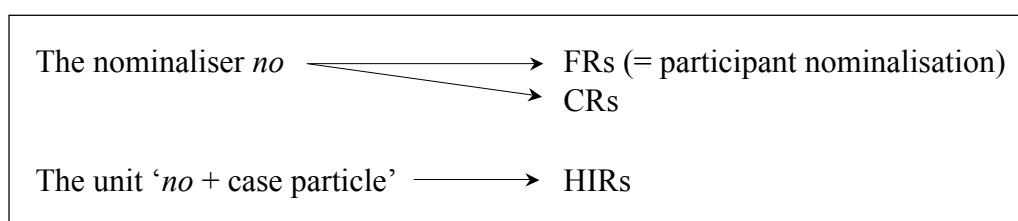


Figure 6.1. Form-function correspondences

2 Properties of HIRs

2.1 Introduction

Dryer (2011) reports that there are at least 63 languages that display Head-Internal Relatives (HIRs), Japanese being one such language. As mentioned at the outset, HIRs are clearly contrasted with Head-External Relatives (HERs).

(6.5) Head-External Relatives (HERs)

Kiki-wa [[e_i tsukue-no-ue-ni oite-atta] ringo_i]-o tabeta.
 K-TOP [[table-GEN-top-at place-existed] apple]-ACC ate
 'Kiki ate an apple that was on a table.'

(6.6) Head-Internal Relatives (HIRs)

Kiki-wa [[ringo-ga tsukue-no-ue-ni oite-atta] no]-o tabeta.
 K-TOP [[apple-NOM table-GEN-top-at place-existed] NO]-ACC ate
 'An apple was on a table and Kiki ate it.'

Descriptively, there are three surface differences. First, in the HER (6.5), the head *ringo* (= ‘apple’) resides outside of the relative clause *tsukue-no-ue-ni oite-atta*; in the HIR (6.6), the head *ringo* is internal to the relative clause. Second, in the HER (6.5), the relative clause has a gap, as indicated by the symbol e_i ; in the HIR (6.6), no gap is found in the relative clause since the position that would be occupied by a gap is actually filled with the internal head. Third, unlike the HER (6.5), the relative clause in the HIR (6.6) ends with the particle *no*. A fundamental question, then, is how the HIR head, though it is **internal** to the relative clause, is licensed by the embedding verb.

In addition to these surface differences, HIRs possess a number of unique characteristics. Following previous studies (and drawing on some new data), this section reveals the indeterminacy of HIRs (Sect. 2.2), the quantificational nature of HIRs (Sect. 2.3), island-insensitivity of HIRs (Sect. 2.4), and the nominal/non-nominal ambivalence of HIRs (Sect. 2.5). Finally, these properties are summarised as desiderata for a satisfactory account of HIRs (Sect. 2.6).

2.2 Indeterminacies

As noted in Kuroda (1976-77, 1992), HIRs are intrinsically indeterminate. There are at least two respects in which these indeterminacies are observed.

The first indeterminacy concerns the **head**. When the HIR clause comprises more than one NP, the head may be identified with any of them as long as a relevant context is provided and other constraints (e.g., the Relevancy Condition; see below) are also met (Kuroda 1992: 153-5). Consider (6.7), where the relative clause contains two NPs: *Osono* and *Jiji*.¹ As shown in (6.7)a-(6.7)b, either of these NPs could be a head. Even a ‘split head’ (6.7)c is possible, where the head is the group of the individuals denoted by these NPs.

¹ Shimoyama (1999: Sect. 5) claims that there seems to be an indefiniteness restriction on the head of HIRs. Kitagawa (2005: 1262-3), however, argues that the restriction is reducible to the Relevancy Condition (see below in the text). A similar remark is found in Hoshi (1995: 103-7).

(6.7) Head-Internal Relatives (HIRs)

Kiki-wa [Osono-ga Jiji-o oikaketeita no]-o tsukamaeta.

K-TOP [O-NOM J-ACC was.chasing NO]-ACC caught

- a. ‘Osono was chasing Jiji and Kiki caught her (= Osono).’
- b. ‘Osono was chasing Jiji and Kiki caught him (= Jiji).’
- c. ‘Osono was chasing Jiji and Kiki caught them (= Osono and Jiji).’

The second indeterminacy pertains to the **relation** between the HIR and the embedding clauses (Kim 2008b: 12-3). In the examples so far considered, HIRs are translated by using clausal coordination, where the relation between the HIR and the embedding clauses is not explicated. But HIRs may be translated by exploiting more specific clausal connectives. For instance, the relation in (6.8) is that of simultaneity, as shown by the use of *while*; the relation in (6.9) is that of concession, as shown by the use of *although*.

(6.8) Head-Internal Relatives (HIRs)

Kiki-wa [Jiji-ga hashi-tteiru no]-o yobitomete.

K-TOP [J-NOM running-PROG NO]-ACC called.to.stop

‘Kiki called Jiji to stop **while** he was running.’

(6.9) Head-Internal Relatives (HIRs)

[Kiki-ga Tombo-ni keiki-o tsukutte-oita no]-o

[K-NOM T-for cake-ACC make-placed NO]-ACC

Jiji-ga tabeteshimatta.

J-NOM has.eaten

‘**Although** Kiki made cake for Tombo, Jiji has eaten it.’

But the relation between the HIR and the embedding clauses is **not unconstrained**. In fact, it is subject to a tight constraint, the ‘Relevancy Condition’ (Kuroda 1975-76, 1992: 147): for an HIR clause to be felicitous, ‘it is necessary that it be interpreted pragmatically in such a way as to be directly relevant to the pragmatic content of its matrix clause.’ Later works such as Kim (2008a) and Grosu (2010) have shed light on

the notion of ‘relevant.’ Though the details are still controversial, there are two main ingredients: temporal and locational contiguity. That is to say, the event described by the HIR clause must intersect temporally and spatially with the event described by the embedding clause. For instance, (6.8) cannot have the reading: ‘Jiji is now running, and Kiki called Jiji to stop in the past when Jiji was, say, walking.’ By contrast, this reading is possible in the HER analogue (6.10); it is accessible when the adverbial nouns *ima* (= ‘now’) and *mukashi* (= ‘in the past’) are present.

(6.10) Head-External Relatives (HERs)

Kiki-wa [(ima) hashitteiru Jiji]-o (mukashi) yobitomete.

K-TOP [(now) be.running J]-ACC (in.the.past) called.to.stop

- a. ‘Kiki called Jiji, who was running, to stop.’
- b. ‘Kiki has called Jiji, who is now running, to stop in the past.’

To sum up, HIRs are inherently indeterminate. The identification of the head requires pragmatic inference; if an appropriate context is provided, even a ‘split head’ is possible. The relation between the HIR and the embedding clauses is also context-dependent (i.e., denoting a wide array of relations) modulo the ‘Relevancy Condition,’ which requires temporal, locational, and other kinds (e.g., ‘modal’ (Grosu 2010: 258)) of coherence between the events denoted by the HIR and the embedding clauses.

2.3 Quantification

Shimoyama (1999, 2001) observes some interpretive differences between HERs and HIRs with regard to quantification. Consider the HER (6.11) and the HIR (6.12).

(6.11) Head-External Relatives (HERs)

Kiki-wa [tsukue-no-ue-ni oite-atta hotondo-no ringo]-o tabeta.

K-TOP [table-GEN-top-at place-existed most-GEN apple]-ACC ate

‘Kiki ate most apples that were on a table.’

(6.12) Head-Internal Relatives (HIRs)

Kiki-wa [hotondo-no ringo-ga tsukue-no-ue-ni oite-atta
 K-TOP [most-GEN apple-NOM table-GEN-top-at place-existed
 no]-o tabeta.
 NO]-ACC ate
 ‘Most apples were on a table and Kiki ate them.’

In the HER (6.11), the domain of the quantifier *hotondo* (= ‘most’) is restricted by **both** the head *ringo* (= ‘apple’) and the relative clause *tsukue-no-ue-ni oite-atta*. The nuclear scope is the matrix clause *Kiki-wa ... tabeta*. Informally, the interpretation is: ‘for most x such that x was an apple and x was on a table, Kiki ate x.’ In this construal, (6.11) is felicitous in a situation where there were 10 apples on a table and Kiki ate 8 of them. That is, 8 apples eaten by Kiki count as ‘most’ with respect to the apples on the table. This situation, however, cannot be appropriately described by the HIR (6.12). Rather, (6.12) is suited for a situation where someone bought 10 apples and placed 8 apples on a table, and Kiki ate **all** of these 8 apples. That is, 8 apples count as ‘most’ with respect to the set of contextually relevant apples. In other words, the domain of *hotondo* is restricted **only** by the head *ringo* and the nuclear scope is the rest of the relative clause *tsukue-no-ue-ni oite-atta*.

Quantification in HIRs is also illuminated by scope relations (Shimoyama 1999, 2001). In Japanese, the scope of some quantifiers such as ‘every’ reflects the linear order in which they are presented (Kuroda 1970). Thus, the quantified head *dono-hon-mo* (= ‘every book’) in the HER (6.13) out-scopes *hotondo-no-gakusei* (= ‘most students’). In the HIR (6.14), by contrast, the opposite relation holds: the quantified head *dono-hon-mo* (= ‘every book’) takes narrow scope below *hotondo-no-gakusei* (= ‘most students’). This indicates that quantifiers within the HIR clause do **not** interact with quantifiers within the embedding clause (see Sect. 4.3.2 for details).

(6.13) Head-External Relatives (HERs)

[Tsukue-no-ue-ni oite-atta dono-hon]-mo
 [table-GEN-top-at place-existed every-book]-too
 hotondo-no-gakusei-ga yonda.
 most-GEN-student-NOM read
 ‘Every book that was on a table, most students read it.’
 (i) ‘every’ > ‘most’ (ii) *‘most’ > ‘every’

(6.14) Head-Internal Relatives (HIRs)

[Dono-hon-mo tsukue-no-ue-ni oite-atta no]-o
 [every-book-too table-GEN-top-at place-existed NO]-ACC
 hotondo-no-gakusei-ga yonda.
 most-GEN-student-NOM read
 ‘Every book was on a table, and most students read it.’
 (i) *‘every’ > ‘most’ (ii) ‘most’ > ‘every’

The above considerations unveil another unique interpretive aspect of HIRs: ‘maximality.’ Consider (6.12), reproduced here as (6.15). Recall that the appropriate situation for this HIR is where someone bought 10 apples and placed 8 apples on a table and Kiki ate **all** of these 8 apples. That is, the internal head seems to denote the maximal set of entities at issue. This effect, called the ‘maximality’ effect (Sells 1986), has been a hallmark of Japanese HIRs (Grosu 2010, Hoshi 1995, Kim 2007, Shimoyama 1999).

(6.15) Head-Internal Relatives (HIRs)

Kiki-wa [hotondo-no ringo-ga tsukue-no-ue-ni oite-atta
 K-TOP [most-GEN apple-NOM table-GEN-top-at place-existed
 no]-o tabeta.
 NO]-ACC ate
 ‘Most apples were on a table and Kiki ate them.’

On closer inspection, however, the maximality effect is not always present in HIRs, hence not a defining property of HIRs. Consider (6.16), where *san-ko-dake* (= ‘only 3’)

occurs without anomalousness. This indicates that the maximality effect is optional and that it is inferred pragmatically.

(6.16) Head-Internal Relatives (HIRs)

Kiki-wa [hotondo-no ringo-ga tsukue-no-ue-ni oite-atta
 K-TOP [most-GEN apple-NOM table-GEN-top-at place-existed
 no]-o san-ko-dake tabeta.
 NO]-ACC 3-CL-only ate
 ‘Most apples were on a table and Kiki ate only 3 of them.’

This type of example is also acknowledged in Shimoyama (1999: 177, endnote 7) and Shimoyama (2001: 97, footnote 8), with a passing remark that maximality ‘can be cancelled.’ This remark also seems to indicate that maximality effects are a result of pragmatic inference and not to be encoded linguistically (Grice 1975). This conclusion is confirmed by (6.17). Kubota & Smith (2007) point out that for the HIR (6.17) to be felicitous, a situation must be where each passenger puts no more than one ticket in the ticket checker even though she/he has multiple tickets, given our world knowledge that the insertion of multiple tickets may cause malfunction of the checker.

(6.17) Head-Internal Relatives (HIRs)

Dono-joukyaku-mo [saifu-ni kaisuiken-ga haitteita no]-o
 every-passenger-too [wallet-in coupon.ticket-NOM was.present NO]-ACC
 toridashite kaisatsu-ni ireta.
 take.out ticket.checker-to put
 ‘Every passenger took out a coupon ticket that she/he had in (her/his) wallet and put it in the ticket checker.’ (adapted from Kubota & Smith (2007: 154))

An adequate account of HIRs must model the context-dependency of the maximality effect. In Sect. 3, some recent analyses of HIRs will be criticised in that they build maximality into the grammar, failing to capture its optionality.

Of relevance to our current discussion is the issue of negation. It has been widely held that HIR clauses cannot license negation (Grosu & Landman 2012: 175-7, Hoshi 1995: 134-5). The following example is adapted from Hoshi (1995: 134).

(6.18) Head-Internal Relatives (HIRs)

*John-wa [Mary-ga teiburu-no-ue-ni ringo-o oitekurenakatta no]-o
 J-TOP [M-NOM table-GEN-top-on apple-ACC did.not.put NO]-ACC
 totte tabeta.
 picked.up ate
 ‘Mary put no apples on the table, and John picked them up and ate them.’

Contrary to the widely entertained belief, negation is **licensed** if the existence of an entity denoted by the head is inferrable. For instance, negation may occur within the HIR (6.19), where it is inferrable that there was a wallet somewhere other than a safe. In this reading, the quantified head *saifu* (= ‘a wallet’) out-scopes the negator. These negation data pose a problem for recent analyses of HIRs (see Sect. 3).²

(6.19) Head-Internal Relatives (HIRs)

Dorobou-wa [saifu-ga kinko-ni haittei-naka-tta no]-o
 thief-TOP [wallet-NOM safe-at put.inside-NEG-PAST NO]-ACC
 manmato nusumi-dashita.
 successfully steal-took.away
 ‘A wallet was not inside a safe (but outside the safe), and a thief successfully stole it.’

In brief, HIRs are radically different from HERs in terms of quantification (Shimoyama 1999, 2001). Contrary to the standard view, however, the maximality

² Even the HIR (6.18) improves if the topic marker *wa* is attached to *teiburu-no-ue-ni* (= ‘on the table’) and the predicate *okanakatta* (= ‘did not put’) (instead of *oitekurenakatta*) is used. When there are multiple occurrences of *wa*, the non-initial occurrence of *wa* induces a contrastive topic (Kuno 1973a: 48). This has an implication that apples were put somewhere else other than the top of the table.

effect is optional (hence, not to be linguistically encoded), and negation may be licensed in HIR clauses (hence, not to be linguistically precluded).

2.4 Islands

It was initially supposed that HIRs were sensitive to syntactic islands (e.g., Ishii 1989, Watanabe 1992). A number of subsequent studies (e.g., Hoshi 1995, Kuroda 2005, Mihara 1994, Mihara & Hiraiwa 2006), however, have claimed that HIRs are indeed not sensitive to islands. They point out that the unacceptable examples which seemed to indicate island-sensitivity of HIRs are degraded for other reasons than island constraints (e.g., the Relevancy Condition). For instance, Mihara (1994: 239-40, my translation) shows that HIRs are not sensitive to the Complex NP Island. (These examples are based on data in Watanabe (1992: 260-1).)

(6.20) Head-Internal Relatives (HIRs) involving a complex NP

[Taro-ga [Hanako-ga subarashii ronbun-o kaita toiu houkoku]-o
[T-NOM [H-NOM excellent paper-ACC wrote TOIU report]-ACC
kiiteita no]-ga tsuini shuppansareta.
has.heard NO]-NOM finally was.published

‘Taro has heard a report that Hanako wrote an excellent paper, and the paper was finally published.’

(6.21) Head-Internal Relatives (HIRs) involving a complex NP

[Kyouju-ga [sono-daigakuinsei-ga kaita ronbun]-o hometeita no]-ga
[professor-NOM [that-grad.student-NOM wrote paper]-ACC praised NO]-NOM
kondo, joshu-de saiyou-sareru kotoninatta.
this.time assistant-as hire-PASS has.turned.out

‘A professor praised a paper which the graduate student wrote, and it has turned out that the graduate student is hired as an assistant this time.’

Kuroda (2005: 215-20) assumes that HIRs are generally not sensitive to islands, but he states that the Complex NP Constraint may be an exception since the string containing

the Complex NP is less acceptable than the string containing other types of islands. For instance, consider the case of *wh*-islands (Kuroda 2005: 216, my translation). (This example is based on the data in Watanabe (1992: 261).)

(6.22) Head-Internal Relatives (HIRs) involving a *wh*-island

[John-ga Tom-ni [Mary-ga itsu ronbun-o shiageru ka]

[J-NOM Tom-DAT [M-NOM when paper-ACC finish Q]

kiiteita no]-ga shuppansareta.

had.asked NO]-NOM was.published

‘John had asked Tom when Mary finished a paper and it was published.’

According to Kuroda (2005), (6.22) is better than (6.20)-(6.21). But, at the same time, Kuroda states that HIRs involving a Complex NP such as (6.20) and (6.21) are not fully degraded. Given this remark, I shall assume in this thesis that HIRs are **not** sensitive to island constraints in general.

2.5 Ambivalence

Chap. 4 defined *no* as a **nominaliser** which maps a proposition onto a type-e term that denotes an entity reflecting the proposition. This entry of *no* integrates participant and situation nominalisation. As stated in Sect. 1, participant nominalisation is an instance of FRs. In the literature, FRs are generally considered to possess **nominal** status (Tonosaki 1996, 1998). This meshes well with the entry of the nominaliser *no*. In this sub-section, I shall argue that (i) HIRs have **both** nominal and non-nominal features and that (ii) the nominaliser *no* is thus unable to reflect HIRs in general.

2.5.1 Non-Nominality of HIRs

The free translations in our HIR examples use clausal coordination. This is the usual way of translating HIRs into English (Kim 2008b), reflecting the native speaker’s intuition that the HIR clause establishes a context for the embedding clause to be

processed. This is not true of FRs; the *no*-part directly serves as an argument of the embedding verb. This dissonance coincides with the non-nominality of HIRs and the nominality of FRs. This claim is further backed up by additional sets of data below.

When the nominaliser *no* denotes a human, it has an expressive connotation (Sect. 3.4.2, Chap. 4). For instance, *no* in the FR (6.23) has such a connotation, whereas *no* in the HIR (6.24) does not (Kitagawa 2005; see also Kuroda 1992). This suggests that individuals are not denoted in HIRs.

(6.23) Free Relatives (FRs)

Kiki-wa [naita no]-o nagusameta.

K-TOP [cried NO]-ACC consoled

‘Kiki consoled a person who cried.’

(6.24) Head-Internal Relatives (HIRs)

Kiki-wa [tomodachi-ga naita no]-o nagusameta.

K-TOP [friend-NOM cried NO]-ACC consoled

‘A friend cried and Kiki consoled her/him.’

Another contrast is that the *no*-part may be modified by demonstratives in FRs but not HIRs (Tonosaki 1996, 1998). Provided that only individual-denoting nominals are modifiable by demonstratives, it seems that an individual is denoted in the FR (6.25) but not in the HIR (6.26).

(6.25) Free Relatives (FRs)

Jiji-wa sono [Kiki-ga katteoita no]-o tabeteshimatta.

J-TOP that [K-NOM had.bought NO]-ACC has.eaten

‘Jiji has eaten that thing which Kiki had bought.’

(6.26) Head-Internal Relatives (HIRs)

*Jiji-wa sono [Kiki-ga pan-o katteoita no]-o tabeteshimatta.

J-TOP that [K-NOM bread-ACC had.bought NO]-ACC has.eaten

Int. ‘Kiki had bought that bread and Jiji has eaten it.’

The *no*-part may be modified by another relative clause in the FR (6.27), but not in the HIR (6.28). Put differently, relative clause stacking is possible only in FRs.

(6.27) Free Relatives (FRs)

Jiji-wa [Kiki-ga tanoshiminishiteita, [Osono-ga katte-kita no]]-o
 J-TOP [K-NOM looked.forward.to [O-NOM buy.came NO]]-ACC
 tabeteshimatta.
 has.eaten
 ‘Jiji has eaten that thing which Osono bought, which Kiki looked forward to.’

(6.28) Head-Internal Relatives (HIRs)

*Jiji-wa [Kiki-ga tanoshiminishiteita, [Osono-ga keiki-o katte-kita
 J-TOP [K-NOM looked.forward.to [O-NOM cake-ACC buy.came
 no]]-o tabeteshimatta.
 NO]]-ACC has.eaten
 Int. ‘Osono bought cake which Kiki looked forward to, and Jiji has eaten it.’

FRs but not HIRs may be used to provide an answer to *wh*-questions that ask about an individual (Matsuda 2002: 635-6, Ohara 1994: 262-3). For example, the *wh*-question *Who did Kiki console?* may be answered felicitously by the FR (6.29) but not by the HIR (6.30). This also suggests another nominal/non-nominal asymmetry between FRs and HIRs.

(6.29) Free Relatives (FRs)

Kiki-wa [naita no]-o nagusameta.
 K-TOP [cried NO]-ACC consoled
 ‘Kiki consoled a person who cried.’

(6.30) Head-Internal Relatives (HIRs)

#Kiki-wa [tomodachi-ga naita no]-o nagusameta.
 K-TOP [friend-NOM cried NO]-ACC consoled
 Int. ‘Kiki consoled a friend who cried.’ (unacceptable as an answer to ‘Who did Kiki console?’)

FRs and HIRs differ in terms of clefting too. As argued in Chap. 5, a focus position (i.e., a pre-copula position) in clefts is occupied only by type-e elements. Crucially, this position is inhabited by FRs, but not HIRs.

(6.31) Free Relatives (FRs) in clefts

[Kiki-ga todoketa no]-wa [Osono-ga yaita no] da.
 [K-NOM delivered NO]-TOP [Osono-NOM baked NO] COP
 ‘It is Osono’s-baked things that Kiki delivered.’

(6.32) Head-Internal Relatives (HIRs) in clefts

*[Kiki-ga todoketa no]-wa [Osono-ga pan-o yaita no] da.
 [K-NOM delivered NO]-TOP [Osono-NOM bread-ACC baked NO] COP
 Int. ‘It is Osono’s-baked bread that Kiki delivered.’

Finally, Hoshi (1995: 122, footnote 5) and Tonosaki (1998: 145-6) note that *no* in FRs may be replaced with a lexical noun such as *yatsu* (= ‘thing’), while this is not true of HIRs. This may be taken as another piece of evidence for the asymmetry between FRs and HIRs. Still, some caution is required. If *no* in FRs and in HIRs is replaced with *yatsu*, as in (6.33) and (6.34), the resulting strings are just instances of head-external relatives. (6.33) is grammatical simply as a head-external relative; (6.34) is ungrammatical as a head-external relative since the gap is occupied by the non-null item *ringo* (= ‘apple’). Thus, I shall set aside these data as orthogonal to the discussion of the nominal/non-nominal nature of HIRs.

(6.33) Free Relatives (FRs) with *yatsu* (= ‘thing’)

Kiki-wa [tsukue-no-ue-ni oite-atta no/yatsu]-o tabeta.
 K-TOP [table-GEN-top-at place-existed NO/thing]-ACC ate
 ‘Kiki ate what was on a table.’ / ‘Kiki ate the thing on a table.’

(6.34) Head-Internal Relatives (HIRs) with *yatsu* (= ‘thing’)

Kiki-wa [ringo-ga tsukue-no-ue-ni oite-atta no/*yatsu]-o tabeta.

K-TOP [apple-NOM table-GEN-top-at place-existed NO/thing]-ACC ate

‘An apple was on a table and Kiki ate it.’ / *‘An apple was on a table and Kiki ate the thing.’

There are numerous contrasts between FRs and HIRs, suggesting that FRs have **nominal** features while HIRs have **non-nominal** features. On the other hand, there are some indications that HIRs exhibit **nominal** characteristics as well, to which we shall turn next.

2.5.2 Nominality of HIRs

In considering the nominality of HIRs, the most striking fact is that the *no*-part is case-marked. In almost all of our examples so far, the *no*-part is marked by the nominative-case particle *ga* or the accusative-case particle *o*. Contrary to initial remarks (e.g., Hoshi 1995, Ishii 1989, Murasugi 1996, Ohori 1994-95), the *no*-part may be marked by a wider range of particles (Kim 2007). For instance, though Murasugi (1996: 344) states that the *no*-part cannot be marked by the particle *kara* (= ‘from’), Kuroda (2005: 185) suggests that it is fine, with which I agree; see (6.35). The *no*-part may also be marked by the particle *e* (= ‘to’) as in (6.36). Watanabe (1992: 259) observes that the *no*-part in HIRs may be marked by the genitive-case particle *no*, as in (6.37).

(6.35) Head-Internal Relatives (HIRs)

Keisatsu-wa [dorobou-ga mise-kara detekita no]-kara nusunda

police-TOP [robber-NOM shop-from coming.out NO]-from stolen

houseki-o toriageta.

jewels-ACC took

‘The police took the stolen jewels from the robber that came out from the shop.’

(6.36) Head-Internal Relatives (HIRs)

Ojisan-wa [Kiki-ga hashitte-kita no]-e dekkiburashi-o tewatashita.
 old.man-TOP [K-NOM run-came NO]-to deck.brush-ACC handed
 ‘Kiki came running and an old man handed a deck brush to her.’

(6.37) Head-Internal Relatives (HIRs)

Mary-ga [John-ga [jibun-no gakusei-ga juuyouna kasetu-o
 M-NOM [J-NOM [self-GEN student-NOM important hypothesis-ACC
 teianshita to] jimanshiteita no]-no kekkan-o shitekishita.
 proposed COMP] boasted NO]-GEN defect-ACC pointed.out
 ‘Mary pointed out a defect of the important hypothesis which John boasted that
 his student proposed.’

Another piece of data that indicates the nominality of HIRs concerns so-called ‘quantifier float’ (Hoshi 1995: 36-50). It is widely held that quantifier float is clause-bounded (e.g., Miyagawa 1989). In (6.38), the numeral quantifier *futatsu* (= ‘2’) is hosted by *kotoba* (= ‘language’). *Futatsu* can be floated clause-internally as in (6.39), but not clause-externally as in (6.40).

(6.38) Clause embedding with a numeral quantifier

Kiki-wa [Jiji-ga kotoba-o futatsu hanaseru to] itta.
 K-TOP [J-NOM language-ACC 2 can.speak COMP] said
 ‘Kiki said that Jiji can speak 2 languages.’

(6.39) Clause embedding with quantifier float

Kiki-wa [futatsu Jiji-ga kotoba-o hanaseru to] itta.
 K-TOP [2 J-NOM language-ACC can.speak COMP] said
 ‘Kiki said that Jiji can speak 2 languages.’

(6.40) Clause embedding with quantifier float

*Kiki-wa [Jiji-ga kotoba-o hanaseru to] futatsu itta.
 K-TOP [J-NOM language-ACC can.speak COMP] 2 said

In (6.41), the numeral quantifier *mittsu* (= ‘3’) is hosted by *pan* (= ‘bread’) within the HIR clause. Interestingly, *mittsu* is licensed outside of the HIR clause in (6.42). Given the locality restriction on quantifier float (see (6.40)), which would prevent *mittsu* from being floated across the HIR clause boundary, it looks as though the case-marked *no*-part stands as a nominal which hosts *mittsu* within a local domain.

(6.41) Head-Internal Relatives (HIRs) with a numeral quantifier

Kiki-wa [pan-ga mittsu teiburu-ni oiteatta no]-o tabeta.
 K-TOP [bread-NOM 3 table-on place.existed NO]-ACC ate
 ‘3 slices of bread were on a table and Kiki ate them.’

(6.42) Head-Internal Relatives (HIRs) with a numeral quantifier

Kiki-wa [pan-ga teiburu-ni oiteatta no]-o mittsu tabeta.
 K-TOP [bread-NOM table-on place.existed NO]-ACC 3 ate
 ‘Some bread was on a table and Kiki ate 3 slices.’

Finally, the nominality of HIRs is also suggested by the licensing of ‘secondary predicates’ (Hoshi 1995: 52-7). It has been pointed out that secondary predicates cannot be licensed across a clause boundary (Koizumi 1994). Consider the contrast between (6.43) and (6.44). In (6.43), the secondary predicate *nama-de* (= ‘raw’) is licensed by *sakana* (= ‘fish’) within the *no*-part, an instance of situation nominalisation. In (6.44), the secondary predicate *nama-de* cannot be licensed by *sakana* since they are not within a local clause.

(6.43) Secondary predicate in situation nominalisation

Kiki-wa [Jiji-ga sakana-o nama-de taberu no]-o mita.
 K-TOP [J-NOM fish-ACC raw eat NO]-ACC saw
 ‘Kiki saw that Jiji ate the fish raw.’

(6.44) Secondary predicate in situation nominalisation

*Kiki-wa [Jiji-ga sakana-o taberu no]-o nama-de mita.
 K-TOP [J-NOM fish-ACC eat NO]-ACC raw saw
 Int. ‘Kiki saw that Jiji ate the fish raw.’

In the HIR (6.45), however, the secondary predicate *nama-de* is licensed outside of the HIR clause. Given the locality constraint on secondary predicates (see (6.44)), it looks as if the case-marked *no*-part serves as a nominal licensing *nama-de* in a local domain.

(6.45) Secondary predicate in HIRs

Jiji-wa [sakana-ga teiburu-ni oite-atta no]-o nama-de tabeta.
 J-TOP [fish-NOM table-on place-existed NO]-ACC raw ate
 ‘Fish was placed on a table, and Jiji ate it raw.’

2.5.3 Consequences

Unlike FRs, which have the nominal status alone, HIRs possess **both** nominal and non-nominal characteristics (Tsubomoto 1995). This ambivalent status resists an account of HIRs in terms of the nominaliser *no* (Sect. 3, Chap. 5) since it models **only** the nominal features. Thus, HIRs cannot be characterised by the nominaliser *no*. There is another set of data that points to the same conclusion: the nominaliser *no* fails to explain why HIRs are subject to the Relevancy Condition (Kuroda 1992). One construal of ‘relevancy’ is temporal contiguity (Sect. 2.2). For instance, the HIR (6.46) cannot be interpreted as: ‘A friend cried 1 year ago and Kiki consoled her/him today.’ By contrast, this reading is possible in the FR (6.47). This asymmetry would remain a mystery if we adopted the nominaliser *no* for **both** HIRs and FRs.

(6.46) Head-Internal Relatives (HIRs)

[Tomodachi-ga naita no]-o Kiki-ga nagusameta.
 [friend-NOM cried NO]-ACC K-NOM consoled
 ‘A friend cried and Kiki consoled her/him.’

(6.47) Free Relatives (FRs)

[Naita no]-o Kiki-ga nagusameta.

[cried NO]-ACC K-NOM consoled

‘Kiki consoled a person who cried.’

2.6 Summary

This section has offered a substantial array of data on HIRs, which are condensed into the following bullet points. These constitute desiderata for a valid account of HIRs.

- The HIR head, though it is internal to the relative clause, is licensed by the embedding verb (Sect. 2.1).
- HIRs are highly context-dependent. The head is identified pragmatically, and the relation between the HIR and the embedding clauses is also determined pragmatically modulo the Relevancy Condition (Sect. 2.2).
- Context is also important for quantifier scope, maximality, and negation. At the same time, there is a grammatical constraint: quantifiers in the HIR clause do not interplay with quantifiers in the embedding clause (Sect. 2.3).
- HIRs with a complex NP seem to be slightly degraded compared to HIRs with other types of island. Still, island effects are generally absent (Sect. 2.4).
- Unlike FRs, which have only the nominal features, HIRs are ambivalent: they exhibit both nominal and non-nominal characteristics (Sect. 2.5).

3 Previous Studies on HIRs

3.1 Introduction

There is a growing body of research on HIRs in various languages (see Cole 1987 and subsequent work). HIRs are arguably the most extensively debated topic in the syntax and semantics of Japanese relatives. Sect. 3.2 succinctly traces the literature on Japanese

HIRs, and the following sub-sections take a closer look at the two influential accounts: Sect. 3.3 on Kim's (2007, 2008a, 2008b, 2009) E-type account and Sect. 3.4 on Grosu (2010) and Grosu & Landman's (2012) ChR account. Sect. 3.5 summarises the section.

3.2 A Brief History

The first work on this topic is a series of papers by Shige-Yuki Kuroda (1974, 1975-76, 1976-77). Kuroda develops an original account in Kuroda (1998, 2005), advocating that an embedding verb assigns its theta role to an argument in the HIR clause across the category S (or CP, TP, etc.). Though his account is insightful, it is not reviewed in this thesis; see Hasegawa (2002) and Hoshi (2002) for thoughtful critiques.

After Kuroda's trendsetting studies, early accounts of Japanese HIRs in the context of generative grammar made crucial use of **LF movement** of the internal head (usually, to an argument position of the embedding verb). This is because one of the chief concerns was how the embedding verb licenses an argument that is deep inside the HIR clause. Ishii (1989) and Ito (1986) contend that there is an external head node, to which the internal head moves at LF. Watanabe (1992) argues that (after a null operator moves at S-Structure), the internal head moves to Spec CP at LF (see also Watanabe 2003). A different version of movement analysis, together with the new semantics of HIRs, is outlined in Erlewine & Gould (in press). A primary piece of evidence for the movement-based approach is island-sensitivity. But it has been noted in Sect. 2.4 that HIRs are not sensitive to islands (Hoshi 1995, Kuroda 2005, Mihara & Hiraiwa 2006).

Island-insensitivity of HIRs motivated an entirely different approach: HIRs are not relatives but **adverbial adjuncts** that modify the embedding clause (Hoshi 1996, Mihara 1994, Murasugi 1994, 1996). Mihara (1994: 245-9) claims that (i) the HIR clause is a complex adjunct NP to be adjoined to IP (or any equivalents, such as TP) or VP of the embedding clause and that (ii) the embedding clause contains a zero pronoun (i.e., small *pro*) that may be co-indexed with NPs in the complex adjunct NP. This account handles, among other things, the indeterminacy of the head (Sect. 2.2): an

argument position in the embedding clause is inhabited by a zero pronoun, and the zero pronoun may refer to a contextually salient individual (or a group of such individuals) denoted by NPs in the complex adjunct NP. But there are daunting counterexamples. As exemplified in Sect. 2.5, the HIR clause may be marked by a wide range of particles; if the HIR is an adjunct that modifies IP or VP, it is not obvious why it can be marked by, say, the genitive-case particle *no*. This issue may be solved by multiple case-checking (Hoshi 1996; see Chomsky 1995), but Hoshi's analysis stipulates a licensing condition on the particle *no*. For additional counterarguments, see Kuroda (2005: 177-88).

Another line of thought to dispense with LF movement to establish a link between the internal head and the embedding verb exploited the **E-type anaphora** mechanism (Hoshi 1995, Kim 2007, 2008a, 2008b, 2009, Matsuda 2002, Shimoyama 1999, 2001); see Cooper (1979), Evans (1977, 1980), and Heim (1990) for E-type anaphora. This is an influential approach, but it has been noted that there are differences between the HIR interpretation and E-type anaphora (Grosu 2010); accordingly, it ends up stipulating machinery solely specific to HIRs. An E-type approach is also adopted by Kitagawa (2005), who classifies Japanese HIRs into subtypes, one of which is analysed by an E-type mechanism (see also Kitagawa & Ross 1982). This non-uniform account, however, is refuted in Kim (2008b). Among a family of E-type accounts of HIRs, the most advanced work is Kim (2007, 2008a, 2008b, 2009). In her analysis, *no* in HIRs denotes a set of entities that bear (i) a salient thematic role and (ii) a salient predicate in the state induced by the eventuality of the HIR clause. This set is maximalised by the feature [+*definite*] of the head D, and the accruing singleton set is the denotation of the internal head. Kim's work will be critically evaluated in Sect. 3.3 below.

The most recent approach is Grosu (2010) and Grosu & Landman's (2012) **ChR** (Choose Role) account. ChR is a functional head that selects VP as a complement. ChRP denotes a set of events where an individual bears a salient thematic role. This is turned into a closed proposition by the event existential closure. It is abstracted over the

individual at issue, and the set denoted by the resulting predicate is maximalised by the feature [MAX] of the head C. The gist of the analysis is that HIRs are treated by the same machinery as employed for other ‘definite’ relatives (Grosu & Landman 1998): relativisation abstracts a proposition over an individual variable and maximalises a set of individuals denoted. The ChR account will be surveyed in Sect. 3.4 below.

Of these previous studies, conspicuous are Kim’s (2007, 2008a, 2008b, 2009) E-type account and Grosu (2010) and Grosu & Landman’s (2012) ChR account. They are critically examined in the ensuing two sub-sections.

3.3 Kim (2007, 2008a, 2008b, 2009)

Kim argues that the correct understanding of HIRs requires us to consider not the event denoted by the HIR clause but the **state** induced by the event. Kim goes on to propose that the internal head of HIRs denotes a set of individuals that bear a salient **thematic role** and a salient **property** in the state. This set of individuals is **maximalised**, and the output is the denotation of the internal head.

Then, how to pick out a set of individuals that bear a salient thematic role and a salient property in a state? This is achieved by the denotation of *no* (6.48). R and P are variables for thematic roles and properties, to be assigned a value by the assignment function g. s and x are variables for states and individuals.

(6.48) Denotation of *no* in Kim (2007, 2008a, 2008b, 2009)³

$$\llbracket no \rrbracket = \lambda s. \lambda x. [g(R)(x)(s) \& g(P)(x)]$$

In the HIR (6.49), adapted from Kim (2007: 281), suppose that g assigns the role *agent*’ to R and the property *thief*’ to P. In (6.50), *agent*’(x)(s) dictates that x bears an agent role in the state s, and *thief*’(x) dictates that x (or, more precisely, an individual x with a salient thematic role in the state s) has the property *thief*’.

³ Double-square brackets are the denotation assignment function (Chierchia & McConnell-Ginet 2000, Heim & Kratzer 1998), as is standard in all Montague-based formalisms.

(6.49) Head-internal Relatives (HIRs)

Antony-wa [[dorobou-ga nige-teiru]- ϕ no]-o tsukamae-ta.
 A-TOP [[thief-NOM run.away-IMPRF]-REL NO]-ACC catch-PAST
 ‘A/the thief was running away and Antony caught him (= the thief).’

(6.50) $\lambda s.\lambda x.[agent'(x)(s)\&thief'(x)]$

The HIR clause denotes a set of events where a thief runs away, and this eventuality induces an in-progress state of a thief who is in a developmental stage of running away. An input to $\llbracket no \rrbracket$ (6.48) must be a state variable s of type-e, but the in-progress state is not of type-e (see below). This type-mismatch motivates LF movement of the HIR clause, leaving behind a state variable s of type-e as a trace. This trace, as a type-e element, is a proper input to $\llbracket no \rrbracket$ (6.48), and the formula (6.51) obtains.

(6.51) $\lambda x.[agent'(x)(s)\&thief'(x)]$

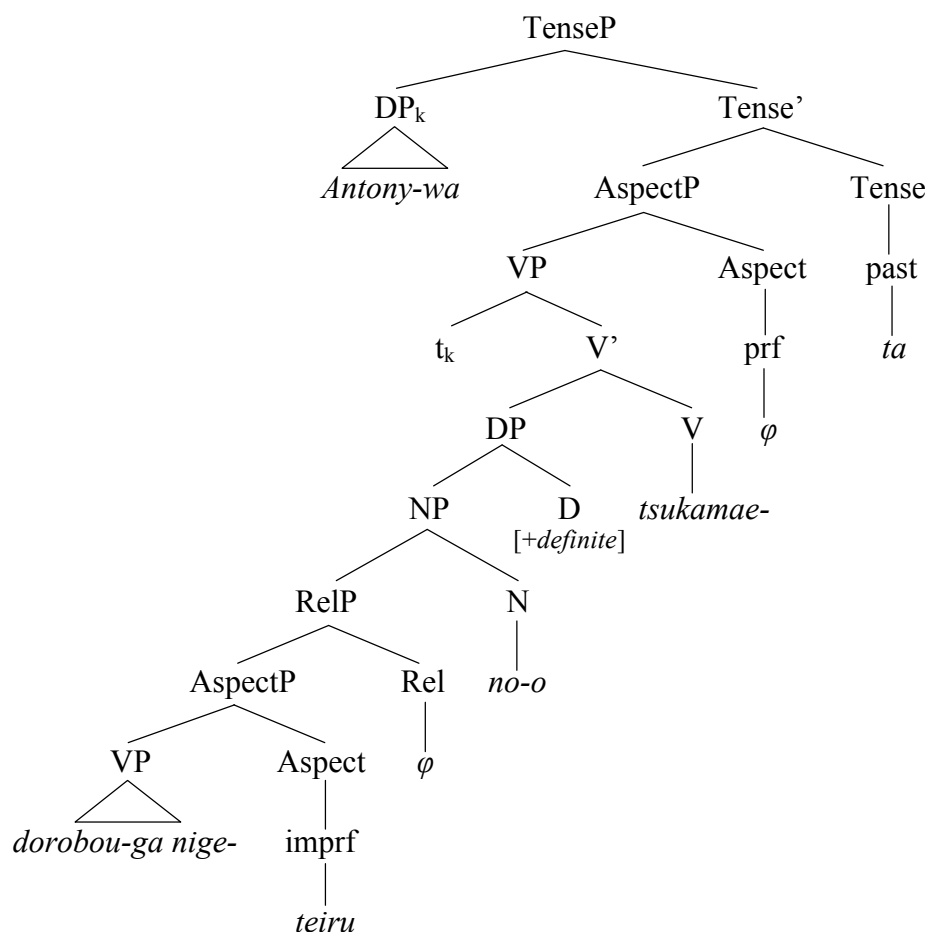
This formula denotes a set of individuals that bear the agent role and the property *thief* in the state s . As will be mentioned shortly, *no* is a complement of a covert determiner D , whose denotation is (6.52), where σ is a maximalising operator (cf., Link 1983). As (6.51) denotes a set of individuals, it is an appropriate input to (6.52). The resulting formula (6.53) represents the maximal set of individuals that bear the agent role and the property *thief* in the state s . This is the denotation of the internal head, selected by the content of the embedding verb *tsukamae-* (= ‘catch’).

(6.52) $\llbracket D \rrbracket = \lambda Q.\sigma x.[Q(x)]$ (6.53) $\sigma x.[agent'(x)(s)\&thief'(x)]$

Let us track Kim’s account in more detail by probing its presumed syntactic structure. The HIR (6.49) is mapped onto the overt syntactic structure (6.54) (see Kim 2007: 299). The HIR clause corresponds to RelP, the head Rel (i.e., Relative) being null in Japanese (but overt in Korean). It is further stipulated that unlike the embedding

structure, Rel directly selects AspectP without TenseP. (This makes sure that the tense of the HIR clause is relativised to that of the embedding clause; see (6.59) below, where the tense variable t for the HIR clause is bound by the operator in the main structure.) RelP is a complement of the head N (i.e., the category of *no*), and the complex NP is a complement of the head D that is associated with the feature [+definite]. This DP serves as an E-type pronoun (see Cooper's (1979) conception of E-type pronouns as definite descriptions) and it stands as an argument of the embedding verb *tsukamae-* (= 'catch').

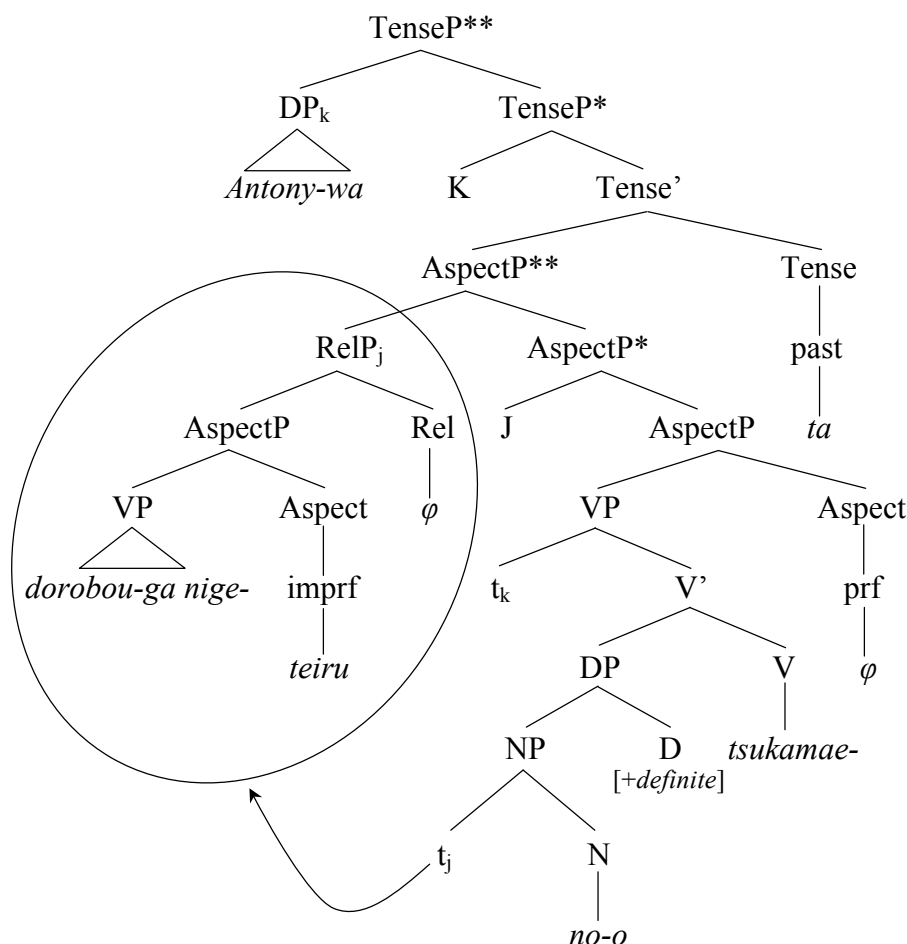
(6.54) Overt syntactic structure of the HIR (6.49)



This overt syntactic structure is mapped onto the LF representation (6.55), an input to the semantic composition (see Kim 2007: 300). RelP has moved to AspectP due to the semantic type mismatch between RelP and N. As stated above, $\llbracket no \rrbracket$ requires a type-e item as an input, but $\llbracket RelP \rrbracket$ is assumed to be of type- $\langle\langle s, it \rangle, \langle i, t \rangle\rangle$, where s is

a type of states and i is a type of times. The denotation of the trace of $\llbracket \text{RelP} \rrbracket$, namely $\llbracket t_j \rrbracket$, is of type- e , and this is an appropriate input to $\llbracket no \rrbracket$.

(6.55) Covert syntactic structure of the HIR (6.49)⁴



To illustrate the semantic composition, two key items need to be highlighted. First, $\llbracket no \rrbracket$ takes $\llbracket t_j \rrbracket$, a state variable s (i.e., trace of the HIR clause) and outputs a set of individuals x such that x bears a salient thematic role and a salient property in the state s . This set of individuals is represented in (6.51), repeated here as (6.56).

(6.56) $\lambda x. [\text{agent}'(x)(s) \& \text{thief}(x)]$

⁴ The indices K and J apply 'predicate abstraction' over a closed proposition (Heim & Kratzer 1998). This is because the sister node of K and J is decorated with a type- t content but the mother node of K and J needs to be decorated with a type- $\langle e, t \rangle$ content.

Second, the denotation of the null determiner D, which bears the feature [+*definite*], is defined as (6.52), repeated here as (6.57). If this applies to the set (6.56), it picks out a singleton set of individuals as in (6.53), repeated as (6.58). This operation engenders the maximality effect of the HIR.

$$(6.57) \quad \llbracket D \rrbracket = \lambda Q. \sigma x. [Q(x)]$$

$$(6.58) \quad \sigma x. [\textit{agent}'(x)(s) \& \textit{thief}'(x)]$$

The final representation is (6.59) (see Kim 2007: 307); see also Kim (2009) for an elaboration in terms of situation theory. The underlined part denotes the maximal set of individuals *x* such that *x* is a thief and *x* bears the agent role in the in-progress state induced by the eventuality of the HIR clause. This singleton set is the theme of the event described by the embedding clause: the Antony's catching event.

$$(6.59) \quad \text{Denotation of the HIR (6.49)}$$

$$\begin{aligned} & \exists t[t < \textit{now}' \& \exists s[\exists e[\exists x[\textit{run.away}'(e) \& \textit{agent}'(x)(e) \& \textit{thief}'(x)] \\ & \& \textit{in-progress}'(s, e) \& t \subseteq \textit{runtime}'(s)] \& \exists e'[\textit{catch}'(e') \& \textit{agent}'(\textit{Antony}')(e') \\ & \& \textit{theme}'(\underline{\sigma x[\textit{agent}'(x)(s) \& \textit{thief}'(x)]}(e') \& \textit{run-time}'(e') \subseteq t]]] \end{aligned}$$

In (6.59), the same operator binds the state variable *s* in the denotation of the head and the state variable *s* in the denotation of the HIR clause. This double binding restricts the domain out of which a thematic role and a property are chosen to the denotation of the HIR clause. The selection of a thematic role and a property is also constrained by other factors, such as the lexical semantics of aspect and an axiom regarding thematic roles (i.e., 'An event and its in-progress state have identical thematic roles, with identical values.' (adapted from Kim 2007: 304)).

In Kim's analysis, the head of HIRs is identified in the state induced by the event described by the HIR clause (rather than the event *per se*). This idea, together with assumptions regarding aspect and tense (Kratzer 1998, Parsons 1990), accounts for how acceptability of HIRs varies depending on the aspect of the embedded verb.

Another merit is that it accommodates cases like (6.60) (adapted from the Korean data in Chung & Kim (2003: 58)).

(6.60) Head-internal Relatives (HIRs)

?John-wa [zubon-ga yogore-ta no]-o fukitotta.

J-TOP [pants-NOM get.dirty-PRF NO]-ACC wiped.from

‘The pants got dirty and John wiped the dirt from the pants.’

Following Parsons (1990), Kim (2007: 293) assumes that when an HIR clause has a telic predicate bearing perfect aspect, the event denoted by the HIR clause yields a ‘target state,’ a temporary state of the (culminating) theme of the event. For instance, a possible target state in (6.60) is the temporal state of the dirt that has accumulated on the pants. This target state is the denotation of the head in this HIR. This account is a nice consequence of Kim’s proposal: the denotation of the head is found not in the event described by the HIR clause but the state induced by the event. (In (6.59), recall that the denotation of the head is found with respect to the in-progress state induced by the event described by the HIR clause.)

The achievements of Kim’s account are twofold. First, building upon Hoshi (1995) and Shimoyama (1999, 2001), Kim advances an E-type account of HIRs by giving the formal conditions on the resolution of the E-type pronoun in the HIR clause. This attempt also contributes to a clarification of the Relevancy Condition. Second, Kim formalises the mechanism of how these conditions are satisfied by proposing (i) an explicitly articulated syntactic structure, (ii) the detailed entries of key items (e.g., *no*), and (iii) the fine-grained semantics of eventuality.

Nonetheless, Kim’s account suffers from some problems, both conceptual and empirical. The conceptual one is that Kim postulates additional constraints on the usual E-type device. This is because there are many **discrepancies** between E-type anaphora and the HIR. First, Shimoyama (1999: 172) notes that the head of the HIR (6.61) may be *shinbun* (= ‘newspaper’) but it cannot be the group of *shinbun* and *hon* (= ‘book’).

This group reading is not possible even though the split head phenomenon is possible in HIRs (Sect. 2.2). By contrast, in E-type anaphora (6.62), the antecedent may be the group of *shinbun* and *hon*.

(6.61) Head-Internal Relatives (HIRs)

*Hanako-ga hon-o 3-satsu katte-kita. Taro-wa [Hanako-ga mata
H-NOM book-ACC 3-CL buy-came T-NOM [H-NOM also
dono shinbun-mo katte-kita no]-o tana-ni narabeta.
every newspaper-also buy-came NO]-ACC shelf-on placed

Int. ‘Hanako bought (and brought) 3 books. Hanako also bought (and brought) every newspaper and Taro shelved them (= books and newspapers).’ (The HIR string is acceptable if the internal head is *shinbun* alone.)

(6.62) E-type anaphora

Hanako-ga hon-o 3-satsu katte-kita. Hanako-ga mata dono
H-NOM book-ACC 3-CL buy-came H-NOM also every
shinbun-mo katte-kita. Taro-wa (sorera-o) tana-ni narabeta.
newspaper buy-came T-TOP (those-ACC) shelf-on placed

‘Hanako bought (and brought) 3 books. Hanako also bought (and brought) every newspaper. Taro shelved them (= books and newspapers).’

Another divergence is the contrast between (6.63) and (6.64) (Grosu 2010: 237, based on Shimoyama (2001: 135-6)). In E-type anaphora (6.63), in an appropriate context, say, where professors are talking about their students who wanted to come to the party, *karera* (= ‘they’) could denote individuals who did not come to the party (but who were writing term papers at home). This is not true of the HIR (6.64). The only possible reading is that 3 students came to the party and they were also writing term papers at home, the absurdness being expressed by ‡. Kim accounts for these discrepancies by stipulating mechanisms which are specific to HIRs (but not E-type anaphora).

(6.63) E-type anaphora

Choudo-3-nin-no insei-ga doyoubi-no-paatii-ni kita.
 exactly-3-CL-GEN grad.student-NOM Saturday-GEN-party-to came
 Karera-wa jitsuwa uchi-de taamupeipaa-o kaiteita.
 they-TOP in.fact home-at term.paper-ACC were.writing
 ‘Exactly 3 graduate students came to the party on Saturday. In fact, they (= the
 students who did not come to the party) were writing term papers at home.’

(6.64) Head-Internal Relatives (HIRs)

#[Choudo-3-nin-no insei-ga doyoubi-no-paatii-ni kita no]-ga
 [exactly-3-CL-GEN grad.student-NOM Saturday-GEN-party-to came NO]-NOM
 jitsuwa uchi-de taamupeipaa-o kaiteita.
 in.fact home-at term.paper-ACC were.writing
 ‘Exactly 3 graduate students came to the party on Saturday, and in fact they (= these 3 students) were writing term papers at home.’

The empirical objections to Kim’s account are as follows. First, the maximality effect is optional, as illustrated in (6.17), reproduced here as (6.65) (Sect. 2.3). In Kim’s account, the maximality effect obtains **through** the feature [+*definite*] of the head D, the repercussion being the failure to predict the context-dependency of maximality.

(6.65) Head-Internal Relatives (HIRs)

Dono-joukyaku_i-mo [saifu-ni kaisuken-ga haitteita no]-o
 every-passenger-too [wallet-in coupon.ticket-NOM was.present NO]-ACC
 toridashite kaisatsu-ni ireta.
 take.out ticket.checker-to put
 ‘Every passenger took out a coupon ticket that she/he had in (her/his) wallet and put it in the ticket checker.’ (adapted from Kubota & Smith (2007: 154))

The second issue is island-insensitivity of HIRs (Sect. 2.4), as in (6.20), repeated here as (6.66). In Kim’s analysis, the head is detected by selecting a thematic role and a predicate in the state induced by the eventuality of the HIR clause. As pointed out in

Grosu (2010: 250), however, Kim's analysis concerns the eventuality of the **highest** clause within the HIR clause and cannot deal with even long-distance dependencies of HIRs, let alone island-insensitivity of HIRs.

(6.66) Head-Internal Relatives (HIRs) involving a complex NP

[Taro-ga [Hanako-ga subarashii ronbun-o kaita toiu houkoku]-o
 [T-NOM [H-NOM excellent paper-ACC wrote TOIU report]-ACC
 kiiteita no]-ga tsuini shuppansareta.
 has.heard NO]-NOM finally was.published
 'Taro has heard a report that Hanako wrote an excellent paper, and the paper
 was finally published.'

Finally, what about the negation data (Sect. 2.3)? Kim's analysis would handle them correctly. In (6.19), reproduced here as (6.67), *saifu* (= 'wallet') would be identified as a head if we let the assignment function *g* attribute the role *theme*' to the variable *R* and the property *saifu*' to the variable *P* in (6.68), as in (6.69).

(6.67) Head-Internal Relatives (HIRs)

Dorobou-wa [saifu-ga kinko-ni haittei-naka-tta no]-o
 thief-TOP [wallet-NOM safe-at put.inside-NEG-PAST NO]-ACC
 manmato nusumi-dashita.
 successfully steal-took.away
 'A wallet was not inside a safe (but outside the safe), and a thief successfully
 stole it.'

(6.68) $\sigma x.[\lambda x.[g(R)(x)(s) \& g(P)(x)(s)]]$

(6.69) $\sigma x.[\lambda x.[theme'(x)(s) \& saifu'(x)(s)]]$

As in (6.70), however, the negation data also display a long-distance dependency. Since Kim's account concerns only the state induced by the eventuality of the highest clause in the HIR clause, *houseki* (= 'jewels') cannot count as a head. Hence, Kim does not account for the full spectrum of negation data.

(6.70) Head-Internal Relatives

Dorobou-wa [aru-yuumeijin-ga [ie-de-wa houseki-o kinko-ni
 thief-TOP [certain-celebrity-NOM [house-at-TOP jewels-ACC safe-at
 irete-nai to] TV-de itteita no]-o manmato nusumi-dashita.
 put.inside-NEG COMP] TV-at said NO]-ACC successfully steal-took.away
 ‘A celebrity said in a TV programme that she tended not to put her jewels in a
 safe at home, and the thief successfully stole it.’

An E-type approach to Japanese HIRs, initiated by Hoshi (1995), has been a promising line of analysis, and this is advanced in Kim (2007, 2008a, 2008b, 2009). Yet, Kim’s account still needs to overcome conceptual and empirical difficulties. In the recent literature, an alternative account which does not resort to the E-type device has been proposed, to which we shall turn next.

3.4 Grosu (2010) and Grosu & Landman (2012)

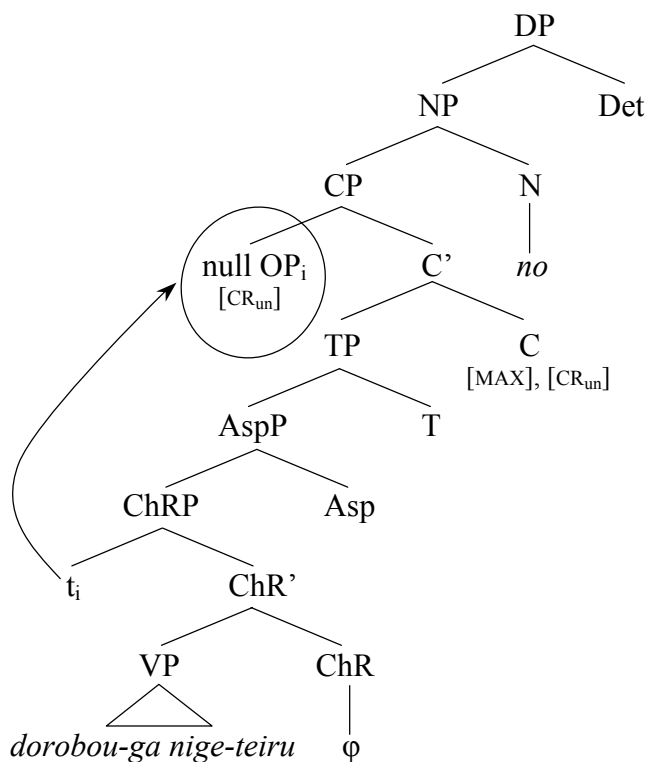
The cutting edge approach to Japanese HIRs is the ChR account (Grosu 2010, Grosu & Landman 2012). ChR (Choose Role) is a functional head that takes VP as a complement. VP denotes a set of events where every participant is assigned a thematic role. What ChR does is to (i) take this event representation, (ii) open up a thematic role, and (iii) equate a free variable *x* with the one bearing this role. Through this process (and other mechanisms assumed for, e.g., aspect), the HIR clause denotes a proposition in which the free variable *x* bears a salient thematic role in the event at issue. This proposition is **abstracted** over the free variable *x*, with the output denoting a set of individuals that bear a salient thematic role in the event. Finally, this set is **maximalised** by the feature [MAX] of the head *C*. The crux of the argument is that HIRs are subsumed under the general machinery for ‘definite’ relatives (Grosu & Landman 1998): relativisation is a process to (i) abstract a proposition over an individual variable and (ii) maximalise a set of individuals characterised by the abstracted proposition.

The above exposition is exemplified by considering the HIR (6.49) in the last sub-section, reproduced as (6.71). The *no*-part is assigned the covert syntactic structure (6.72) (cf., Grosu 2010: 262).

(6.71) Head-internal Relatives (HIRs)

Antony-wa [[dorobou-ga nige-teiru]- ϕ no]-o tsukamae-ta.
 A-TOP [[thief-NOM run.away-IMPRF]-REL NO]-ACC catch-PAST
 ‘A/the thief was running away and Antony caught him (= the thief).’

(6.72) Covert syntactic structure of the HIR clause + *no* in (6.71)



ChR is a null functional head and selects VP as a complement. Spec of ChRP hosts the null OP, which has the uninterpretable feature [CR_{un}]. The null OP moves to Spec of CP to check this uninterpretable feature by Spec-Head feature checking; the head C is assumed to bear the feature [CR_{un}]. This OP movement yields island effects in the form of cyclic A'-movement. (In long-distance cases, any number of CPs may be interpolated between TP and CP.) The head C encodes the feature [MAX] too; as we shall see shortly,

the denotation of TP, a closed proposition, is abstracted over an individual variable, and the resulting set is maximalised by the feature [MAX]. CP is a complement of N (i.e., the category of *no*), which is devoid of semantic content. This complex NP is part of DP, which serves as the head of the HIR.

Now, let us consider the semantic compositions of HIRs in detail to explicate the ChR mechanism and the processes ‘abstraction’ and ‘maximalisation.’ ChR selects VP as a complement. $\llbracket \text{VP} \rrbracket$ in (6.72) is (6.73), which characterises a set of events where a thief runs away (cf., Parsons 1990).

$$(6.73) \quad \lambda e. [\exists y. [nige'(e) \& [agent'(e)=y \& thief(y)]]]$$

$\llbracket \text{ChR} \rrbracket$ is defined as (6.74), where E is a set of events denoted by VP, x is an individual variable, e is an event variable, and C_E is a function that maps events e in E onto individuals that bear a salient thematic role in e (Grosu & Landman 2012: 169).

$$(6.74) \quad \llbracket \text{ChR} \rrbracket = \lambda E. \lambda x. \lambda e. [E(e) \& C_E(e)=x]$$

The application of (6.74) to (6.73) yields the representation (6.75), which denotes a relation between individuals x and events e : x bears the salient thematic role *agent'* in an event e , or a ‘thief’s running away’ event. In the present context, $C_E(e)$ is *agent'(e)*, which denotes an individual bearing the agent role in an event e .

$$(6.75) \quad \lambda x. \lambda e. [\exists y. [nige'(e) \& [agent'(e)=y \& thief(y)]] \& agent'(e)=x]$$

This representation is combined with $\llbracket t_i \rrbracket$, which is assumed to be a free variable x . It is then enriched with aspect and tense. In particular, an aspect applies existential closure over the event variable e , as in (6.76). (For the sake of simplicity, the semantic content of aspect is ignored.)

$$(6.76) \quad \exists e. [\exists y. [nige'(e) \& [agent'(e)=y \& thief(y)]] \& agent'(e)=x]$$

Now, this proposition is abstracted over the individual variable x , as in (6.77), which characterises a set of individuals x such that there is a ‘thief’s running away’ event and x bears the agent role in the event. This set of individuals is then maximalised by the feature [MAX]. The singleton set (6.78) is the denotation of the head of the HIR clause.

$$(6.77) \quad \lambda x. [\exists e. [\exists y. [nige'(e) \& [agent'(e)=y \& thief(y)]] \& agent'(e)=x]]$$

$$(6.78) \quad \sigma(\lambda x. [\exists e. [\exists y. [nige'(e) \& [agent'(e)=y \& thief(y)]] \& agent'(e)=x]])$$

HIRs are thus modelled by the general machinery independently motivated for ‘definite’ relatives: (i) the abstraction of a proposition over an individual variable and (ii) the maximalisation of the resulting set of individuals. For the treatment of more complex data involving quantification, see Grosu & Landman (2012: Sect. 5).

Grosu (2010) and Grosu & Landman (2012) point out that the Relevancy Condition concerns not only time and space but also modals: ‘The event in which the denotation of the internal head is a participant, or some state resulting from this event, must temporally, spatially, and modally intersect with the event described by the matrix clause.’ (Grosu & Landman 2012: 172). The adequacy of this proposal is not discussed here since the content of the Relevancy Condition is still in dispute (Kim 2008a). Still, it is worth pinpointing the vagueness lurking in this formulation. Grosu (2010: 258) states that the condition is ‘essentially pragmatic,’ to be imposed on some constructions (e.g., HIRs) in some languages (e.g., Japanese). But in this conception, the pragmatic status of the Relevancy Condition is equivocal: even if it is ‘essentially pragmatic,’ one has to encode that this restriction is operative for HIRs (but **not** other constructions) and in Japanese (and Korean, but **not** other languages).⁵

The ChR account achieves both formal explicitness and broad coverage of data to a higher degree than the precursors in the literature. Still, the analysis is not feasible

⁵ In our analysis (Sect. 4), the grammar encodes that Japanese HIRs are subject to the Relevancy Condition, though the details of the condition are set aside, leaving it a possibility that the content of the condition itself may be essentially pragmatic.

in several respects. To begin with, there is a conceptual issue. The ChR account is based on the general relativisation mechanism: (i) abstraction of a proposition over an individual variable and (ii) maximalisation of the resulting set of individuals. The nature of this account may be general, but it is also based on stipulations **solely** specific to HIRs in Japanese (and Korean): the null functional category ChR, the null OP at Spec of ChRP, the feature $[CR_{un}]$ at Spec of CP, to mention a few. There are minor remarks on cross-linguistic implications of ChR, but they are not substantiated.

Some empirical issues also stand out. First, the ‘maximality’ problem for Kim (2007, 2008a, 2008b, 2009) repeats itself. In the ChR account, the maximality effect springs from the **encoded** feature $[MAX]$. Thus, the context-dependency of maximality (Kubota & Smith 2007) does not follow. Movement of the null OP is also dubious. It is claimed that the successive A’-movement of the null OP produces island effects, but we saw in Sect. 2.4 that HIRs were generally **not** sensitive to islands. Grosu (2010) and Grosu & Landman (2012) cite Watanabe’s (2003) data that appear to designate island-sensitivity of HIRs but the unacceptability of these examples seem to stem from other factors (e.g., the Relevancy Condition) (Hoshi 1995, Kuroda 2005, Mihara 1994, Mihara & Hiraiwa 2006; see Sect. 2.4). The ChR account is also challenged by the negation data (6.19), repeated here as (6.79). One might argue that the quantified head *saiфу* (= ‘a wallet’) moves over NegP at LF so that it out-scopes the negator. But this remedy is untenable since ChR **cannot** select NegP. This is because Grosu & Landman (2012: 176) assume that (i) VP denotes an open proposition with an event slot; (ii) ChR must select such an open proposition; but (iii) NegP has closed the proposition over the event slot before it is selected by ChR. Therefore, the HIR (6.79) would be wrongly predicted to be ungrammatical.

(6.79) Head-Internal Relatives (HIRs)

Dorobou-wa [saifu-ga kinko-ni haittei-naka-tta no]-o
 thief-TOP [wallet-NOM safe-at put.inside-NEG-PAST NO]-ACC
 manmato nusumi-dashita.
 successfully steal-took.away

‘A wallet was not inside a safe (but outside the safe), and a thief successfully stole it.’

The ChR account, the most up-to-date analysis in the literature, presents a rigorous mechanism for handling both structural and interpretive aspects of Japanese HIRs. Yet, the analysis is nonetheless incomplete. Of particular note is that the ChR account seems to be unable to license negation within HIR clauses, the novel data provided in Sect. 2.3.

3.5 Summary

The literature on Japanese HIRs presents a wide variety of analyses. The two forefront analyses of these are Kim’s (2007, 2008a, 2008b, 2009) E-type account and Grosu (2010) and Grosu & Landman’s (2012) ChR account. Both approaches uncover new characteristics of HIRs and offer a fairly solid analysis, but there are conceptual and empirical reservations. In the next section, we shall articulate a dynamic account of HIRs by expanding on the analyses of *no*-nominalisation (Chap. 4) and clefts (Chap. 5). The dynamic account deals with the data set out in Sect. 2 (including the ones that may jeopardise those previous analyses) and draws some new predictions.

4 A Dynamic Account of HIRs

4.1 Introduction

The main challenges posed by HIRs are (i) the intricate relation between structural and contextual circumstances and (ii) the ambivalent status of being nominal and non-nominal. The aim of this section is to construct an account of HIRs by blending the analyses of *no*-nominalisation (Chap. 4) and cleft constructions (Chap. 5).

The analysis to be presented relies on the dynamics of the HIR construal: the HIR clause establishes a context for the embedding clause to be processed. This flow of information is redolent of the cleft construal: the *no-wa* part sets a context for the focus item to be scanned. In Chap. 5, this information stream was modelled by proposing that the unit *no-wa* (i) maps a propositional tree of the presupposition clause onto another propositional tree and (ii) posits the requirement that this tree will be developed relative to a type-e term in the previous tree. This insight of **pairing two propositional trees** will be inherited by our analysis of HIRs.

To conceive of HIRs merely as pairing propositional trees is not enough since the relation between the HIR and the embedding clauses is tightly constrained (i.e., the Relevancy Condition). This is where the analysis of clefts is married with that of *no*-nominalisation. Chap. 4 contended that the nominaliser *no* built a type-e node within a new structure. This insight of **building type-e nodes** is also made use of in our analysis of HIRs. It will be proposed that the mapped tree has been partially articulated when it is introduced; this partial tree contains a situation node with a ‘Relevancy’ requirement.

Sect. 4.2 advocates a dynamic account of HIRs and Sect. 4.3 illustrates it with a simple example. Sect. 4.4 tests the adequacy of the account against the data presented in Sect. 2. Finally, Sect. 4.5 summarises the results.

4.2 The Proposal

Our account of HIRs takes over part of the analyses of *no*-nominalisation (Chap. 4) and clefts (Chap. 5). First, as in clefts, HIRs inversely LINK a propositional tree to **another propositional** tree, where the embedding clause will be processed. Second, as in *no*-nominalisation, the LINK transition creates **type-e** nodes in the emergent tree. But there is a crucial difference: *no*-nominalisation creates a node for an individual term, while HIRs create not only a node for an individual term **but also** a node for a situation term. The creation of a node for an individual term accounts for how the embedding predicate selects the content of an internal head. The creation of a situation node comes with the requirement that this situation term will stand in a certain relation to the situation term in the previous proposition. This requirement constrains the relation between the HIR and the embedding clauses (i.e., the Relevancy Condition).

A pressing question is how such LINK transitions are triggered. In clefts, the driving force was the cleft-marker *no-wa*. One might expect that there are other possible combinations of the nominaliser *no* and particles. I propose that *no* and a range of case particles form a unit for the HIR interpretation. That is, the fusion of *no* and the **topic** marker *wa* models clefts, while the fusion of *no* and a **case** marker models HIRs. This division nicely accords with the distribution of the topic marker *wa* and case markers in clefts and HIRs: the HIR clause cannot be followed by the topic marker *wa* (Ito 1986: 127, Matsuda 2002: 636-7).⁶

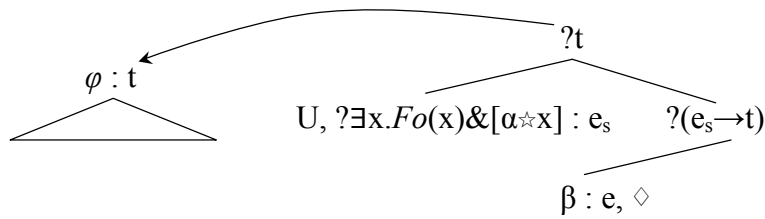
Our proposal is informally put in (6.80). As a schematic illustration, the unit *no-ga*, where *ga* is a nominative-case particle, generates the tree (6.81).

⁶ Kuroda (2005: 203) points out that there are some cases where the *no*-part in HIRs seems to be marked by *wa*, especially if it induces a ‘contrastive’ topic. Note that our analysis of *wa* set out in Chap. 3 (Sect. 5.3) mainly concerns a ‘thematic’ topic (rather than a ‘contrastive’ topic).

(6.80) Proposal

The unit ‘*no* + case particle’ encodes the procedures to perform these actions:

- a. to inversely LINK the tree of the HIR clause to another propositional tree,
- b. to build a situation node with the requirement that the term at this node will stand in a certain relation to the situation term in the previous proposition, and
- c. to build a node for an individual term (whose position is identified by the case particle) and decorate it with a type-e term within the previous proposition.

(6.81) Output structure of parsing *no-ga*

where α is a situation term in the evaluated proposition of the HIR clause

β is an individual term in the evaluated proposition of the HIR clause

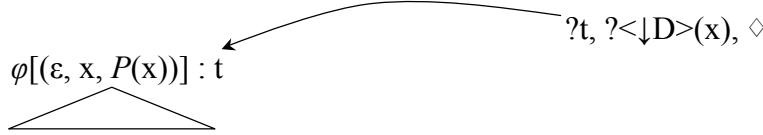
ϕ is the proposition of the HIR clause

In the emergent tree, a situation node is decorated with $?\exists x.Fo(x)\&[\alpha\star x]$, where Fo is a predicate for formula value, x ranges over situation terms, and $\star$ stands for whatever relationship holds between the two situation terms as governed by the Relevancy Condition.⁷ This requirement states that the current node will be decorated with a situation term and that this situation term will be in a Relevancy relation to the situation term α in the evaluated proposition of the HIR clause. The partial tree contains another type-e node, which is decorated with an individual term in the evaluated proposition of the HIR clause. This is the content of the internal head. The position of this node is signalled by a case particle. In (6.81), it is identified as a subject node position by the nominative-case particle *ga*.

⁷ In this thesis, the details of the relational predicate $\star$ are not addressed. Still, there are some new predictions that may be drawn from the present analysis; see Sect. 4.4.1.

The LINK transition for HIRs is sharply contrasted with clefts. In clefts, the cleft marker *no-wa* maps the propositional tree of the preceding clause onto an emergent propositional tree **without** internal structure, as in (6.82).

(6.82) Output structure of parsing *no-wa* in clefts



In HIRs, the unit ‘*no* + case particle’ induces a partially articulated tree. This partial tree ensures that (i) HIRs are subject to the Relevancy Condition (as modelled by the relational predicate $\star$) and that (ii) the head, though internal to the relative clause, is licensed by the embedding predicate.

The entry of the unit ‘*no* + case particle’ is formally presented as (6.83). φ_E is an evaluated proposition of the HIR clause. α is a situation term in φ_E and β is an individual term in φ_E . μ is a LOFT relation to be determined by a case particle: the nominative-case selects $\downarrow_1\downarrow_0$ (for subject), the accusative-case selects $\downarrow_1\downarrow_1\downarrow_0$ (for object), and the dative-case selects $\downarrow_1\downarrow_1\downarrow_1\downarrow_0$ (for indirect object). I assume only these three case specifications here, but the set could be enlarged (see Sect. 4.4.4).

(6.83) Entry of the unit ‘*no* + case particle’

```

IF      t
THEN    IF       $\varphi_E[(\alpha : e_s), (\beta : e)]$ 
        THEN    make/go(<L-1>); put(?t);
                make/go(<\downarrow_0>); put(U, ? $\exists x.Fo(x) \& [\alpha \star x] : e_s$ ); go(<\uparrow_0>);
                make/go(<\mu>); put( $\beta : e$ )
        ELSE    ABORT
ELSE    ABORT                                where  $\mu \in \{\downarrow_1\downarrow_0, \downarrow_1\downarrow_1\downarrow_0, \downarrow_1\downarrow_1\downarrow_1\downarrow_0, \dots\}$ 

```

This entry is not a mere stipulation. First, it is an extension of *no* (Chap. 4) and *no-wa* (Chap. 5): it blends the type-e node creation (i.e., part of the entry of *no*) with the pairing of two propositional trees (i.e., part of the entry of *no-wa*). Second, (6.83) has

the residue of the entry of case particles (i.e., the action to determine a node position in a local tree). Third, the fusion of *no* and a case particle is historically plausible: Kuroda (2005: 230, footnote 37) suggests that the fusion of *no* and a case particle yielded a number of clausal connectives such as *no-ni* (= ‘though’) through the use of HIRs.⁸

4.3 The Analysis

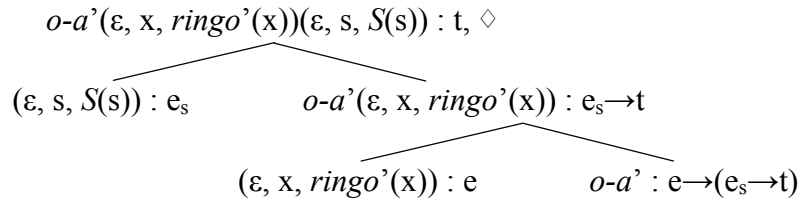
I shall now illustrate how the entry of ‘*no* + case particle’ (6.83) works by giving a tree transition for the simple HIR (6.84).⁹

(6.84) Head-Internal Relatives (HIRs)

[Ringo-ga oite-atta no]-o Kiki-ga tabe-ta.
 [apple-NOM place-existed NO]-ACC K-NOM eat-PAST
 ‘There was an apple and Kiki ate it.’

The parse of the HIR clause generates the tree (6.85), where *o-a’* is the content of *oite-atta*. The propositional formula at the top node is subject to QUANTIFIER EVALUATION (Q-EVALUATION; see Sect. 2.2, Chap. 3), as in (6.86).

(6.85) Parsing *Ringo-ga oite-atta*



⁸ One may object that the entry (6.83) stipulates the predicate $\star$. This is true, but this itself does not favour previous accounts over our account. Recall that previous studies devise a number of new theoretical artefacts. Kim (2007, 2008a, 2008b, 2009) is based on structural stipulations (e.g., the absence of TP in the HIR clause) and lexical ones (e.g., the denotation of *no*). Grosu (2010) and Grosu & Landman (2012) postulate the null category ChR and the feature [CR_{un}], both of which are solely specific to HIRs.

⁹ In this example, the HIR clause occurs sentence-initially. If the embedding clause occurs sentence-initially, GENERALISED ADJUNCTION will fire before parsing the HIR clause.

(6.86) Evaluating the proposition in (6.85)

$$S(b) \& [ringo'(a_b) \& o-a'(a_b)(b)]$$

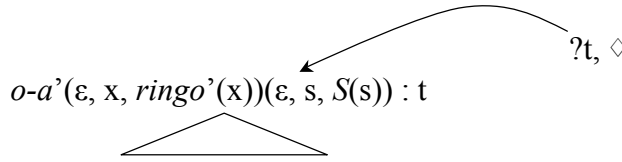
$$\text{where } b = (\epsilon, s, S(s) \& [ringo'(a_s) \& o-a'(a_s)(s)])$$

$$a_b = (\epsilon, x, ringo'(x) \& o-a'(x)(b))$$

$$a_s = (\epsilon, x, ringo'(x) \& o-a'(x)(s))$$

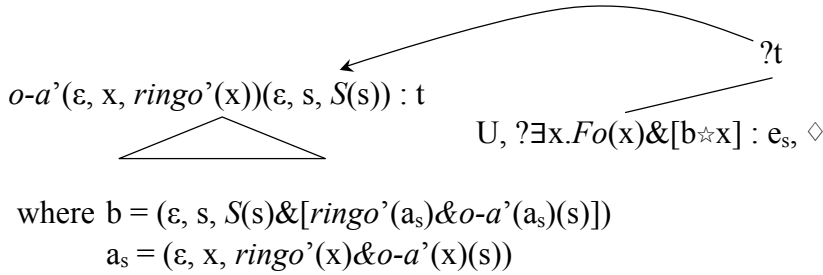
Now, the unit *no-o* drives a LINK transition in three steps: (i) the inverse LINK mapping (= (6.80)a), (ii) the construction of a situation node (= (6.80)b), and (iii) the construction of a node for an individual term (= (6.80)c). First, the current type-*t* node is inversely LINKed to a fresh type-*t*-requiring node.

(6.87) Part (i): the inverse LINK mapping



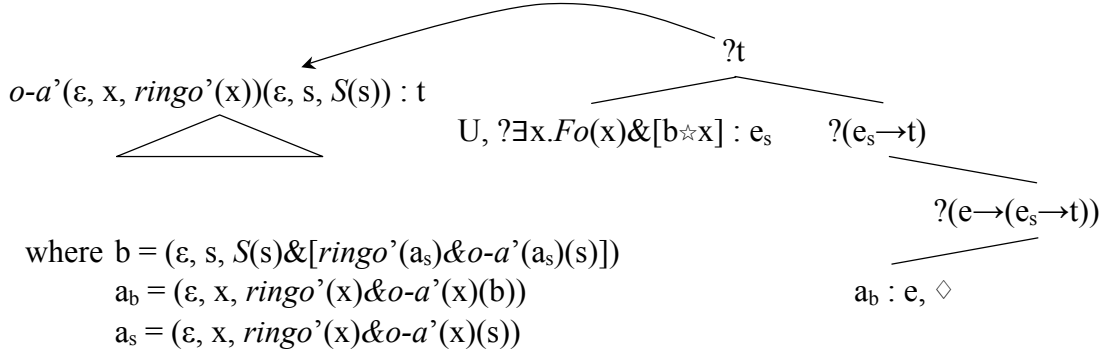
Second, the parser creates a situation node and posits the requirement that the term at this situation node will be in a ‘Relevancy’ relation (modelled by $\star$) to the situation term in the evaluated proposition of the HIR clause, namely the situation term *b* in (6.86). This requirement is written as $? \exists x. Fo(x) \& [b \star x]$.

(6.88) Part (ii): the construction of a situation node



Finally, the parser constructs a type-*e* node and annotates it with an individual term in the previous proposition. In the present case, the parser makes a copy of the term a_b in (6.86). Further, the accusative-case particle *o* identifies the current node as the object node in the unfolding tree, as in (6.89).

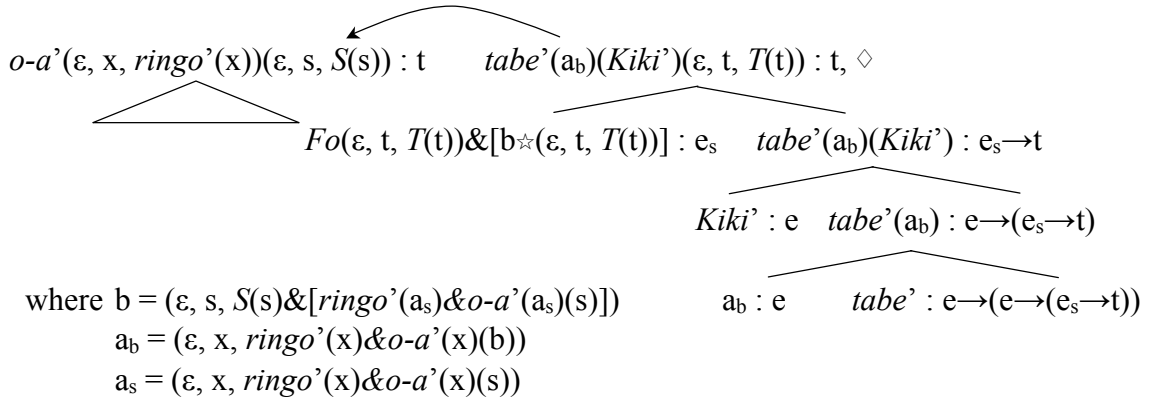
(6.89) Part (iii): the construction of a node for an individual term



This LINK transition has led to the **partial** tree where the situation node is decorated with the Relevancy requirement and the object node is decorated with the internal head of the HIR clause.

Within this partial tree, the matrix subject *Kiki-ga* creates a subject node, and the matrix verb *tabe-* (= ‘eat’) projects a propositional schema, where the situation and the object nodes harmlessly collapse with the pre-existing nodes in (6.89). In particular, the situation meta-variable U is substituted with $(\epsilon, t, T(t))$, which stands in the Relevancy relation $\star$ to the situation term b in the previous proposition. Finally, the tree is cleaned up by ELIMINATION (i.e., functional application, type deduction). The proposition at the top node represents the interpretation of (6.84): ‘There was an apple and Kiki ate it.’

(6.90) ELIMINATION (three times)



In this analysis, the dynamics of the HIR understanding in real time is directly captured: the processing of the HIR clause establishes a context with respect to which

the embedding clause is subsequently parsed in a fresh tree. This emergent tree is a rich environment, ‘rich’ in the sense that the tree has had internal structure partially when it is introduced. This partial tree models (if not explains) that only HIRs are subject to the Relevancy Condition and accounts for how the head is licensed by the embedding verb.

4.4 Consequences

Our account predicts a number of HIR properties. The organisation of this sub-section parallels that of Sect. 2: ‘indeterminacies’ (Sect. 4.4.1), ‘quantification’ (Sect. 4.4.2), ‘islands’ (Sect. 4.4.3), and ‘ambivalence’ (Sect. 4.4.4).

4.4.1 Indeterminacies

The lexical entry of the unit ‘*no* + case particle’ (6.83) straightforwardly accounts for the indeterminacy of the head. Consider (6.7), repeated here as (6.91).

(6.91) Head-Internal Relatives (HIRs)

Kiki-wa [Osono-ga Jiji-o oikaketeita no]-o tsukamaeta.

K-TOP [O-NOM J-ACC was.chasing NO]-ACC caught

- a. ‘Osono was chasing Jiji and Kiki caught her (= Osono).’
- b. ‘Osono was chasing Jiji and Kiki caught him (= Jiji).’
- c. ‘Osono was chasing Jiji and Kiki caught them (= Osono and Jiji).’

According to the entry of ‘*no* + case particle’ (6.83), the content of the head must be a term within the previous proposition but it is **not** specified in advance. Thus, the parser chooses a relevant term pragmatically. If *Osono* is selected, the interpretation (6.91)a follows. If *Jiji* is selected, the interpretation (6.91)b follows.

In the split head (6.91)c, there are technical issues to be sorted out. In the remit of the current DS framework, plurality is not fully expressible. Pending a full analysis, I simply assume the plurality operator $\oplus$ which groups atomic individuals together as the

sum, which itself is of type-e (cf., Link 1983). For instance, a group of Osono and Jiji is represented as the type-e term (6.92).

$$(6.92) \quad (Osono') \oplus (Jiji')$$

I shall assume that the operation to construct a group term applies to the evaluated proposition of the HIR clause. If this assumption is licit, the reading (6.91)c follows when the parser makes a copy of the group term (6.92) at the object node created by the unit *no-o*. Although the detail needs to be worked out, the split head is in principle accounted for along with the lines suggested here.¹⁰

According to the entry of ‘*no* + case particle’ (6.83), the content of a head is an **evaluated** term. Owing to QUANTIFIER EVALUATION (Sect. 2.2, Chap. 3), an evaluated term reflects the **full** content of the HIR clause. This accounts for (6.64), repeated here as (6.93), for which Kim (2007) imposed additional conditions on the usual E-type device (Sect. 3.3). If the content of a head were simply the term of *insei* (= ‘graduate students’), the non-absurd reading should be possible: ‘Graduate students were writing term papers at home.’ This reading is absent, however. This is because the content of the head is not the term of *gakusei* but the evaluated term that denotes 3 graduate students who came to the party on Saturday. This necessarily gives rise to the absurd reading: ‘3 graduate students came to the party on Sunday and they were also writing term papers at home.’

¹⁰ Chap. 4 (Sect. 4.2) noted that participant nominalisation with the coordinate pre-*no* clauses did not have a group reading (i.e., the reading where distinct individuals satisfy different aspects of the pre-*no* clauses). In this case, the group reading is unavailable as the clause-linking suffix *i* forces the sharing of the content of a gap (see Sect. 4.3, Chap. 4).

(6.93) Head-Internal Relatives (HIRs)

#[Choudo-3-nin-no insei-ga doyoubi-no-paatii-ni kita no]-ga
 [exactly-3-CL-GEN grad.student-NOM Saturday-GEN-party-to came NO]-NOM
 jitsuwa uchi-de taamupeipaa-o kaiteita.
 in.fact home-at term.paper-ACC were.writing
 ‘Exactly 3 graduate students came to the party on Saturday, and in fact they (= these 3 students) were writing term papers at home.’

The entry of ‘*no* + case particle’ requires that the content of a head should be a term **within** the evaluated proposition of the HIR clause (or, possibly, a group term to be created by applying the plurality operation to the evaluated proposition). With this in mind, consider (6.61), repeated here as (6.94), which led Kim (2007) to enrich the usual E-type device with additional constraints (Sect. 3.3). In (6.94), the content of the head cannot be a group term that denotes a group of newspapers and books. This is predicted in our analysis; such group terms cannot be constructed on the basis of the evaluated proposition of the HIR clause itself.

(6.94) Head-Internal Relatives (HIRs)

*Hanako-ga hon-o 3-satsu katte-kita. Taro-wa [Hanako-ga mata
 H-NOM book-ACC 3-CL buy-came T-NOM [H-NOM also
 dono shinbun-mo katte-kita no]-o tana-ni narabeta.
 every newspaper-also buy-came NO]-ACC shelf-on placed
 Int. ‘Hanako bought (and brought) 3 books. Hanako also bought (and brought) every newspaper and Taro shelved them (= books and newspapers).’ (The HIR string is acceptable if the internal head is *shinbun* alone.)

There are two apparent counterexamples to this analysis. First, Kim & Chung (2003) point out that the head in Korean HIRs may be covert. As attested in (6.60), repeated here as (6.95), this is true of Japanese. It looks as though the evaluated proposition of the HIR clause does not contain a term that denotes dirt. Kurosawa (2003), however, suggests that a resulting state of the eventuality of an HIR clause may be represented

within a situation term. The details need to be explored, but it is licit to hold that the term denoting a **resulting** state of the HIR clause in (6.95) represents a situation where the accumulated dirt is a participant. The parser then copies the individual term denoting the dirt in this representation.

(6.95) Head-internal Relatives (HIRs)

?John-wa [zubon-ga yogore-ta no]-o fukitotta.

J-TOP [pants-NOM get.dirty-PRF NO]-ACC wiped.from

‘The pants got dirty and John wiped the dirt from the pants.’

Second, Shimoyama (2001: 143) cites (6.96), which seems to show that the genitive-case marked expression cannot be a head. But the unacceptability of (6.96) may be due to the Relevancy Condition: each host’s mother is not directly engaged in the sushi-serving event. Furthermore, the praising event could apply to both mothers and sushi. Grosu & Landman (2012: 171) show that if these factors are controlled, the genitive-case marked head is possible, as in (6.97).

(6.96) Head-Internal Relatives (HIRs)

*Dono-hosuto-mo [[soitsuno-hahaoya-no sushi]-o

which-host-too [[his-mother-GEN sushi]-ACC

dashita no]-o suguni hometa.

served NO]-ACC immediately praised

‘Every host served his mother’s sushi and praised her immediately.’

(6.97) Head-Internal Relatives (HIRs)

Dono-otokonohito-mo [[daidokoro-no jibun-no tsuma-no sushi]-o

which-man-too [[kitchen-GEN self-GEN wife-GEN sushi]-ACC

kyaku-ni dashita no]-o kyaku-ga suguni yon-de hometa.

guest-DAT served NO]-ACC guest-NOM immediately call-and praise

‘Every man served to the guest the sushi of his wife, who was in the kitchen, and the guest called and praised her immediately after that.’

Let us turn to another indeterminacy of HIRs: the relation between the HIR and the embedding clause. According to the lexical entry of ‘*no* + case particle’ (6.83), the situation node is decorated with $?\exists x.Fo(x) \& [\alpha \star x]$. This requires the situation term α in the evaluated proposition of the HIR clause to be related to the situation term in the tree of the embedding clause in a **pragmatically specified** fashion (e.g., temporal, causal), modulo the Relevancy restriction $\star$. In (6.8), repeated here as (6.98), the HIR and the embedding clauses may be associated in a temporally contiguous way. At the same time, the Relevancy restriction $\star$ precludes temporally-disjoint readings such as: ‘Jiji is now running, and Kiki called Jiji to stop in the past when Jiji was, say, walking.’

(6.98) Head-Internal Relatives (HIRs)

Kiki-wa [Jiji-ga hashi-tteiru no]-o yobitomete.
 K-TOP [J-NOM running-PROG NO]-ACC called.to.stop
 ‘Kiki called Jiji to stop **while** he was running.’

The temporally-distinct reading is possible in the HER counterpart (6.10), repeated here as (6.99). This is because the Relevancy restriction $\star$ is written in to the unit ‘*no* + case particle’ but not to any expressions in HERs.

(6.99) Head-External Relatives (HERs)

- Kiki-wa [(ima) hashitteiru Jiji]-o (mukashi) yobitomete.
 K-TOP [(now) be.running J]-ACC (in.the.past) called.to.stop
- a. ‘Kiki called Jiji, who was running, to stop.’
 - b. ‘Kiki has called Jiji, who is now running, to stop in the past.’

At present, $\star$ does not spell out the Relevancy Condition. But it provides a basis for modelling how only HIRs are subject to the condition. A research avenue is to substantiate $\star$ by representing aspect and tense in situation terms (Cann 2011; see also Sect. 4.4.1, Chap. 4). Still, even in this present form, the entry of ‘*no* + case particle’

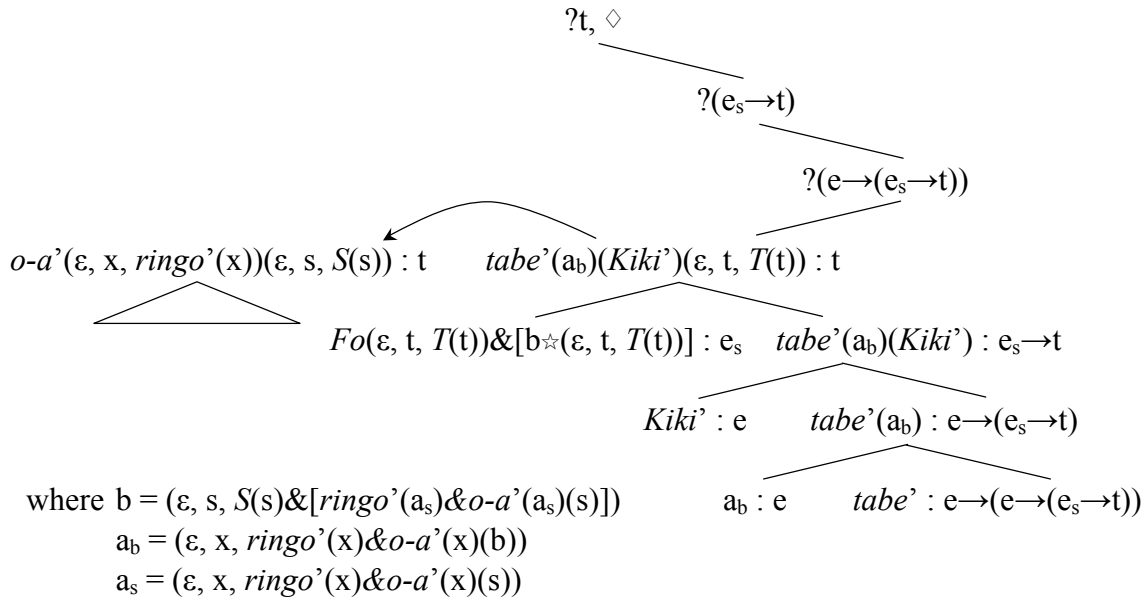
makes a novel prediction: the condition holds only of the HIR and the **immediately** embedding clauses. Consider (6.100).

(6.100) Head-Internal Relatives (HIRs)

[[Ringo-ga oite-atta no]-o Kiki-ga tabeta to] Osono-ga itta.
 [[apple-NOM place-existed NO]-ACC K-NOM ate COMP] O-NOM said
 ‘Osono said that [there was an apple and Kiki ate it].’

In this case, once the inversely LINKed type-*t*-requiring node is fully developed, the sentential complementiser *to* marks it as an object node within a new structure, as in (6.101) (see Sect. 4.2, Chap. 3 for the complementiser *to*).

(6.101) Parsing (6.100) up to the sentential complementiser *to*



The Relevancy predicate $\star$ associates the situation term b in the previous proposition with the situation term in the structure of the immediately embedding clause, but not the matrix clause. In fact, the HIR (6.100) may have the reading: ‘There was an apple, and Kiki ate it. Then, 3 years later, Osono talked about it.’ In other words, the Relevancy Condition operates **locally**, holding only of the HIR and the immediately embedding clauses, excluding any higher clauses.

Is this generalisation expressible in previous studies? In the ChR account, the Relevancy Condition is ‘essentially pragmatic’ (Sect. 3.4). Thus, it is not clear how the structural locality between the HIR and the immediately embedding clauses could be formulated. In Kim’s E-type analysis, the HIR clause moves up and adjoins to a higher AspP (Sect. 3.3). Thus, it must be assumed that the HIR clause does not move over the AspP for *Kiki-ga tabeta* in (6.100). It may be possible to justify this assumption in terms of computational efficiency (e.g., Minimal Link Condition (Chomsky 1995)), but no such justification is as yet offered; and the resulting account, if any, will resort to the movement operation that lacks independent motivations other than the semantic-type mismatch (see the discussion about (6.55)).

The upshot is that various indeterminacies of HIRs and restrictions on them (e.g., (6.93), (6.94), locality in the Relevancy Condition) fall out from the interplay of grammatical constraints (e.g., Q-EVALUATION, the entry of ‘*no* + case particle’) and pragmatic inference.

4.4.2 Quantification

A quantified head within the HIR clause does not interact with quantifiers within the embedding clause (Shimoyama 1999, 2001; see Sect. 2.3). In (6.14), repeated here as (6.102), the quantified head *dono-hon-mo* (= ‘every book’) in the HIR clause cannot take scope over the quantified expression *hotondo-no-gakusei* (= ‘most students’) in the embedding clause, though the former linearly precedes the latter.

(6.102) Head-Internal Relatives (HIRs)

[Dono-hon-mo tsukue-no-ue-ni oite-atta no]-o
 [every-book-too table-GEN-top-at place-existed NO]-ACC
 hotondo-no-gakusei-ga yonda.
 most-GEN-student-NOM read
 ‘Every book was on a table, and most students read it.’
 (i) *‘every’ > ‘most’ (ii) ‘most’ > ‘every’

In our analysis, the HIR clause is treated as a clausal adverb modifying the immediately embedding clause in that the propositional tree of the HIR clause is **subordinate** to an emergent propositional tree **through** a LINK relation. As envisaged, the same scope pattern is observed in the clausal adverb counterpart (6.103).

(6.103) Clausal adverbs

[Dono-hon-mo tsukue-no-ue-ni oite-atta toki]
 [every-book-too table-GEN-top-at place-existed when]
 hotondo-no-gakusei-ga yonda.
 most-GEN-student-NOM read
 ‘When every book was on a table, most students read it.’
 (i) *‘every’ > ‘most’ (ii) ‘most’ > ‘every’

By contrast, the quantified head in HERs interacts with quantified expressions in the embedding clause, as in (6.13), repeated here as (6.104).

(6.104) Head-External Relatives (HERs)

[Tsukue-no-ue-ni oite-atta dono-hon]-mo
 [table-GEN-top-at place-existed every-book]-too
 hotondo-no-gakusei-ga yonda.
 most-GEN-student-NOM read
 ‘Every book that was on a table, most students read it.’
 (i) ‘every’ > ‘most’ (ii) *‘most’ > ‘every’

Note that the head in HERs is **part** of the matrix clause. So, the scope phenomenon in (6.104) is just the usual pattern associated with ‘every’ and ‘most’ in Japanese: that is, the linearly preceding item out-scopes the ensuing items within the same clause. In fact, the scope fact in (6.104) remains the same in (6.105), where the relative clause modifier has been dropped.

- (6.105) *Dono-hon-mo hotondo-no-gakusei-ga yonda.*
 every-book-too most-GEN-student-NOM read
 ‘Every book, most students read it.’
 (i) ‘every’ > ‘most’ (ii) *‘most’ > ‘every’

Next, consider maximality in (6.15), repeated here as (6.106). In our analysis, the maximality effect is not encoded in the entry of ‘*no* + case particle’ (6.83). Rather, the effect is derived **pragmatically**. This leads us to expect that the effect is not specific to HIRs and it may also be found in a clausal adverb counterpart of (6.106). This is confirmed by (6.107). Suppose that 8 apples were on a table; though the demonstrative *sorera* (= ‘those’) is compatible with any integer from 2 to 8, the most accessible interpretation is that Kiki ate all of these 8 apples.

(6.106) Head-Internal Relatives (HIRs)

Kiki-wa [hotondo-no ringo-ga tsukue-no-ue-ni oite-atta
 K-TOP [most-GEN apple-NOM table-GEN-top-at place-existed
no]-o tabeta.
 NO]-ACC ate
 ‘Most apples were on a table and Kiki ate them.’

(6.107) Clausal adverbs

[*Hotondo-no ringo-ga tsukue-no-ue-ni oite-atta toki*]
 [most-GEN apple-NOM table-GEN-top-at place-existed when]
Kiki-wa sorera-o tabeta.
 K-TOP those-ACC ate
 ‘When most apples were on a table, Kiki ate all of them.’

Therefore, whatever pragmatic principle (e.g., the Principle of Relevance (Sperber & Wilson 1995)) underlies the maximality effect in the clausal adverb (6.107) is operative in the HIR (6.106) as well. This pragmatic account is applicable to examples where the effect is absent, as in (6.17), repeated here as (6.108). What inhibits the maximality

effect in (6.108) is the contextual information that the ticket checker takes no more than one ticket at a time (Kubota & Smith 2007).

(6.108) Head-Internal Relatives (HIRs)

Dono-joukyaku-mo [saifu-ni kaisuuken-ga haitteita no]-o
 every-passenger-too [wallet-in coupon.ticket-NOM was.present NO]-ACC
 toridashite kaisatsu-ni ireta.
 take.out ticket.checker-to put

‘Every passenger took out a coupon ticket that she/he had in (her/his) wallet and put it in the ticket checker.’ (adapted from Kubota & Smith (2007: 154))

Finally, the issue of negation. As observed in Sect. 2.3, negation is licensed in the HIR clause if the existence of the denotation of the head is inferrable. In (6.19), repeated here as (6.109), it may be pragmatically inferred that the wallet was not in a safe but it was kept in another place, say, on a table near the safe.

(6.109) Head-Internal Relatives (HIRs)

Dorobou-wa [saifu-ga kinko-ni haittei-naka-tta no]-o
 thief-TOP [wallet-NOM safe-at put.inside-NEG-PAST NO]-ACC
 manmato nusumi-dashita.
 successfully steal-took.away

‘A wallet was not inside a safe (but outside the safe), and a thief successfully stole it.’

Negation has not been covered within DS, but it is far from contentious that the negator interacts with quantifiers to determine the scope. In (6.109), QUANTIFIER EVALUATION may yield an evaluated proposition where the term of *saifu* (= ‘a wallet’) takes scope over the negator. The parser then manipulates this term in enriching the structure of the immediately embedding clause.

4.4.3 Islands

Another fact about the entry of ‘*no* + case particle’ (6.83), repeated here as (6.110), is that it predicts that HIRs are not sensitive to islands. In (6.110), β is a term of the head, and crucially there is **no** structural restriction on where β is detected in the evaluated proposition φ_E . Thus, the parser may make a copy of the term β even if β corresponds to a node position within an island (in DS terms, within a LINKed tree).

(6.110) Entry of the unit ‘*no* + case particle’

```

IF      t
THEN    IF       $\varphi_E[(\alpha : e_s), (\beta : e)]$ 
        THEN    make/go(<L-1>); put(?t);
                make/go(<↓0>); put(U,  $\exists x.Fo(x) \& [\alpha \star x] : e_s$ ); go(<↑0>);
                make/go(<μ>); put(β : e)
        ELSE    ABORT
ELSE    ABORT                                where  $\mu \in \{\downarrow_1 \downarrow_0, \downarrow_1 \downarrow_1 \downarrow_0, \downarrow_1 \downarrow_1 \downarrow_1 \downarrow_0, \dots\}$ 

```

To corroborate this prediction, (6.20) and (6.22) in Sect. 2.4 are repeated here as (6.111) and (6.112). The former shows that the HIR is not sensitive to the Complex NP island (i.e., the NP headed by *houkoku* (= ‘report’)), while the latter shows that the HIR is not sensitive to the *wh*-island (i.e., the *wh*-clause formed by *itsu* (= ‘when’)).

(6.111) Head-Internal Relatives (HIRs) involving a complex NP

[Taro-ga [Hanako-ga subarashii ronbun-o kaita toiu houkoku]-o
 [T-NOM [H-NOM excellent paper-ACC wrote TOIU report]-ACC
 kiiteita no]-ga tsuini shuppansareta.
 has.heard NO]-NOM finally was.published
 ‘Taro has heard a report that Hanako wrote an excellent paper, and the paper
 was finally published.’

(6.112) Head-Internal Relatives (HIRs) involving a *wh*-island

[John-ga Tom-ni [Mary-ga itsu ronbun-o shiageru ka]

[J-NOM Tom-DAT [M-NOM when paper-ACC finish Q]

kiiteita no]-ga shuppansareta.

had.asked NO]-NOM was.published

‘John had asked Tom when Mary finished a paper and it was published.’

4.4.4 Ambivalence

Our account of HIRs, which relates a propositional node to another propositional node, models the non-nominality of HIRs straightforwardly. The entry of ‘*no* + case particle’ corresponds to the native speaker’s intuition that the HIR clause provides background information against which the embedding clause is interpreted (Kim 2008b). However, HIRs also exhibit nominal features. Thus, the challenge to our account is how to derive the nominal nature of HIRs. I shall first address the non-nominal properties and then the nominal properties.

Consider the FR (6.23), repeated here as (6.113), where the *no*-part denotes a human with an expressive connotation (Sect. 3.4.2, Chap. 4). In our analysis of HIRs, the sequence ‘*no* + case particle’ constitutes a unit, and thus the particle *no*, a locus of connotation in FRs, no longer stands as a separate item.¹¹ Thus, the lack of connotation in the HIR (6.24), repeated here as (6.114), is expected.

(6.113) Free Relatives (FRs)

Kiki-wa [naita no]-o nagusameta.

K-TOP [cried NO]-ACC consoled

‘Kiki consoled a person who cried.’

¹¹ Seraku (2013a) observes that the compound particle *toiuno* (Sect. 3, Chap. 2), which was derived from *no*-nominalisation, does not exhibit an expressive connotation even if it denotes a human. It is claimed that the nominaliser *no*, a locus of a connotation, is not recognised as a separate element in *toiuno*.

(6.114) Head-Internal Relatives (HIRs)

Kiki-wa [tomodachi-ga naita no]-o nagusameta.

K-TOP [friend-NOM cried NO]-ACC consoled

‘A friend cried and Kiki consoled her/him.’

The nominaliser *no* in FRs (or participant nominalisation) inversely LINKs a propositional tree to a **type-e** node (Sect. 3, Chap. 4). On the other hand, the unit ‘*no* + case particle’ in HIRs inversely LINKs a propositional tree to a **type-t**-requiring node. Given the assumption that modifiers such as demonstratives and relative clauses only modify nominal expressions, the ungrammaticality of the HIRs (6.26) and (6.28), repeated as (6.115) and (6.116), is accounted for. The latter is particularly interesting since the impossibility of relative clause stacking in HIRs is correctly predicted.

(6.115) Head-Internal Relatives (HIRs)

*Jiji-wa sono [Kiki-ga pan-o katteoita no]-o tabeteshimatta.

J-TOP that [K-NOM bread-ACC had.bought NO]-ACC has.eaten

‘Kiki had bought that bread and Jiji has eaten it.’

(6.116) Head-Internal Relatives (HIRs)

*Jiji-wa [Kiki-ga tanoshiminishteita, [Osono-ga keiki-o katte-kita

J-TOP [K-NOM looked.forward.to [O-NOM cake-ACC buy.came

no]]-o tabeteshimatta.

NO]]-ACC has.eaten

Int. ‘Osono bought cake which Kiki looked forward to, and Jiji has eaten it.’

The same form of analysis is applicable to the infelicity of (6.30), reproduced here as (6.117), as an answer to the *wh*-question asking about an individual: *Who did Kiki console?* This is because ‘*no* + case particle’ in HIRs does not map a propositional tree onto a **type-e** node.

(6.117) Head-Internal Relatives (HIRs)

#Kiki-wa [tomodachi-ga naita no]-o nagusameta.

K-TOP [friend-NOM cried NO]-ACC consoled

Int. ‘Kiki consoled a friend who cried.’ (unacceptable as an answer to ‘Who did Kiki console?’)

Similarly, HIRs cannot be at a focus position in clefts since the node created by the unit ‘*no* + case particle’ in HIRs is a type-t-requiring node while the focus position in clefts is a type-e node (Sect. 5, Chap. 2). This is shown in (6.32), repeated as (6.118).

(6.118) Head-Internal Relatives (HIRs) in clefts

*[Kiki-ga todoketa no]-wa [Osono-ga pan-o yaita no] da.

[K-NOM delivered NO]-TOP [Osono-NOM bread-ACC baked NO] COP

Int. ‘It is Osono’s-baked bread that Kiki delivered.’

The non-nominality of HIRs has been reflected in our analysis. Then, how to derive the nominality of HIRs? First, Sect. 2.5.2 pointed out that the *no*-part may be marked by a range of case particles such as the directive-case particle (or, possibly, a postposition) *e* (= ‘to’), as illustrated in (6.36), repeated here as (6.119).

(6.119) Head-Internal Relatives (HIRs)

Ojisan-wa [Kiki-ga hashitte-kita no]-e dekkiburashi-o tewatashita.

old.man-TOP [K-NOM run-came NO]-to deck.brush-ACC handed

‘Kiki came running and an old man handed a deck brush to her.’

The flexibility of case-marking is congruent with the entry of ‘*no* + case particle’ (6.83), repeated here as (6.120).

(6.120) Entry of the unit ‘*no* + case particle’

```

IF      t
THEN    IF       $\varphi_E[(\alpha : e_s), (\beta : e)]$ 
        THEN    make/go(<L-1>); put(?t);
                make/go(<↓0>); put(U,  $\exists x.Fo(x) \& [\alpha \star x] : e_s$ ); go(<↑0>);
                make/go(<μ>); put(β : e)
        ELSE    ABORT
ELSE    ABORT

```

where $\mu \in \{\downarrow_1\downarrow_0, \downarrow_1\downarrow_1\downarrow_0, \downarrow_1\downarrow_1\downarrow_1\downarrow_0, \dots\}$

In this entry, the proper LOFT relation μ must be selected from the set $\{\downarrow_1\downarrow_0, \downarrow_1\downarrow_1\downarrow_0, \downarrow_1\downarrow_1\downarrow_1\downarrow_0, \dots\}$. The selection is guided by the case particle with which *no* forms a unit. For example, in the unit *no-ga*, where *ga* is a nominative-case particle, μ is realised as $\downarrow_1\downarrow_0$, indicating a subject position. In (6.120), only three LOFT relations are spelt out: (i) $\downarrow_1\downarrow_0$ (for subject), (ii) $\downarrow_1\downarrow_1\downarrow_0$ (for object), and (iii) $\downarrow_1\downarrow_1\downarrow_1\downarrow_0$ (for indirect object). But this set may be enlarged. For example, if *no* is combined with the particle *e* (= ‘to’), μ may be $\downarrow_1\downarrow_1\downarrow_1\downarrow_1\downarrow_0$, a LOFT relation for a type-*e* adjunct node (Marten 2002); see the discussion of adjuncts in Sect. 3.4.6, Chap. 4. Thus, the entry (6.120) is consistent with the case-marking fact of HIRs.

Second, the nominal status of HIRs seems to be indicated by ‘quantifier float’ (Sect. 2.5.2). To recapitulate, consider (6.121), which is a slightly modified version of (6.42).¹² Given that quantifier float is clause-bounded, it is implausible to assume that the numeral quantifier *mittsu* (= ‘3’) is floated out of the HIR clause; rather, it looks as though the case-marked *no*-part stands as a host NP for *mittsu*.

(6.121) Head-Internal Relatives (HIRs) with a numeral quantifier

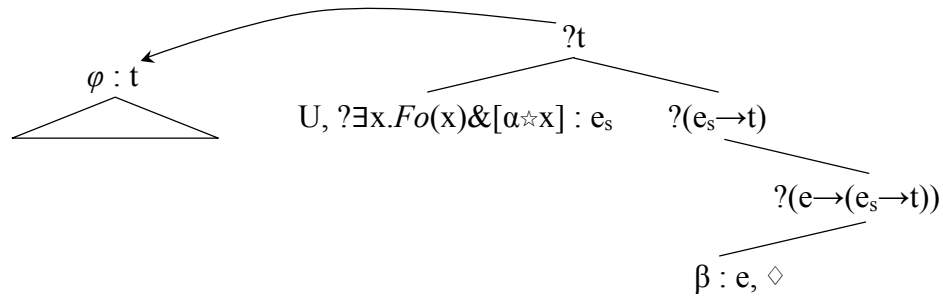
[Pan-ga teiburu-ni oiteatta no]-o Kiki-ga mittsu tabe-ta.
 [bread-NOM table-on place.existed NO]-ACC K-NOM 3 eat-PAST
 ‘Some bread was on a table and Kiki ate 3 slices.’

But the current example does not necessarily show that HIRs have nominal status. The unit ‘*no* + case particle’ maps the tree of the HIR clause onto that of the embedding

¹² The matrix subject *Kiki* is placed after the HIR clause for ease of illustration.

clause, where the evaluated term of the head *pan* (= ‘bread’) is copied at an object node (as designated by the accusative-case particle *o*). It is **within** this inversely LINKed tree that the numeral quantifier *mittsu* is licensed. Schematically, the parse of (6.121) up to *no-o* engenders the tree (6.122).

(6.122) Parsing (6.121) up to *no-o*



where α is the situation term in the evaluated proposition of the HIR clause

β is the evaluated term of the head

φ is the proposition of the HIR clause

The evaluated content of the head *pan* (= ‘bread’), namely β , is present in the tree of the matrix clause. So, if the numeral quantifier *mittsu* is parsed, it is locally licensed by the term β within the same structure.

One might object that even though *mittsu* is licensed by the evaluated content of *pan*, it is not licensed by the **overt** NP. This objection is not sustainable. As evinced in (6.123), *mittsu* is licensed even when the host NP is missing. In our analysis, *tabe-* (= ‘eat’) projects a propositional template, where the object node is decorated with a meta-variable. This meta-variable is saturated as a term denoting a salient object, say, bread. The numeral quantifier *mittsu* is then licensed by this term over the semantic tree.

(6.123) Numeral quantifier without a host NP

Kiki-ga mittsu tabe-ta.

K-NOM 3 eat-PAST

‘Kiki ate 3 of something (e.g., slices of bread).’

Finally, ‘secondary predicates’ seem to show that HIRs behave as a nominal (Sect. 2.5.2). A relevant example is (6.124), a slightly modified version of (6.45).¹³ Given the locality constraint on secondary predicates, it is not legitimate to hold that the secondary predicate *nama-de* (= ‘raw’) is licensed by the head *sakana* (= ‘fish’); rather, it looks as if the case-marked *no*-part serves as a nominal that licenses *nama-de*.

(6.124) Secondary predicate in HIRs (Head-Internal Relatives)

[Sakana-ga teiburu-ni oite-atta no]-o Jiji-ga nama-de tabeta.
 [fish-NOM table-on place-existed NO]-ACC J-NOM raw ate
 ‘Fish was placed on a table, and Jiji ate it raw.’

The above example, however, does not necessarily indicate the nominality of HIRs. In our analysis, the parse of (6.124) up to the unit *no-o* drives a LINK transition, where the tree of the HIR clause is inversely LINKed to the tree of the embedding clause. As schematically displayed in (6.122) above, the tree of the embedding clause contains a node which is decorated with the evaluated term of the head (in the present case, *sakana* (= ‘fish’)). This evaluated term licenses the content of the secondary predicate *nama-de* within the local tree.

For completeness, note that *nama-de* may be licensed even when a host NP is not present in the surface sequence as in (6.125). This is expected in our analysis; *tabe-* (= ‘eat’) generates a propositional schema, where the object node is decorated with a meta-variable. If this meta-variable is saturated as a term that denotes fish, it licenses *nama-de* over the semantic structure.

(6.125) Secondary predicate without a host NP

Jiji-ga nama-de tabe-ta.
 J-NOM raw eat-PAST
 ‘Jiji ate something raw.’

¹³ The matrix subject *Jiji* is placed after the HIR clause for expository purposes.

The dynamic account thus captures the (putatively) ambivalent nature of HIRs. First, the non-nominality readily stems from the mapping from a propositional tree onto another propositional tree. Second, the nominality of HIRs is illusory. The case-marking of HIRs does not contradict our analysis because it is the very amalgam ‘*no* + case particle’ that drives a LINK transition for the HIR processing. The fusion of *no* and a wide range of particles is accredited by the flexibility of LOFT relations. The locality restrictions on ‘quantifier float’ and ‘secondary predicates,’ which initially suggested the nominality of HIRs, are recast as restrictions on the semantic tree of the embedding clause. This tree contains a node for the head, where the content of the head licenses the content of a numeral quantifier or that of a secondary predicate.

4.5 Summary

HIRs are properly handled by combining the instructions encoded in the nominaliser *no* (Chap. 4) and the cleft marker *no-wa* (Chap. 5). The resulting action package, stored in the unit ‘*no* + case particle,’ maps the propositional tree of the HIR clause onto an emergent tree of the embedding clause. This emergent tree has been partially articulated when it is induced. First, the partial tree contains a situation node; this comes with the requirement that the situation term will be in a Relevancy relation (as specified by the predicate ☆) to the situation term in the evaluated proposition of the HIR clause. This makes sure that HIRs (but not other kinds of relatives) are governed by the Relevancy Condition. Second, the partial tree also contains a node for an internal head, together with pragmatic leeway to choose a term that denotes a salient individual (or a group of individuals). This ensures that the head, though internal to the relative clause, serves as an argument of the embedding verb. The dynamic account elucidates a number of HIR characteristics, some of which raise a problem for recent accounts of HIRs (e.g., maximality, negation, island-insensitivity, locality in the Relevancy Condition).

5 Change Relatives

5.1 Introduction

This section delves into another, much less investigated type of relatives: Change Relatives (CRs). Consider (6.4), repeated here as (6.126). In CRs, the *no*-part denotes the state of the individual as a result of the change described in the relative clause.

(6.126) Change Relatives (CRs)

[Otamajakushi-ga kaeru-ni natta no]-ga niwa-o haneteiru.
 [tadpole-NOM frog-COP became NO]-NOM garden-in is.hopping
 ‘A frog which is the result of changing from a tadpole is hopping in the garden.’ (Tonosaki 1998: 144)

The surface properties of CRs are almost identical to that of HIRs: (i) the head *kaeru* (= ‘frog’) is found inside the relative clause, (ii) the relative clause lacks a gap, and (iii) the relative clause is followed by the particle *no*. These affinities have led most previous studies (e.g., Hoshi 1995, Kim 2007, Grosu 2010) to lump CRs and HIRs together and apply the same form of analysis to them. The goal of this section is to show that (i) CRs and HIRs are radically different (Tonosaki 1996, 1998) and that (ii) CRs are modelled by the nominaliser *no* (rather than ‘*no* + case particle’).

Sect. 5.2 observes that unlike HIRs, CRs display only nominal features, and Sect. 5.3 notes that unlike HIRs, the Relevancy Condition is inactive in CRs. Based on these two arguments, Sect. 5.4 claims that (i) CRs are best analysed by the nominaliser *no* and that (ii) this clarifies the classification of *no*-nominalising phenomena. Finally, Sect. 5.5 summarises the main results.

5.2 Nominality of CRs

Tonosaki (1996, 1998) argues that despite the surface similarities between CRs and HIRs, CRs behave more like FRs (Free Relatives). In Sect. 2.5, we saw that FRs show

nominal characteristics whereas HIRs indicate non-nominality (as well as putatively nominal characteristics). Therefore, Tonosaki's claim is understood to mean that CRs have **nominal** status.

Tonosaki draws out several parallelisms between CRs and FRs, not shared by HIRs.¹⁴ The most convincing set of data is the following. As in FRs, but unlike HIRs, the demonstrative *sono* (= 'that') may modify CRs. Consider the FR (6.127) (originally, (6.25)), the HIR (6.128) (originally, (6.26)), and the CR (6.129).

(6.127) Free Relatives (FRs)

Jiji-wa sono [Kiki-ga katteoita no]-o tabeteshimatta.
 J-TOP that [K-NOM had.bought NO]-ACC has.eaten
 'Jiji has eaten that thing which Kiki had bought.'

(6.128) Head-Internal Relatives (HIRs)

*Jiji-wa sono [Kiki-ga pan-o katteoita no]-o tabeteshimatta.
 J-TOP that [K-NOM bread-ACC had.bought NO]-ACC has.eaten
 'Kiki had bought that bread and Jiji has eaten it.'

(6.129) Change Relatives (CRs)

Sono [otamajakushi-ga kaeru-ni natta no]-ga niwa-o haneteiru.
 that [tadpole-NOM frog-COP became NO]-NOM garden-in is.hopping
 'That frog which is the result of changing from a tadpole is hopping in the garden.'

The modifiability argument also holds of the relative clause modifier. In this respect, too, FRs and CRs are grouped together excluding HIRs. Consider the FR (6.130) (originally, (6.27)), the HIR (6.131) (originally, (6.28)), and the CR (6.132).

¹⁴ CRs lack an expressive connotation (Tonosaki 1996, 1998). In this respect, CRs behave more like HIRs than FRs (see Sect. 3.4.2, Chap. 4). This is a remaining problem for future research.

(6.130) Free Relatives (FRs)

Jiji-wa [Kiki-ga tanoshiminishiteita, [Osono-ga katte-kita no]]-o
 J-TOP [K-NOM looked.forward.to [O-NOM buy.came NO]]-ACC
 tabeteshimatta.
 has.eaten
 ‘Jiji has eaten that thing which Osono bought, which Kiki looked forward to.’

(6.131) Head-Internal Relatives (HIRs)

*Jiji-wa [Kiki-ga tanoshiminishiteita, [Osono-ga keiki-o katte-kita
 J-TOP [K-NOM looked.forward.to [O-NOM cake-ACC buy.came
 no]]-o tabeteshimatta.
 NO]]-ACC has.eaten
 Int. ‘Osono bought cake which Kiki looked forward to, and Jiji has eaten it.’

(6.132) Change Relatives (CRs)

Jiji-wa [niwa-o haneteiru, [otamajakushi-ga kaeru-ni natta
 J-TOP [garden-in is.hopping [tadpole-NOM frog-COP became
 no]]-o tsukamaeta.
 NO]]-ACC caught
 ‘Jiji caught a frog which is the result of changing from a tadpole and which is
 hopping in the garden.’

The translational convention for CRs is also telling. HIRs are generally translated into English by using clausal coordination, while CRs are not, as in the above examples.

In what follows, I shall offer additional data not discussed in Tonosaki (1996, 1998). As in the case of FRs, but unlike HIRs, CRs may provide an appropriate answer to *wh*-questions asking about individual entities (see Matsuda 2002: 635-6 for HIRs). For instance, the *wh*-question *What did Kiki catch?* may be appropriately answered by the FR (6.133) and the CR (6.135), but not by the HIR (6.134). In (6.135), the intended situation is the one where the speaker points to a tadpole in a picture and explains that what Kiki caught is the frog which is the result of changing from the tadpole.

(6.133) Free Relatives (FRs)

Kiki-wa [niwa-de haneteita no]-o tsukamaeta-yo.

K-TOP [garden-at was.hopping NO]-ACC caught-SFP

‘Kiki caught the one which was hopping in the garden.’

(6.134) Head-Internal Relatives (HIRs)

#Kiki-wa [kaeru-ga niwa-de haneteita no]-o tsukamaeta-yo.

K-TOP [frog-NOM garden-in was.hopping NO]-ACC caught-SFP

Int. ‘Kiki caught a frog which was hopping in the garden.’ (not acceptable as an answer to ‘What did Kiki catch’)

(6.135) Change Relatives (CRs)

Kiki-wa [kono otamajakushi-ga kaeru-ni natta no]-o tsukamaeta-yo.

K-TOP [this tadpole-NOM frog-COP became NO]-ACC caught-SFP

‘Kiki caught a frog which is the result of changing from this tadpole.’

They are also distinguished in terms of clefting: FRs and CRs, but not HIRs, may be at a focus position (i.e., a type-e position) in clefts. Consider the FR (6.136) (originally, (6.31)), the HIR (6.137) (originally, (6.32)), and the CR (6.138).

(6.136) Free Relatives (FRs) in clefts

[Kiki-ga todoketa no]-wa [Osono-ga yaita no] da.

[K-NOM delivered NO]-TOP [Osono-NOM baked NO] COP

‘It is Osono’s-baked things that Kiki delivered.’

(6.137) Head-Internal Relatives (HIRs) in clefts

*[Kiki-ga todaketa no]-wa [Osono-ga pan-o yaita no] da.

[K-NOM delivered NO]-TOP [Osono-NOM bread-ACC baked NO] COP

Int. ‘It is Osono’s-baked bread that Kiki delivered.’

(6.138) Change Relatives (CRs) in clefts

[Niwa-o haneteita no]-wa [otamajakushi-ga kaeru-ni natta
 [garden-ACC was.hopping NO]-TOP [tadpole-NOM frog-COP became
 no] da.
 NO] COP

‘It is [a frog which is the result of changing from a tadpole] that was hopping
 in the garden.’

Recall that HIRs present some **seemingly nominal** characteristics. As we see
 below, where HIRs exhibit such properties, **all** of FRs, HIRs, and CRs behave the same.
 First, the *no*-part may be marked by a wide range of particles such as the directive-case
 particle *e* (= ‘to’). The HIR (6.140) is a re-citation of (6.36).

(6.139) Free Relatives (FRs)

Ojisan-wa [hashitte-kita no]-e dekkiburashi-o tewatashita.
 old.man-TOP [run-came NO]-to deck.brush-ACC handed
 ‘Someone came running and an old man handed a deck brush to her.’

(6.140) Head-Internal Relatives (HIRs)

Ojisan-wa [Kiki-ga hashitte-kita no]-e dekkiburashi-o tewatashita.
 old.man-TOP [K-NOM run-came NO]-to deck.brush-ACC handed
 ‘Kiki came running and an old man handed a deck brush to her.’

(6.141) Change Relatives (CRs)

Kiki-wa [otamajakushi-ga kaeru-ni natta no]-e esa-o ageta.
 K-TOP [tadpole-NOM frog-COP became NO]-to food-ACC gave
 ‘Kiki gave food to a frog which is the result of changing from a tadpole.’

Moreover, all of the three constructions license numeral quantifiers, as illustrated in
 (6.142)-(6.144). The HIR (6.143) was originally cited as (6.42).

(6.142) Free Relatives (FRs) with a numeral quantifier

Kiki-wa [teiburu-ni oiteatta no]-o mittsu tabeta.

K-TOP [table-on place.existed NO]-ACC 3 ate

‘Kiki ate 3 of the things on a table.’

(6.143) Head-Internal Relatives (HIRs) with a numeral quantifier

Kiki-wa [pan-ga teiburu-ni oiteatta no]-o mittsu tabeta.

K-TOP [bread-NOM table-on place.existed NO]-ACC 3 ate

‘Some bread was on a table and Kiki ate 3 slices.’

(6.144) Change Relatives (CRs) with a numeral quantifier

Kiki-wa [otamajakushi-ga kaeru-ni natta no]-o san-biki tsukamaeta.

K-TOP [tadpole-NOM frog-COP became NO]-ACC 3-CL caught

‘Kiki caught 3 of the frogs which are the result of changing from tadpoles.’

Finally, all of the three constructions licence secondary predicates, as demonstrated in (6.145)-(6.147). The HIR (6.146) originally appeared as (6.45).

(6.145) Secondary predicate in FRs (Free Relatives)

Jiji-wa [teiburu-ni oite-atta no]-o nama-de tabeta.

J-TOP [table-on place-existed NO]-ACC raw ate

‘Jiji ate the thing on a table raw.’

(6.146) Secondary predicate in HIRs (Head-Internal Relatives)

Jiji-wa [sakana-ga teiburu-ni oite-atta no]-o nama-de tabeta.

J-TOP [fish-NOM table-on place-existed NO]-ACC raw ate

‘Fish was placed on a table, and Jiji ate it raw.’

(6.147) Secondary predicate in CRs (Change Relatives)

Jiji-wa [ikura-ga sake-ni natta no]-o nama-de tabeta.

J-TOP [salmon.roe-NOM salmon-COP became NO]-ACC raw ate

‘Kiki ate [the salmon which is the result of changing from a salmon roe] raw.’

We have discovered the ‘nominality’ correlations between CRs and FRs, excluding HIRs except where HIRs themselves exhibit seemingly nominal features (i.e., case-marking, quantifier float, secondary predicates). Therefore, in spite of the surface similarities, CRs radically differ from HIRs; rather, CRs behave more like FRs. It is indicated that the appropriate encoding for CRs is not the entry of ‘*no* + case particle’ (as used for HIRs) but that of the nominaliser *no* (as used for FRs).¹⁵

5.3 Inertness of the Relevancy Condition in CRs

It has been claimed that CRs/FRs differ from HIRs in that the former presents nominal characteristics alone. I shall raise another point of contrast between CRs/FRs and HIRs: the (in)activeness of the Relevancy Condition.

The relation between the situations described by the HIR and the embedding clauses is subject to an interpretive constraint, the Relevancy Condition (Sect. 2.2). One construal of ‘Relevancy’ is temporal intersection. Thus, it is not possible to assign (6.98), repeated here as (6.148), the temporally-disjoint reading: ‘Kiki has called Jiji, who is now running, to stop in the past.’ This is so even if the HER counterpart (6.99), repeated here as (6.149), may have this interpretation.

¹⁵ In Chap. 5, we analysed clefts but we did not examine whether clefts have nominal status or not. Our analysis of clefts predicts that they have non-nominal properties. This is because *no-wa* maps a propositional tree onto another **propositional** tree. But it is hard to test this prediction because a cleft string has a free relative reading (see Seraku 2011). For example, the string (i) below has the ‘specificational’ reading (ia), the ‘equative’ reading (ib), and the ‘predicational’ reading (ic) in the sense of Nishiyama (2003); see also Declerck (1988). In (ia), the string is a case of clefts, and in (ib)-(ic), it is a case of free relatives. Given that free relatives have nominal nature, it is difficult to elicit the non-nominal nature of clefts.

- (i) [Jitensha-ni notteiru no]-wa Tombo da.
 [bicycle-in is.riding NO]-TOP T COP
 a. ‘It is Tombo that is riding in a bicycle.’
 b. ‘That person riding in a bicycle is the same person as the one called Tombo.’
 c. ‘That person riding in a bicycle has the name *Tombo*.’

(6.148) Head-Internal Relatives (HIRs)

Kiki-wa [Jiji-ga hashi-tteiru no]-o yobitomete.

K-TOP [J-NOM running-PROG NO]-ACC called.to.stop

‘Kiki called Jiji to stop **while** he was running.’

(6.149) Head-External Relatives (HERs)

Kiki-wa [(ima) hashitteiru Jiji]-o (mukashi) yobitomete.

K-TOP [(now) be.running J]-ACC (in.the.past) called.to.stop

a. ‘Kiki called Jiji, who was running, to stop.’

b. ‘Kiki has called Jiji, who is now running, to stop in the past.’

In Sect. 2.5.3, it was pointed out that the Relevancy Condition was inert in FRs. Given the conclusion reached in the last sub-section that CRs behave more like FRs than HIRs, one may anticipate that CRs are free from the Relevancy Condition. This is ascertained by (6.150) (originally, (6.141)), which may be interpreted as: ‘A tadpole became a frog 1 year ago and Kiki fed it today.’

(6.150) Change Relatives (CRs)

Kiki-wa [otamajakushi-ga kaeru-ni natta no]-e esa-o ageta.

K-TOP [tadpole-NOM frog-COP became NO]-to food-ACC gave

‘Kiki gave food to a frog which is the result of changing from a tadpole.’

One might object that (i) the Relevancy Condition is also operative for CRs but that (ii) it is **trivially** met by the resulting state denoted by CRs, which usually coincides with the event described by the embedding clause. For instance, one could argue that the resulting state in (6.150) (i.e., the state of being a frog) temporally intersects with the event described by the matrix clause (i.e., the event where Kiki fed the frog). In order to show the inactiveness of the Relevancy Condition in CRs more convincingly, consider the example (6.151).

(6.151) Change Relatives (CRs)

Kiki-wa [kono-otamajakushi-ga kaeru-ni natta no]-o
 K-TOP [this-tadpole-NOM frog-COP became NO]-ACC
 fukasuru-maeni kinjo-no-kawa-de mitsuketeita.
 hatch-before neighbourhood-GEN-river-at has.found

‘Kiki has found [the frog which is the result of changing from this tadpole] in the neighbouring river when it was still an egg.’

Due to the clausal adverb *fukasuru-maeni* (= ‘before it hatches from an egg’), the resulting state described by the CR clause (i.e., the state of being a frog) does not temporally intersect with the situation described by the matrix clause (i.e., the event where Kiki saw the egg of the frog in the river). That is, the CR (6.151) may have the interpretation: ‘A tadpole became a frog today, and Kiki had found it at a river before it hatched.’ The availability of this interpretation shows that the Relevancy Condition is inactive in CRs, like FRs but unlike HIRs.

This ‘Relevancy Condition’ parallelism between CRs and FRs (but excluding HIRs) bolsters our claim that CRs are to be modelled by the nominaliser *no* (as used for FRs), but not the unit ‘*no* + case particle’ (as used for HIRs).

5.4 The Analysis

It has been revealed that CRs are to be treated on the same grounds as FRs, but not as HIRs. In this sub-section, I shall first analyse CRs by manipulating the entry of the nominaliser *no*. This account has theoretical implications for the classification of *no*-nominalising constructions.

The tree update for CRs proceeds in essentially the same manner as for FRs. Let us go through a tree transition for the CR (6.152). The exposition is cursory; for details, see Chap. 4 (Sect. 3).

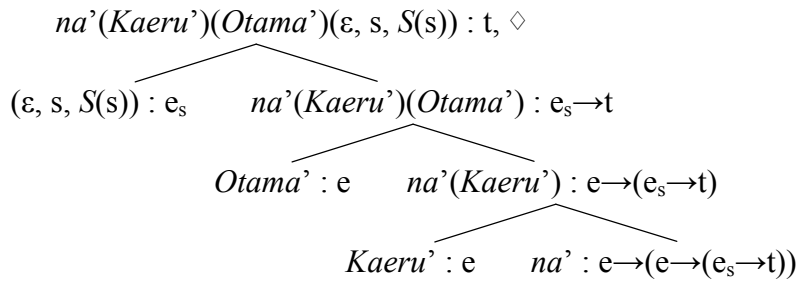
(6.152) Change Relatives (CRs)

[Otamajakushi-ga kaeru-ni na-tta no]-ga niwa-o hane-teiru.

[tadpole-NOM frog-COP become-PAST NO]-NOM garden-in hop-PROG

‘A frog which is the result of changing from a tadpole is hopping in the garden.’ (Tonosaki 1998: 144)

As always, the AXIOM sets out an initial type-*t*-requiring node. The parse of the pre-*no* clause brings forth the tree (6.153) (ignoring tense). The propositional content at the current top node is then Q-EVALUATED as in (6.154). For the sake of simplicity, both *otamajakushi* (= ‘a tadpole’) and *kaeru* (= ‘a frog’) are treated as if they were proper names, as in *Otama’* and *Kaeru’*.

(6.153) Parsing *Otamajakushi-ga kaeru-ni na-tta*

(6.154) Evaluating the proposition in (6.153)

$S(a) \& na'(Kaeru')(Otama')(a)$

where $a = (\epsilon, s, S(s) \& na'(Kaeru')(Otama')(s))$

The next item is the nominaliser *no*, whose entry is reproduced as (6.155). That is, the parser (i) inversely LINKs the type-*t* node to a fresh node and (ii) annotates it with a term in the evaluated proposition (in the present case, *Kaeru'*)¹⁶ and the logical type *e*.

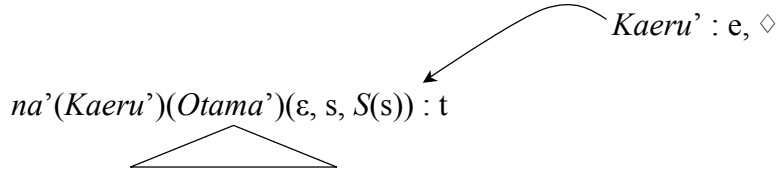
¹⁶ Technically, the other individual term *Otama'* may be copied. The resulting interpretation is something like ‘A tadpole which has become a frog is hopping in the garden.’ But this reading is not possible to draw in the CR (6.152), though it is possible in the HER counterpart. This may be because in CRs, focus is put on an individual that has undergone a change of state.

(6.155) Entry of the nominaliser *no*

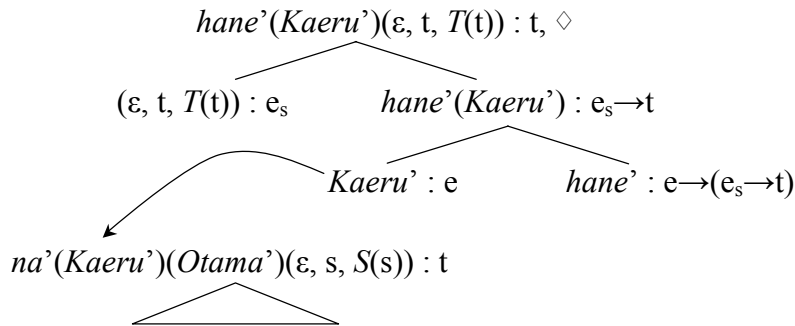
```

IF      t
THEN    IF       $\varphi_E[\alpha]$ 
        THEN    make/go( $\langle L^{-1} \rangle$ ); put( $\alpha : e$ )
        ELSE    ABORT
ELSE    ABORT

```

(6.156) Parsing *Otamajakushi-ga kaeru-ni na-tta no*

The current node is identified as a subject node within a new tree by the nominative-case particle *ga*, and the tree is fleshed out by the matrix verb *hane-* (= ‘hop’). After ELIMINATION is run, the final output (6.157) emerges (ignoring the adjunct *niwa-o* (= ‘in the garden’) and the tense/aspect).

(6.157) Parsing [*Otamajakushi-ga kaeru-ni na-tta no*]-*ga niwa-o hane-teiru*

The analysis of CRs in terms of the nominaliser *no* has implications for the classification of *no*-nominalising constructions. In Chap. 4 (Sect. 3), it was argued that the two types of *no*-nominalisation are reducible to a parser’s choice of what term it copies in processing *no*. As summarised in Table 6.1, if an individual term is selected, it leads to participant nominalisation, whereas if a situation term is selected, it leads to situation nominalisation.

(6.158) Free Relatives (FRs) (= participant nominalisation)

Mei-wa [Chibi-Totoro-ga otoshi-ta no]-o hiro-tta.

M-TOP [CT-NOM drop-PAST NO]-ACC pick.up-PAST

‘Mei picked up a thing which Chibi-Totoro dropped.’

(6.159) Situation nominalisation

Mei-wa [Nekobasu-ga hashiru no]-o mi-ta.

M-TOP [N-NOM run NO]-ACC see-PAST

‘Mei saw an event where Nekobasu was running.’

type of a term to be copied	construction
individual term	Free Relatives (FRs)
situation term	situation nominalisation

Table 6.1. Classification of *no*-nominalising constructions (preliminary)

Note that an inherent property of FRs is the presence of a gap in the relative clause. But the nominaliser *no* (6.155) does **not** impose any restrictions on the presence of a gap. So, the parser may copy an individual term even when a gap is absent. In fact, copying of an individual term **without** a gap characterises CRs, as demonstrated above. Thus, the classification is delineated as in Table 6.2.

type of a term to be copied	with a gap	without a gap
individual term	Free Relatives (FRs)	Change Relatives (CRs)
situation term	situation nominalisation	

Table 6.2. Classification of *no*-nominalising constructions (final version)

Is the presence/absence of a gap relevant to situation nominalisation? In our analysis, a situation term represents tense and aspect, among other things (see Sect. 4.4, Chap. 4). For example, in (6.159), the past tense marker *ta* encodes the instruction to construct a situation term which denotes a past event (see the appendix for the lexical entry of *ta*). I shall assume that in situation nominalisation, a situation node is always decorated with a situation term modelling tense or aspect (rather than a situation term modelling a gap).

Put differently, a string manifests situation nominalisation whenever the parser copies a situation term at the time of processing *no*.

One nontrivial task remains outstanding. I have argued that the nominaliser *no* was not suited for HIRs and that the unit ‘*no* + case particle’ was not suited for CRs. Though the critical differences between CRs and HIRs have been catalogued, they are quite similar at a surface level. Thus, the question remains of how the parser recognises the nominaliser *no* in the CR string (but not in the HIR string) and the unit ‘*no* + case particle’ in the HIR string (but not in the CR string). A possible clue is that the ‘change of state’ is involved only in CRs, but this is a residual issue for future research.

5.5 Summary

FRs and CRs (but not HIRs) share the nominality and the inertness of the Relevancy Condition. These resonances justify the analysis of CRs in terms of the nominaliser *no* (rather than the unit ‘*no* + case particle’). As a bonus, this analysis refines the typology of *no*-involving constructions. FRs, CRs, and situation nominalisation are distinguished in terms of (i) whether the string contains a gap or not and (ii) whether the parser copies an individual term or a situation term at the time of processing *no*.

6 Conclusion

Continuing the analyses of *no*-nominalisation (Chap. 4) and clefts (Chap. 5), the present chapter has inquired into the *no*-involving relative clause constructions: Free Relatives (FRs), Head-Internal Relatives (HIRs), and Change Relatives (CRs).

A substantial part of this chapter is devoted to the analysis of HIRs. We have regarded HIRs as instantiating the dynamic flow of language understanding: the HIR clause provides a context against which the immediately embedding clause is processed within a partially-articulated tree. To this end, the unit ‘*no* + case particle’ is assigned

the procedures to inversely LINK the tree of the HIR clause to a partial structure with two type-e nodes: (i) the node for a situation term with the ‘Relevancy’ restriction and (ii) the node for an internal head. This action package is defined based on the entries of the nominaliser *no* and the cleft marker *no-wa*. This dynamic account predicts a variety of data (e.g., indeterminacies of the head, quantification, island-insensitivity), including the data problematic for recent accounts of HIRs (e.g., optionality of maximality effects, negation, locality in the Relevancy Condition).

This account, however, is not applicable to FRs and CRs because they are contrasted with HIRs in terms of the nominality and the Relevancy Condition. Firstly, FRs and CRs exhibit nominal status, whereas HIRs exhibit non-nominal status (as well as seemingly nominal properties). Secondly, FRs and CRs are free from the Relevancy Condition, while HIRs are not. It is argued that FRs and CRs are analysed by the nominaliser *no* (rather than the unit ‘*no* + case particle’). (In fact, FRs are equated with ‘participant nominalisation’). Our dynamic analysis clarifies the classification of *no*-nominalising phenomena: FRs, CRs, and situation nominalisation are reduced to the combinations of two factors: (i) whether the parser copies an individual term or a situation term and (ii) whether the string involves a gap or not.

In conclusion, this thesis has investigated the various kinds of the *no*-involving phenomena. Key items in these constructions are depicted in the following figure. In the next, closing chapter, the dynamic conception of ‘construction’ is suggested.

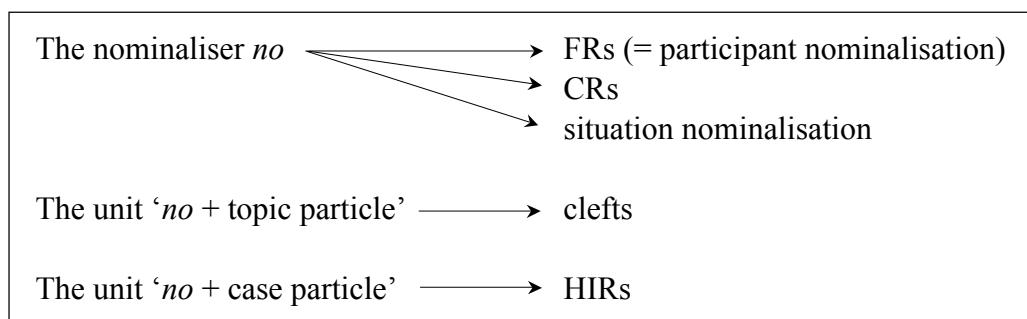


Figure 6.2. Form-function correspondences

Chapter 7

Synopsis and Beyond

1 Overall Summaries

Our naïve conception of language is that language is a tool for communication. It is something which enables us to talk and listen in a specific contextual setting. We have focussed on the comprehension side, especially the dynamic process in which we take an incoming string word-by-word and accumulate a context-dependent interpretation in a piecemeal fashion. Our central theme is that this naïve view of language, if formalised as a grammatical model, provides a promising line of analysis of various issues in the syntax and semantics of natural languages. In this thesis, we have defended the theme by analysing several *no*-involving constructions in Japanese.

We have embarked upon our inquiry by setting out empirical and theoretical scaffoldings. Chap. 2 presents spontaneous data of the *no*-involving constructions and draws some generalisations. They reveal various properties of *no*-nominalisation and clefts, laying a milestone in developing our account of these constructions in subsequent chapters. Chap. 3 establishes our theoretical underpinnings, the framework of Dynamic Syntax (DS). We largely follow the orthodox literature (Cann et al. 2005, Kempson et al. 2001, Kempson et al. 2011) but also advance some formal aspects of the framework (Seraku 2013b, in press a). Most importantly, it is assumed that the node set out by the AXIOM is underspecified for a node position and that the initial structure may turn out to

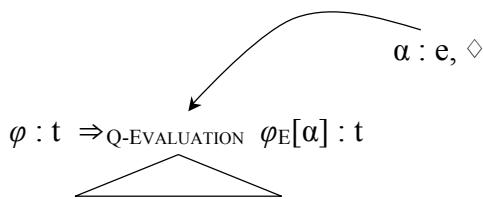
be a subordinate structure. Further, the node-identification process clarifies how the globally unfixed relation (as induced by GENERALISED ADJUNCTION) is fixed during the course of incremental parsing. Some lexical actions are also modified. For instance, the lexical entry of case particles is enriched with a bottom-up tree building process. These amendments lead to a more adequate account of head-external relatives in Japanese.

Based on these empirical and theoretical backgrounds, several *no*-involving constructions are addressed. First of all, Chap. 4 develops an account of the particle *no* by analysing participant and situation nominalisation. We follow the entry of *no* as a nominaliser (Cann et al. 2005: 285), which maps a proposition of the pre-*no* clause onto a type-e term that denotes an entity reflecting the proposition. In the tree display (7.2), α is a type-e term that reflects the evaluated proposition of the pre-*no* clause φ_E .

(7.1) Entry of the nominaliser *no*

IF	t	
THEN	IF	$\varphi_E[\alpha]$
	THEN	make/go(<L ⁻¹ >); put($\alpha : e$)
	ELSE	ABORT
ELSE	ABORT	

(7.2) Output structure of parsing *no*



The entry of *no* (7.1) integrates participant and situation nominalisation. The distinction boils down to a parser's choice of what term (i.e., an individual or a situation term) it copies in processing *no*. This unitary treatment is motivated by functional, diachronic, cross-linguistic, and cross-dialectal considerations. It has also been noted that the two types of *no*-nominalisation exhibit a contrast when coordinate clauses are nominalised.

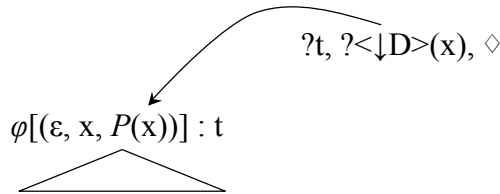
This asymmetry follows from our analysis without losing uniformity if the issues of coordination and tense are addressed in DS terms.

In Chap. 5, we turn to clefts. In the literature (Hoji 1990), clefts are divided into two types depending on whether the focus has a case particle: ‘clefts_{+C},’ where the focus is case-marked and ‘clefts_{-C},’ where the focus is case-less. Language dynamics enables a unified analysis of these two kinds of clefts. A key sequence is the succession of the nominaliser *no* and the topic particle *wa*. The entries of these items are combined, as in (7.3). ϕ is a proposition of the pre-*no-wa* clause, and $(\varepsilon, x, P(x))$ is a content of a gap in ϕ . The unit *no-wa* maps a propositional tree of the pre-*no-wa* clause onto another propositional tree. As shown in (7.4), the focus will be parsed in the emergent tree, with the presupposition tree as context. In particular, $?<\downarrow D>(x)$ requires that the content of the focus will share the variable x as the content of the gap in the presupposition tree.

(7.3) Entry of the cleft marker *no-wa*

IF $\phi[(\varepsilon, x, P(x))] : t$
 THEN $\text{make/go}(<L^{-1}>); \text{put}(?t, ?<\downarrow D>(x))$
 ELSE ABORT

(7.4) Output structure of parsing *no-wa*



Our dynamic account integrates clefts_{+C} and clefts_{-C}. The two kinds of clefts differ solely in the mode of resolving an unfixed node for the focus: the lexical action of the case particle (clefts_{+C}) vs. the general action UNIFICATION (clefts_{-C}). The tree updates finally converge, and the truth-conditional equivalence holds between (a minimal pair of) clefts_{+C} and clefts_{-C}. This uniform treatment meshes well with the corpus findings (e.g., the indication that the two types of clefts should not be distinguished structurally)

and with the experimental evidence for incrementality in cleft processing (Kahraman et al. 2011a).

Our unitary analysis also succeeds in explaining distinct behaviours between the two classes of clefts. First, it has been held that multiple foci are possible only in clefts_{+C} (Koizumi 1995). But the situation is not this simple; we observe that a case particle may be dropped off the final focus (and only the final focus). This mixed case is problematic for previous works which assign a completely distinct structure to a cleft string depending on whether the focus is case-marked (Hiraiwa & Ishihara 2012, Kizu 2005). In our analysis, the unfixed node of each focus is immediately resolved before the next focus is scanned. This incremental resolution process of an unfixed node explains the multiple foci asymmetry, the mixed case, and further sets of data.

The second set of contrast concerns islands (Hoji 1990). We propose that the optionality of general actions allows us to process the focus by invoking LINK relations (rather than by inducing an unfixed relation). This LINK-based processing of the focus is equally available for the two types of clefts, but a well-formed final state (i.e., a tree without outstanding requirements) obtains only when the focus lacks a case particle. This predicts that clefts_{+C} are sensitive to islands while clefts_{-C} are not.

Our account of clefts is extendable to stripping and sluicing. The overarching picture is that clefts, stripping, and sluicing are uniformly characterised regardless of whether the focus item is case-marked. This is in sharp contrast with previous studies of stripping and sluicing, where completely distinct structures are hypothesised depending on the case-marking of a focus item (Fukaya 2007; see also Takahashi 1994). Our analysis predicts that clefts, stripping, and sluicing all share (i) the same case-marking pattern (including partial case-marking, problematic data for previous works) and (ii) the same island-sensitivity pattern (Fukaya 2007).

Finally, Chap. 6 examines the three *no*-involving relative constructions: FRs (Free Relatives), Change Relatives (CRs), and Head-Internal Relatives (HIRs). ‘Free

relatives' is another term for 'participant nominalisation.' So, FRs are dealt with by the nominaliser *no* (7.1). It is shown that CRs behave the same as FRs in that (i) they have nominal status and that (ii) the Relevancy Condition (Kuroda 1992) is inactive. So, CRs are modelled by the nominaliser *no* (7.1) on a par with FRs (cf., Tonosaki 1998). This analysis, however, is inapplicable to HIRs: (i) HIRs display non-nominal properties (as well as seemingly nominal features); (ii) HIRs are subject to the Relevancy Condition. We propose to coalesce the analysis of *no*-nominalisation (i.e., the type-e node building) and that of clefts (i.e., the pairing of propositional trees). In (7.5), φ_E is an evaluated proposition of the HIR clause, α is a situation term in φ_E , and β is an individual term in φ_E . The unit '*no* + case particle' maps the evaluated proposition φ_E onto an emergent propositional tree. This partial tree contains a situation node with the 'Relevancy' requirement (modelled by the predicate $\star$) and a node for an internal head. The position of the latter node is guided by the case particle. In *no-ga*, the nominative-case particle *ga* selects $\langle \downarrow_1 \downarrow_0 \rangle$, indicating that the node is in a subject position.

(7.5) Entry of the HIR marker '*no* + case particle'

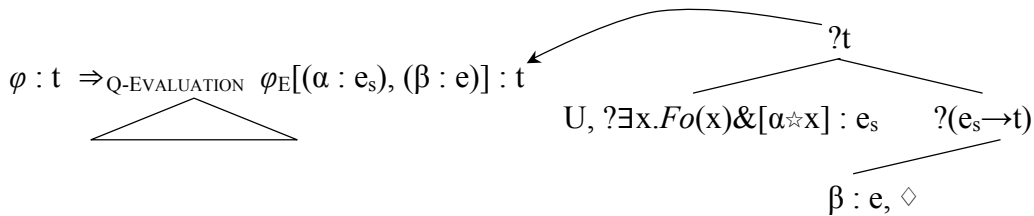
```

IF      t
THEN    IF       $\varphi_E[(\alpha : e_s), (\beta : e)]$ 
        THEN    make/go( $\langle L^{-1} \rangle$ ); put( $?t$ );
                make/go( $\langle \downarrow_0 \rangle$ ); put( $U, ?\exists x.Fo(x) \& [\alpha \star x] : e_s$ ); go( $\langle \uparrow_0 \rangle$ );
                make/go( $\langle \mu \rangle$ ); put( $\beta : e$ )
        ELSE    ABORT
ELSE    ABORT

```

where $\mu \in \{ \downarrow_1 \downarrow_0, \downarrow_1 \downarrow_1 \downarrow_0, \downarrow_1 \downarrow_1 \downarrow_1 \downarrow_0, \dots \}$

(7.6) Output structure of parsing '*no* + *ga*'



This dynamic account predicts a wide array of data on HIRs, some of which falsify recent accounts (e.g., Grosu 2010, Grosu & Landman 2012, Kim 2007).

The form-function pairs of the central expressions are charted in Figure 7.1. As stressed in Chaps. 5-6, the entries of ‘*no* + topic particle’ and ‘*no* + case particle’ are not mere stipulations. First, ‘*no* + topic particle’ is formed through routinisation to save the cost of multiplying LINK relations separately induced by *no* and the topic particle *wa*. Second, the fusion of *no* and a case particle is diachronically plausible (Kuroda 2005) and also motivated by the analysis of *no*-nominalisation (i.e., the type-e node building) and clefts (i.e., the pairing of propositional trees); further, it has the residue of the entry of case particles (i.e., the action to mark a node position within a local tree).

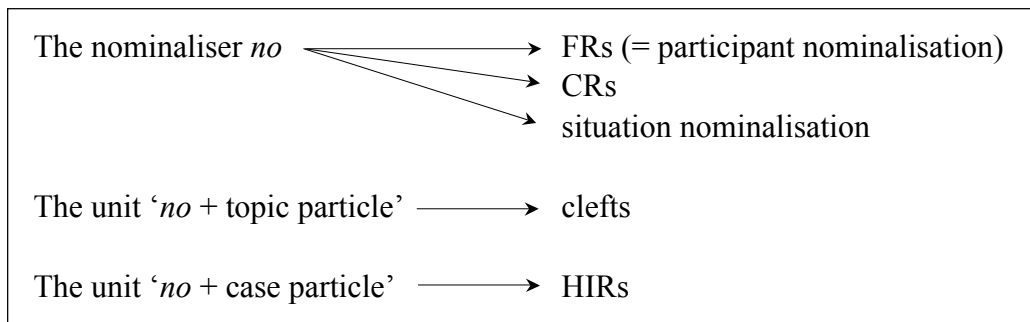


Figure 7.1. Form-function correspondences

As shown in Figure 7.1, the nominaliser *no* is a basis for modelling the three distinct kinds of constructions: FRs, CRs, and situation nominalisations. These constructions are classified in terms of (i) a parser’s decision online (i.e., whether the parser copies an individual term or a situation term) and (ii) the nature of an incoming string of words (i.e., whether it contains a gap or not).

type of a term to be copied	with a gap	without a gap
individual term	Free Relatives (FRs)	Change Relatives (CRs)
situation term	situation nominalisation	

Table 7.1. Classification of *no*-nominalising constructions

In conclusion, language dynamics, as formalised in Dynamic Syntax, is vital for articulating an explanatory, realistic, integrated, and empirically favourable account of the various *no*-involving constructions in Japanese. That is, the account (i) unifies a range of seemingly distinct sets of data (e.g., two types of *no*-nominalisation, two types of clefts) and (ii) solves a number of formidable puzzles (e.g., partial case-marking of clefts, the nominal/non-nominal ambivalence of HIRs) as an outcome of progressive growth of semantic representation driven by incremental processing online. This result inexorably constitutes a strong set of arguments for our theme: language dynamics is a significant concept in linguistic theorising.

2 Future Prospects

We have accomplished internally-consistent, satisfactory grammatical descriptions (and, where possible, explanations) of our fragments of the Japanese language, the *no*-involving constructions: *no*-nominalisation, clefts, and certain kinds of relative clauses. The term ‘construction’ has been used pre-theoretically, but what does it exactly mean? I shall close this thesis by speculating on a dynamic conception of construction.

2.1 Construction in Linguistic Theories

The concept ‘construction’ has a long tradition in the linguistics literature. It has already been discussed in the 12th century by the Medieval Linguists, the Modistae (Goldberg & Casenhiser 2007). Still, in the present day, there are many conflicting views on how ‘construction’ should be defined and what counts as a construction. In this sub-section, we shall take a brief look at how constructions have been treated in some representative linguistic frameworks.

Within MGG (Mainstream Generative Grammar), at least in the Principles-&-Parameters setting (including Minimalist Program), ‘construction’ is a descriptive term

and there is **no** set of principles or rules that are tailored for a particular form-meaning pair. Syntactic properties of such pairs are ideally explained through the interaction of fundamental principles (set out by Universal Grammar, a theory of the initial state of language faculty) and rules (specified by an individual grammar, a theory of a final state of language faculty) (Chomsky 1981, 1986, 1995). This is in fact a leading hypothesis in previous MGG studies on Japanese clefts (Kizu 2005: 1) and Japanese head-internal relatives (Hoshi 1995: 2).

Unlike MGG, construction is a **central** unit in a family of CxG (Construction Grammar) theories (Boas & Sag 2013, Croft 2001, Fillmore 1985, Fillmore et al. 1988, Goldberg 1995, Lakoff 1987, Langacker 2003; see Hoffmann & Trousdale 2013 for overviews). In these theoretical strands, construction is an elaborated version of ‘sign’ (Saussure 1916). First, while Saussurean signs are a form-meaning pair, constructions encode not only form-meaning information but also other sorts of information (e.g., syntactic). Second, while Saussurean signs are restricted to lexical items, constructions are defined at both lexical and phrasal levels (Sag et al. 2003: 474-5). In SBCG (Sign-Based Construction Grammar), signs and constructions are clearly distinguished: signs are linguistic objects, represented as feature structures; constructions are not linguistic objects but descriptions of certain linguistic objects called ‘constructs’ (see (7.8) below) (Sag 2013). Construction in CxG generally includes (i) prototypical lexical patterns (e.g., lexical expressions such as *plane* and *fly*), (ii) less prototypical lexical patterns (e.g., idiomatic expressions such as *keep tabs on*), (iii) prototypical phrasal patterns (e.g., *wh*-questions), and (iv) less prototypical phrasal patterns (e.g., *way*-constructions as in *Frank dug his way out of the prison* (Goldberg 1995: 199)). But this classification is just a matter of descriptions; CxG assumes that the lexicon-grammar distinction is not clear-cut but forms a **continuum**.

In CxG theories, constructions are **classified** by partially linked taxonomies, which express commonalities and differences among constructions. They also serve as a

platform where new constructions emerge, such as the denominal use of *sister* in *We sistered the joints* (Michaelis 2009: 155). There are various network models; to mention a few, constructions are linked by ‘family resemblance’ (Lakoff 1987), by ‘inheritance links’ (Goldberg 1995), and by ‘elaboration’ and ‘extension’ (Langacker 2003). Of note are ‘multiple inheritance hierarchies’ of types in SBCG (Boas & Sag 2013). Consider Figure 7.2 (adapted from Sag et al. 2003: 473). This taxonomy organises types (i.e., classes of linguistic objects modelled as feature structures) in a cross-cutting manner: a feature structure labelled by a type must inherit all constraints imposed on the feature structures labelled by the dominating types. For example, the feature structure labelled by *core-clause* must obey all constraints imposed on the feature structures labelled by *clause*, *phrasal-construct*, and *construct*.

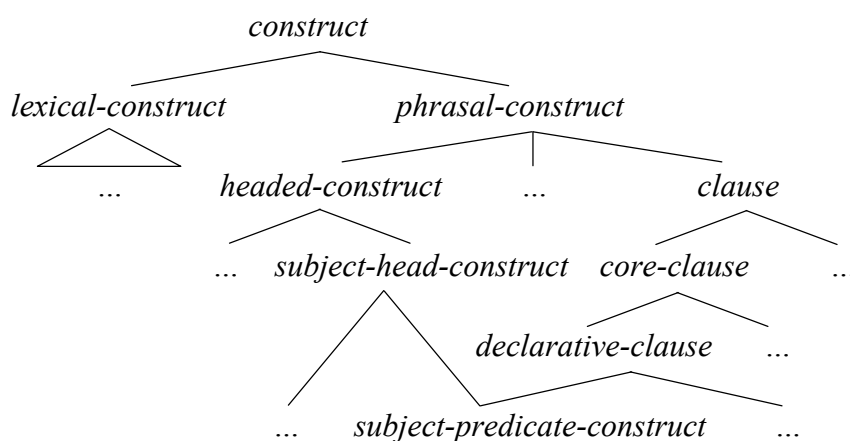


Figure 7.2. Multiple inheritance hierarchy

The type *construct* is divided into *lexical-construct* and *phrasal-construct*. The former labels a feature structure which corresponds to a word or a lexem. The latter labels a feature structure that exhibits combinatorial properties, as licensed by a construction. A case in point is the construct labelled by the type *subject-predicate-construct* (7.7) (adapted from Sag et al. 2013: 17).

(7.7) Construct labeled by *subject-predicate-construct*

<i>subject-predicate-construct</i>		
MOTHER	$\begin{bmatrix} \text{PHON} & \langle \text{Horikoshi, runs} \rangle \\ \text{SYN} & \text{S}[\textit{finite}] \\ \text{SEM} & \dots \end{bmatrix}$	
DAUGHTERS	$\left\langle \begin{bmatrix} \text{PHON} & \langle \text{Horikoshi} \rangle \\ \text{SYN} & \text{NP} \\ \text{SEM} & \dots \end{bmatrix}, \begin{bmatrix} \text{PHON} & \langle \text{runs} \rangle \\ \text{SYN} & \text{V}[\textit{finite}] \\ \text{SEM} & \dots \end{bmatrix} \right\rangle$	

Constructs are licensed by constructions, or type constraints stated in conditional format as shown in (7.8) (adapted from Sag et al. 2013: 16). The construct (7.7) must satisfy (i) whatever the dots ... in the construction (7.8) state and (ii) the constraints imposed on the feature structures labelled by any super-types of *subject-predicate-construct*.

(7.8) Construction that licenses the construct (7.7)

subject-predicate-construct $\Rightarrow$...

An alternative is offered in LFG (Lexical Functional Grammar). Asudeh et al. (2013) claim that ‘templates’ capture the concept ‘construction’ **without** removing the lexicon-syntax and the word-phrase distinctions. LFG postulates two levels of syntactic structure: c(onstituent)-structure and f(unctional)-structure. C-structure trees express syntactic category of words, linear order, and phrasal dominance/grouping. This level of structure is governed by Lexical Integrity (Bresnan 2001: 92), the underlying idea being that syntactic operations are blind to word formation. Thus, the lexicon-syntax and the word-phrase divides are maintained in c-structure. By contrast, f-structure represents more abstract syntactic information such as grammatical functions (e.g., subject, object) and syntactic features (e.g., person, case). Asudeh et al. (2013) regard a template as an abbreviation of f(unctional)-descriptions as in (7.9). With this template, the entries of *laughs* and *laugh* are expressed as (7.10) and (7.11); @ is a substitution marker.

(7.9) Template: 3SG

(↑ SUBJ PERS) = 3

(↑ SUBJ NUM) = SG

(7.10) Entry of *laughs*

(↑ PRED) = 'laugh<SUBJ>'

@3SG

(7.11) Entry of *laugh*

(↑ PRED) = 'laugh<SUBJ>'

¬@3SG

(adapted from Asudeh et al. 2013: 18)

Templates are organised as in Figure 7.3 (Asudeh et al. 2013: 19). This taxonomy classifies f-descriptions (rather than feature structures as in Figure 7.2). A relation between nodes is that of 'inclusion.' That is, if a template A dominates a template B, the description labelled by A is included in the description labelled by B. In the present case, the description of 3SG (= (7.9)) is included in the entries of *laughs* (= (7.10)) and *laugh* (= (7.11)).¹

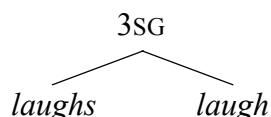


Figure 7.3. Template hierarchy

Templates may be schematised as in (7.12), where P is to be saturated as the content of an intransitive verb. This template models the intransitive verb construction.

(7.12) Template: INTRANSITIVE(P)

(↑ PRED) = 'P<SUBJ>'

¹ Note that the template 3SG is included in *laugh* even if it is negated. This is not possible in the inheritance relation as in Figure 7.2 (Asudeh et al. 2013: 19).

This analysis posits **no** primitive notion ‘construction’ because templates are merely abbreviations of f-descriptions. Asudeh et al. (2013) present a case study of the *way*-construction in English and the Dutch/Swedish analogues, arguing that parallelisms and differences among these constructions are captured by a template hierarchy.

To sum up, while MGG does not put ‘construction’ on a theoretical footing, a family of CxG theories is centred upon variously construed notions of construction, with SBCG being one of the most explicitly articulated versions of CxG. The central ideas shared by the CxG practitioners are that (i) constructions (i.e., form-meaning units, possibly with other linguistic information) are not restricted in size, that (ii) they form a network or a hierarchy, and that (iii) they reflect the lexicon-grammar cline. In LFG, the lexicon and a grammar are kept apart by Lexical Integrity, and constructions are treated as epiphenomenal with the use of templates.

2.2 Towards Dynamic Construction Grammar

The concept of construction has not been lucidly characterised in the DS literature, and I shall put forward my conjecture in a preliminary but suggestive way.²

The sense of dynamics within DS is the time-linear processing of a string and attendant growth of an interpretation. This is modelled by successive updating of a semantic tree without a separate level of syntactic representation. What inhabits a DS tree is not the word but a semantic content (as well as labels of the content, such as the logical type). The incremental tree-update is triggered by general, lexical, and pragmatic actions. So, the DS information encoding is inherently **procedural**, though declarative constraints are also posited. This is even true of nouns; consider the encoding of the proper name *Horikoshi*.

² I am grateful to Arash Eshghi and Ruth Kempson for their helpful comments on the earlier version of the present sub-section.

(7.13) Entry of *Horikoshi*

```

IF      ?e
THEN    put(Horikoshi' : e)
ELSE    ABORT

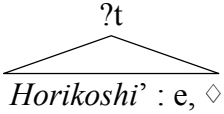
```

This entry declares that if the input condition in the IF-block is met, the action sequence in the THEN-block applies. *Horikoshi*' is a semantic content of *Horikoshi*, but the entry *per se* is not a content but an action to decorate a node (if the condition is fulfilled) and an action to abort a tree update (if the condition is not fulfilled). Therefore, unlike CxG, constructions cannot be simply viewed as a form-meaning pair but a tripartite of 'form,' 'action,' and 'interpretation.' This is a **dynamic tripartite**: 'form' leads the parser to run 'action' to construct 'interpretation' in online processing.

Since my primary aim here is to suggest a possible DS notion of construction, the dynamic tripartite is represented relatively informally as in (7.14). In the FORM part, the underlined item indicates form information (e.g., orthographical, phonological). It is angle-bracketed to indicate linearity. In the ACTION part, tree-update actions are linearly ordered; A_n is the conditional action sequence encoded in *Horikoshi*. In the TREE part, T_0 is an initial state specified by the AXIOM; A_1 maps T_0 onto T_1 , A_2 maps T_1 onto T_2 , and so on up to T_{n-1} . All of these form part of the context for that action which is encoded in *Horikoshi*. T_n is then the tree generated by A_n as a mapping from T_{n-1} onto T_n . T_n is schematically displayed for there is more than one tree sequence that may lead to the truth-conditionally equivalent final states (see the discussion about (7.16)).³

³ The TREE part might look merely a graphical display of the ACTION part. But the TREE part corresponds to the 'meaning' part in the 'form'/'meaning' pair within Construction Grammar. As will be clarified in the text, however, the TREE part does not represent an interpretation but a contribution towards interpretation. This is because the TREE part is not complete in that it has left outstanding requirements.

(7.14) The proper name *Horikoshi* construction

FORM	<..., <u>Horikoshi</u> , ...>
ACTION	for some sequence <A ₁ , A ₂ , ..., A _{n-1} >, A _n , the action encoded in <u>Horikoshi</u> , is: IF ?e THEN put(<i>Horikoshi</i> ' : e) ELSE ABORT
TREE	for some sequence < T ₀ , T ₁ , T ₂ , ..., T _{n-1} >, A _n maps T _{n-1} onto T _n , and T _n is represented as: ?t  <i>Horikoshi</i> ' : e, ◇

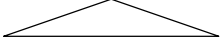
One might object that the ACTION part conflates different types of actions (i.e., general, lexical, pragmatic). This objection is untenable since (i) the DS architecture is **domain-general**, (ii) it employs **the same vocabulary** to express all DS actions (e.g., formula values, LOFT), and (iii) it also exploits **the same format** to define all DS actions (i.e., conditional format consisting of IF, THEN, and ELSE). Thus, the ACTION part is sound even if it comprises all general, lexical, and pragmatic actions. Moreover, since general and lexical actions are treated essentially the same ('optionality' aside), the existence of a lexicon-grammar cline is then **directly** predicted with the DS architectural design.

This dynamic tripartite (7.14) specifically refers to the proper name *Horikoshi*. It is possible to replace notations specific to *Horikoshi* with **rule meta-variables**, or place-holders in formulating DS rules (Eshghi et al. 2011: 20).⁴ For instance, consider the label *Ty*(X), where *Ty* is a predicate for logical-type values (see the appendix) and X is a rule meta-variable. *Ty*(X) is true at a node just in case there is an instantiation for the rule meta-variable X such that *Ty*(X) is present at the node. If a current node is decorated with *Ty*(e), *Ty*(X) holds at this node, where X is instantiated as the type e. With rule meta-variables, (7.14) is schematised as in (7.15). This abstract tripartite is the

⁴ See the appendix for the use of 'rule meta-variables' in defining DS actions.

‘proper name construction.’ This schema is nothing more than a general description of the dynamic tripartite (7.14) and it does not have to be (though it could be) stored in linguistic competence.⁵

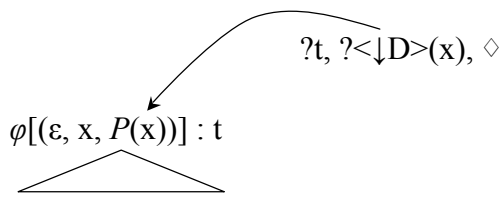
(7.15) The proper name construction

FORM	$\langle \dots, \underline{X}, \dots \rangle$
ACTION	for some sequence $\langle A_1, A_2, \dots, A_{n-1} \rangle$, A_n , the action encoded in $\underline{X}$, is: IF ?e THEN put($X' : e$) ELSE ABORT
TREE	for some sequence $\langle T_0, T_1, T_2, \dots, T_{n-1} \rangle$, A_n maps T_{n-1} onto T_n , and T_n is represented as: $?t$  $X' : e, \diamond$

Turning to a slightly more complex example, the cleft construction in Japanese may be modelled as the dynamic tripartite in (7.16). Note that there is more than one way the tree T_n is subsequently developed; the parser may execute LOCAL *ADJUNCTION or LINK INTRODUCTION to process an upcoming focus item (see Sect. 5, Chap. 5).

⁵ The present thesis does not make any particular claims about the schematisation process. For suggestive proposals, see Langacker’s (1987, 2000) ‘Usage-Based Model.’ See also Tomasello (2003) for evidence from language acquisition.

(7.16) The cleft construction

FORM	$\langle \dots, \text{no-wa}, \dots \rangle$
ACTION	for some sequence $\langle A_1, A_2, \dots, A_{n-1} \rangle$, A_n , the action encoded in <u>no-wa</u> , is: IF $\phi[(\epsilon, x, P(x))] : t$ THEN $\text{make/go}(\langle L^{-1} \rangle); \text{put}(\text{?t}, \text{?}\langle D \rangle(x))$ ELSE ABORT
TREE	for some sequence $\langle T_0, T_1, T_2, \dots, T_{n-1} \rangle$, A_n maps T_{n-1} onto T_n , and T_n is represented as: 

The action encoded as intrinsic to any one word may be considerably more complex than (7.14) and (7.16): such conditional actions may involve nested sub-sequences of actions, according as the word in question provides actions triggered by a number of distinct structural conditions (see the entries of *no* (7.1) and ‘*no* + case particle’ (7.5)).

In what follows, I shall further clarify the ‘form-action-interpretation’ tripartite. Firstly, the ‘interpretation’ in the tripartite (as represented in TREE) is, more strictly, a **contribution** towards interpretation. This is because the execution of A_n will leave some requirements unsatisfied. In (7.14), the node decorated with *Horikoshi*’ is situated in a type- t -requiring tree. Unless further actions are run, outstanding requirements (e.g., $?t$) remain in the tree. The same point is more vividly illustrated in (7.16). The parse of *no-wa* does not introduce any substantial content into the tree but contributes to the gradual development of an interpretation by imposing requirements as constraints.

Secondly, the ‘interpretation’ in the tripartite (or, more precisely, a contribution towards interpretation) is **context-dependent**. This is because the DS framework is not encapsulated, allowing pragmatic inference to interact with DS tree transitions at every step. This is demonstrated in our analysis of *no*-involving constructions; for instance,

the parser decides what type-*e* term it copies as the content of an internal head in processing HIRs online (see Sect. 4.4.1, Chap. 6).

Thirdly, what is the status of the dynamic tripartite? In (7.14) and (7.16), the information in the tripartite may be seen as the **augmented** lexical entry of *Horikoshi* and *no-wa* respectively, ‘augmented’ in that not only actions but also tree growth are specified. As for the schema (7.15), it is an **abstract description** of (7.14). I assume that such abstract descriptions may be stored in the competence of language users, with grammars **varying** according to the number of such descriptions so stored internally to the grammar. Overall, the dynamic tripartite, or the DS notion of construction, covers (i) augmented lexical entries such as (7.14) and (7.16) and (ii) the abstract descriptions of augmented entries such as (7.15) which may be part of competence or may be merely epiphenomenal depending on language users.

These remarks on the dynamic, procedural treatment of construction may be a basis for developing a theory that may be dubbed ‘Dynamic Construction Grammar.’ To get to the end, however, there are lots of issues to be sorted out. First, the proposed view of construction reflects only part of language use: comprehension. A fuller account has to reflect production as well. This is a feasible task since DS has been shown to model comprehension and production based on the same mechanism (see Eshghi et al. 2011 and references therein). Second, how to predict commonalities and differences among constructions systematically? Solutions available in the literature include construction networks (Goldberg 1995, Lakoff 1987, Langacker 2003), type hierarchies in SBCG (Boas & Sag 2013), and template hierarchies in LFG (Asudeh et al. 2013). This issue hinges on how we formally represent the dynamic tripartite. A possible line of thought is to make use of action templates and ‘macros’ (sequences of atomic actions) (Eshghi et al. 2011, Eshghi et al. 2013) for constructing a network or a hierarchy to model relations among constructions. This task raises a number of related issues (e.g., *What are the relations between DS actions?*, *What are the possible classes of DS actions?*,

How are DS actions manipulated so that new constructions emerge?). These insights and questions, I hope, stimulate future studies on construction in the light of language dynamics as an intriguing research agenda.

Appendix

The Dynamic Syntax Formalism

1 Introduction

This appendix introduces the Dynamic Syntax (DS) formalism. We shall largely follow the orthodox literature (Cann et al. 2005, Kempson et al. 2001, Kempson et al. 2011) but depart from it slightly by making some amendments (Seraku 2013b, in press a). The framework is presented relatively informally in Chap. 3. Therefore, this appendix is meant to supplement the content of Chap. 3 in more formal terms.

Sect. 2 replaces DS notations used in the body of the thesis with more rigid descriptions and adds some formal apparatus to the inventory. Sect. 3 then constructs a mini grammar of certain fragments of Japanese by defining an AXIOM and various tree-update actions (i.e., general actions, lexical actions, pragmatic actions).

2 Supplements to Chapter 3

In Chap. 3, each node, if fully developed, is decorated with a pair $\alpha : \beta$, where α stands for a content and β stands for labels (e.g., logical type, tree-node address); if a node is not fully developed, it is annotated with requirements. In more formal terms, each node is decorated with a **DU** (Declarative Unit), a set of **statements** concerning a content, labels, and requirements.

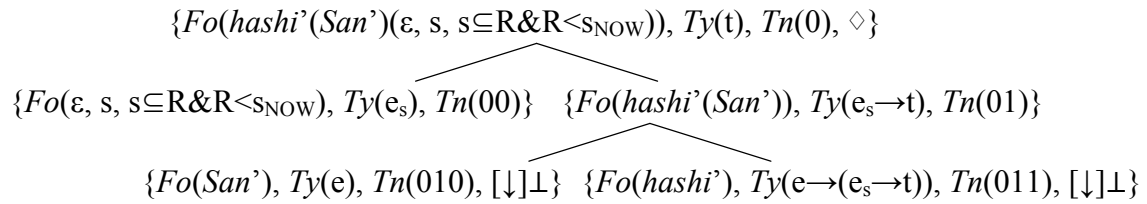
To form statements in a DU, the following three predicates are defined. Firstly, the formula predicate **Fo** takes a content X , and $Fo(X)$ declares that the content X holds at this node. Secondly, the type predicate **Ty** takes a type X , and $Ty(X)$ declares that the type X holds at this node. Finally, the tree-node predicate **Tn** takes a node address X , and $Tn(X)$ declares that the node address X holds at this node.

Further, there is another type of statement to be included in some DUs: the **bottom-restriction** $[\downarrow]\perp$. This reads ‘necessarily below the current node, the falsum $\perp$ holds (i.e., no nodes exist).’ The label $[\downarrow]\perp$ is encoded in content words (e.g., nouns, verbs), ensuring that content words are used to decorate a **terminal** node in a tree.

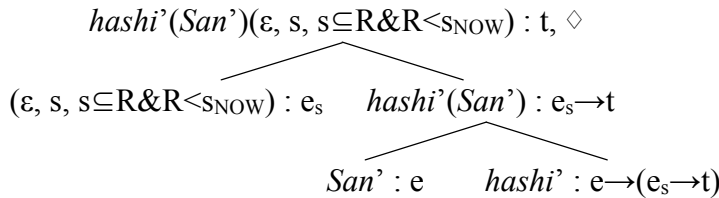
To illustrate the above points, the string (A.1) is mapped onto the tree (A.2). In Chap. 3, the same tree is represented as in (A.3). (In each tree, the internal structure of a situation term is disregarded.)

(A.1) San-ga hashi-tta.
S-NOM run-PAST
‘San ran.’

(A.2) Parsing *San-ga hashi-tta*



(A.3) Parsing *San-ga hashi-tta* (informal display)



The domain of Fo is the **infinite** set $D_{Fo} = \{San', hashi', \dots\}$. This set includes meta-variables (e.g., U, V). The domain of Ty is the **finite** set D_{Ty} whose elements are

pinned down in Table A.1. Since each predicate takes a situation term as an argument, the content of an intransitive verb is a 2-place predicate, the content of a transitive verb is a 3-place predicate, and the content of a ditransitive verb is a 4-place predicate. D_{Ty} is a restricted set, and no operations to generate types (e.g., type-lifting) are stipulated.

Type	Description	Example Expressions
$Ty(e)$	individual term	$Fo(San')$, $Fo(\epsilon, x, forest'(x))$
$Ty(e_s)$	situation term	$Fo(\epsilon, s, s \subseteq R \& R <_{S_{NOW}})$
$Ty(t)$	proposition	$Fo(run'(San'))(\epsilon, s, s \subseteq R \& R <_{S_{NOW}})$
$Ty(e_s \rightarrow t)$	1-place predicate	$Fo(run'(San'))$
$Ty(e \rightarrow (e_s \rightarrow t))$	2-place predicate	$Fo(run')$
$Ty(e \rightarrow (e \rightarrow (e_s \rightarrow t)))$	3-place predicate	$Fo(see')$
$Ty(t \rightarrow (e \rightarrow (e_s \rightarrow t)))$	3-place predicate	$Fo(tell')$
$Ty(e \rightarrow (e \rightarrow (e \rightarrow (e_s \rightarrow t))))$	4-place predicate	$Fo(give')$
$Ty(cn)$	nominal	$Fo(x, apple'(x))$
$Ty(cn_s)$	nominal (situation)	$Fo(s, s \subseteq R \& R <_{S_{NOW}})$
$Ty(cn \rightarrow e)$	quantifier	$Fo(\lambda P.(\epsilon, P))$
$Ty(cn_s \rightarrow e_s)$	quantifier (situation)	$Fo(\lambda S.(\epsilon, S))$
$Ty(e \rightarrow cn)$	restrictor	$Fo(\lambda y.(y, apple'(y)))$
$Ty(e_s \rightarrow cn_s)$	restrictor (situation)	$Fo(\lambda z.(z, z \subseteq R \& R <_{S_{NOW}}))$
$Ty(e_s \rightarrow (e_s \rightarrow cn_s))$	function ¹	$Fo(\lambda y.\lambda z.(z, z \subseteq y \& y <_{S_{NOW}}))$
$Ty(e_s \rightarrow (e_s \rightarrow (e_s \rightarrow cn_s)))$	function ²	$Fo(\lambda x.\lambda y.\lambda z.(z, z \subseteq y \& y <_x))$

Table A.1. Repertoire of logical types

There is a caveat for notations in LOFT (Logic Of Finite Trees) (Blackburn & Meyer-Viol 1994). In the body of the thesis, when there are multiple LOFT operators, only a single pair of angle brackets is used, as in $\langle \downarrow_1 \downarrow_* \rangle$ and $\langle \downarrow D \rangle$. Strictly, however, a pair of angle brackets is required for **each** operator; thus, the more precise notations are $\langle \downarrow_1 \rangle \langle \downarrow_* \rangle$ and $\langle \downarrow \rangle \langle D \rangle$. LOFT operators used in this thesis are listed in Table A.2.

¹ This is a function that takes an underspecified time interval R and outputs a restrictor.

² This is a function that takes the utterance time constant s_{NOW} and outputs the above function (i.e., the function that takes an underspecified time interval R and outputs a restrictor).

LOFT statement	Description
$\langle \downarrow \rangle X$	X holds at the daughter node
$\langle \downarrow_0 \rangle X$	X holds at the argument-daughter node
$\langle \downarrow_1 \rangle X$	X holds at the functor-daughter node
$\langle \downarrow_* \rangle X$	X holds at the current node or somewhere below (but not across a LINK boundary)
$\langle D \rangle X$	X holds somewhere below (may be across a LINK boundary)
$\langle L \rangle X$	X holds at the LINKed node
$\langle \uparrow \rangle X$	X holds at the mother node
$\langle \uparrow_0 \rangle X$	X holds at the mother node from the perspective of the argument-daughter node
$\langle \uparrow_1 \rangle X$	X holds at the mother node from the perspective of the functor-daughter node
$\langle \uparrow_* \rangle X$	X holds at the current node or somewhere above (but not across a LINK boundary)
$\langle U \rangle X$	X holds somewhere above (may be across a LINK boundary)
$\langle L^{-1} \rangle$	X holds at the inversely LINKed node
$\langle \downarrow_+ \rangle X$	X holds somewhere below (but not across a LINK boundary) ³
$[\downarrow]X$	X holds at all daughter nodes ⁴

Table A.2. Repertoire of LOFT operators

So far, we have been concerned with the declarative part of the framework. As for the procedural part, let us clarify tree-update actions. DS actions consist of **atomic actions** such as `go` and `put` (Sect. 3, Chap. 3). A list of atomic actions is presented in Table A.3 (cf., Eshghi et al. 2011: 19). For brevity, I use the notation `make/go(X)` as an abbreviation of the two atomic actions to be executed in batch, as in `make(X); go(X)`.

³ The inverse $\langle \uparrow_+ \rangle$ may be defined, but this operator is not used in this thesis.

⁴ The inverse $[\uparrow]$ may be defined, but this operator is not used in this thesis.

Atomic action	Description
make	to take a LOFT operator as argument and create a node whose position is specified by the LOFT operator
go	to take a LOFT operator as argument and move a pointer $\diamond$ to the node that is specified by the LOFT operator
put	to take a content or a label as argument and posit it at the active node marked by a pointer $\diamond$
freshput	to take a fresh variable as argument and posit it at the active node marked by a pointer $\diamond$
delete	to take a satisfied requirement as argument and delete it
beta-reduce	to perform functional application and type deduction
do	to take an action sequence as argument and re-run it

Table A.3. Repertoire of atomic actions

Finally, the DS notion of context must be formalised. As discussed in Sect. 6, Chap. 3, context reflects (i) the linear processing, (ii) attendant progressive tree growth, and (iii) the actions already executed (Cann et al. 2007, Purver et al. 2006). Let us first define a triple $\langle T, W, A \rangle$, where T is a (possibly, partial) **tree** already built up, W is a sequence of **words** already parsed, and A is a sequence of **actions** which built the tree T . Since the applications of general actions are optional, there may be more than one such triple at any point of tree transitions. So, let us define a parse state S as a **set** of such triples. In the initial parse state S_0 , there is a single triple, where T is specified by an AXIOM, W is empty since no words have been parsed, and A is also empty since no actions have been run. Thus, $S_0 = \{\langle T_0, \varphi, \varphi \rangle\}$, where T_0 is the AXIOM.⁵ S_0 will expand as a string is parsed incrementally. Based on parse states, the DS notion of context is defined as follows: the context C for a partial tree T in a parse state S consists of (i) $S' = \{\dots, \langle T_i, W_i, A_i \rangle, \dots\}$ (i.e., a set of triples already constructed by the parse of the preceding string(s)) and (ii) $\langle T, W, A \rangle$ (i.e., the current triple under development).

⁵ Given the non-encapsulated DS architecture, it is in principle possible to represent pragmatic, non-linguistic information in DS trees, hence part of T , in which case S_0 is more richly set out.

With these minor changes of DS notations and the newly added mechanisms, the next section constructs a mini grammar of Japanese. This mini grammar generates the various structures surveyed in the body of the thesis.

3 A Mini Grammar of Japanese

This section constructs a grammar of our interested fragments of Japanese by defining an AXIOM and a relevant set of tree-update actions. The AXIOM determines the initial state (Sect. 3.1), and it is updated incrementally and monotonically by a combination of general actions (Sect. 3.2), lexical actions (Sect. 3.3), and pragmatic actions (Sect. 3.4).⁶

3.1 An Axiom

(A.4) AXIOM (Sect. 3.1, Chap. 3)

$$\{?Ty(t), Tn(U), ?\exists x.Tn(x), \diamond\}$$

3.2 General Actions

General Actions for Basic Tree-Updates

(A.5) ANTICIPATION (Sect. 4.1, Chap. 3)

IF	$\langle \mu \rangle \exists x. ?x$	
THEN	$go(\langle \mu \rangle)$	
ELSE	ABORT	where $\mu \in \{\downarrow_0, \downarrow_1, \downarrow^*, L\}$

(A.6) THINNING (Sect. 4.1, Chap. 3)

IF	$?X, X$
THEN	$delete(?X)$
ELSE	ABORT

⁶ In formulating some general actions and pragmatic actions, I consulted Eshghi et al. (2011); I also benefited from suggestions made by Stergious Chatzikyriakidis, Arash Eshghi, and Ruth Kempson. In this appendix, I shall disregard QUANTIFIER EVALUATION because (i) it requires some new devices (e.g., scope statement) to be introduced and (ii) it complicates the encoding of some expressions (e.g., common nouns, tense particles). See Cann et al. (2005: Sect. 3.3) and Kempson et al. (2001: Chap. 7) for details.

(A.7) COMPLETION (Sect. 4.1, Chap. 3)

IF	$Ty(X)$	
THEN	IF	$\langle \mu \rangle \exists x.x$
	THEN	$go(\langle \mu \rangle)$
	ELSE	ABORT
ELSE	ABORT	where $\mu \in \{\uparrow_0, \uparrow_1, \uparrow^*, L^{-1}\}$

(A.8) ELIMINATION (Sect. 4.1, Chap. 3)

```

IF      ?Ty(X),  $\neg \exists x.Fo(x)$ 
THEN    IF       $\langle \downarrow_0 \rangle Ty(Y)$ ,  $\neg \langle \downarrow_0 \rangle \exists x.?x$ 
        THEN    IF       $\langle \downarrow_1 \rangle Ty(Y \rightarrow X)$ ,  $\neg \langle \downarrow_1 \rangle \exists x.?x$ 
                THEN    beta-reduce
                ELSE    ABORT
        ELSE    ABORT
ELSE    ABORT

```

General Actions for Structural Underspecification

(A.9) LOCAL *ADJUNCTION (Sect. 3.3, Chap. 3)

IF	$Tn(a), ?Ty(t)$
THEN	$\text{make/go}(\langle \downarrow_1^* \rangle \langle \downarrow_0 \rangle); \text{put}(\text{?Ty}(e), \langle \uparrow_0 \rangle \langle \uparrow_1^* \rangle Tn(a), Tn(U), \text{?}\exists x.Tn(x))$
ELSE	ABORT

(A.10) *ADJUNCTION (Sect. 3.3, Chap. 3)

```

IF       $Tn(\alpha), ?Ty(t)$ 
THEN    $\text{make/go}(\langle \downarrow_* \rangle); \text{put}(\text{?Ty}(t), \langle \uparrow_* \rangle Tn(\alpha), Tn(U), \text{?}\exists x. Tn(x))$ 
ELSE   ABORT

```

(A.11) GENERALISED ADJUNCTION (Sect. 3.3, Chap. 3)

```

IF       $Tn(a), ?Ty(t)$ 
THEN   make/go(<D>); put(?Ty(t), <U> $Tn(a), Tn(U), ?\exists x.Tn(x)$ )
ELSE   ABORT

```

(A.12) UNIFICATION (Sect. 3.3, Chap. 3)

IF	DU, DU'	
THEN	DU $\sqcup$ DU'	
ELSE	ABORT	where $\diamond \in \text{DU}'$

General Actions for LINK Transitions

(A.13) LINK INTRODUCTION (for head-external relatives) (Sect. 5.2.1, Chap. 3)

IF $Ty(t)$
 THEN IF $\langle D \rangle Fo(a)$
 THEN $\text{make/go}(\langle L^{-1} \rangle); \text{put}(?Ty(e), ?\exists z.Fo(z[a]))$
 ELSE ABORT
 ELSE ABORT

(A.14) LINK ENRICHMENT (for head-external relatives) (Sect. 5.2.1, Chap. 3)

IF $Fo(\varepsilon, y, \varphi(y)), Ty(e)$
 THEN IF $\langle L \rangle Fo(\psi[(\varepsilon, x, P(x))])$
 THEN $\text{put}(Fo(\varepsilon, y, \varphi(y) \& \psi[y/(\varepsilon, x, P(x))]))$
 ELSE ABORT
 ELSE ABORT

(A.15) LINK INTRODUCTION (for the *wa*-topic construction) (Sect. 5.3, Chap. 3)⁷

IF $?Ty(t)$
 THEN $\text{make/go}(\langle L \rangle); \text{put}(?Ty(e))$
 ELSE ABORT

(A.16) LINK ENRICHMENT (for clausal coordination) (Sect. 4.3.1, Chap. 4)⁸

IF $Fo(\psi), Ty(t)$
 THEN IF $\langle L^{-1} \rangle Fo(\varphi)$
 THEN $\text{go}(\langle L^{-1} \rangle); \text{put}(Fo(\varphi \& \psi))$
 ELSE ABORT
 ELSE ABORT

(A.17) LINK INTRODUCTION (for clefts, stripping, and sluicing) (Sect. 5.3, Chap. 5)

IF $?Ty(t)$
 THEN $\text{make/go}(\langle L^{-1} \rangle); \text{put}(?Ty(e))$
 ELSE ABORT

⁷ LINK ENRICHMENT (for the *wa*-topic construction) is formulated as part of the lexical entry of the topic marker *wa*. See Sect. 3.3 of this appendix.

⁸ LINK INTRODUCTION (for clausal coordination) is formulated as part of the lexical entry of the clause-linking suffix *i*. See Sect. 3.3 of this appendix.

(A.18) LINK ENRICHMENT (for clefts, stripping, and sluicing) (Sect. 5.3, Chap. 5)

```

IF       $Fo(a), Ty(e)$ 
THEN    IF       $\langle L \rangle ?Ty(t)$ 
          THEN     $go(\langle L \rangle); put(? \langle \downarrow \rangle \langle D \rangle Fo(a))$ 
          ELSE    ABORT
ELSE    ABORT

```

3.3 Lexical Actions

(A.19) Proper name: *San* (Sect. 3.1, Chap. 3)

```

IF       $?Ty(e)$ 
THEN     $put(Fo(San'), Ty(e), [\downarrow] \perp)$ 
ELSE    ABORT

```

(A.20) Noun: *kodama* (= ‘spirit living in a rich forest’) (Sect. 2.2, Chap. 3)⁹

```

IF       $?Ty(e)$ 
THEN     $make/go(\langle \downarrow_1 \rangle); put(Fo(\lambda P.(\beta, P)), Ty(cn \rightarrow e)); go(\langle \uparrow_1 \rangle);$ 
           $make/go(\langle \downarrow_0 \rangle); put(?Ty(cn));$ 
           $make/go(\langle \downarrow_1 \rangle); put(Fo(\lambda y.(y, kodama'(y))), Ty(e \rightarrow cn), [\downarrow] \perp); go(\langle \uparrow_1 \rangle);$ 
           $make/go(\langle \downarrow_0 \rangle); (fresh)put(Fo(x)); put(Ty(e))$ 
ELSE    ABORT where  $\beta \in \{\varepsilon, \tau, \iota\}$ 

```

(A.21) Nominative-case particle: *ga* (Sect. 5.2.2, Chap. 3)

```

IF       $Ty(e)$ 
THEN    IF       $\langle \uparrow_0 \rangle \langle \uparrow_1^* \rangle (Tn(a), ?Ty(t))$ 
          THEN     $put(? \langle \uparrow_0 \rangle Ty(e_s \rightarrow t))$ 
          ELSE    IF       $\langle \uparrow \rangle \perp$ 
                    THEN  $make/go(\langle \uparrow_0 \rangle); put(?Ty(e_s \rightarrow t));$ 
                           $make/go(\langle \uparrow_1 \rangle); put(?Ty(t), Tn(U), ?\exists x.Tn(x));$ 
                           $(make/go(\langle \uparrow^* \rangle); put(?Ty(t), Tn(V), ?\exists y.Tn(y)))$ 
                    ELSE ABORT
ELSE    ABORT

```

⁹ In $(fresh)put(Fo(x))$, the parser needs to choose $freshput(Fo(x))$ or $put(Fo(x))$. The introduction of a non-fresh variable is relevant to the processing of a focus NP in Japanese clefts, an appositive NP in English, and so on (see Sect. 3.3, Chap. 5).

(A.22) Accusative-case particle: *o* (Sect. 5.2.2, Chap. 3)

```

IF      Ty(e)
THEN    IF      <↑0><↑1*>(Tn(α), ?Ty(t))
        THEN    put(?<↑0>Ty(e→(es→t)))
        ELSE    IF      <↑>⊥
                THEN    make/go(<↑0>); put(?Ty(e→(es→t)));
                        make/go(<↑1>); put(?Ty(es→t));
                        make/go(<↑1>); put(?Ty(t), Tn(U), ?∃x.Tn(x));
                        (make/go(<↑*>); put(?Ty(t), Tn(V), ?∃y.Tn(y)))
                ELSE    ABORT
ELSE    ABORT

```

(A.23) Topic particle: *wa* (Sect. 5.3, Chap. 3)

```

IF      Fo(α), Ty(e)
THEN    make/go(<L-1>); put(?Ty(t), ?<↓><D>(Fo(α)))
ELSE    ABORT

```

(A.24) Intransitive verb: *hashi*- (= ‘run’) (Sect. 3.2, Chap. 3)

```

IF      ?Ty(t)
THEN    make/go(<↓0>); put(Fo(U), ?∃x.Fo(x), Ty(es)); go(<↑0>);
        make/go(<↓1>); put(?Ty(es→t));
        make/go(<↓0>); put(Fo(V), ?∃y.Fo(y), Ty(e)); go(<↑0>);
        make/go(<↓1>); put(Fo(hashi'), Ty(e→(es→t)), [↓]⊥)
ELSE    ABORT

```

(A.25) Transitive verb: *mi*- (= ‘see’) (Sect. 3.2, Chap. 3)

```

IF      ?Ty(t)
THEN    make/go(<↓0>); put(Fo(U), ?∃x.Fo(x), Ty(es)); go(<↑0>);
        make/go(<↓1>); put(?Ty(es→t));
        make/go(<↓0>); put(Fo(V), ?∃y.Fo(y), Ty(e)); go(<↑0>);
        make/go(<↓1>); put(?Ty(e→(es→t)))
        make/go(<↓0>); put(Fo(W), ?∃z.Fo(z), Ty(e)); go(<↑0>);
        make/go(<↓1>); put(Fo(mi'), Ty(e→(e→(es→t))), [↓]⊥)
ELSE    ABORT

```

(A.26) Copula: *da*- (Sect. 5.3.3, Chap. 5)

```

IF      ?Ty(t)
THEN    IF      <↓+>⊤
        THEN    put(Fo(U), Ty(t), <L-1>⊥)
        ELSE    put(Fo(U), Ty(t))
ELSE    ABORT

```

(A.27) Past-tense marker: *ta* (Sect. 4.4.1, Chap. 4)

```

IF      Ty(t)
THEN    go(<↓0>);
        make/go(<↓1>); put(Fo(λP.(β, P)), Ty(cns→es)); go(<↑1>);
        make/go(<↓0>); put(?Ty(cns));
        make/go(<↓0>); put(Fo(s), Ty(es)); go(<↑0>);
        make/go(<↓1>); put(?Ty(es→cns));
        make/go(<↓0>); put(Fo(R), Ty(es)); go(<↑0>);
        make/go(<↓1>); put(?Ty(es→(es→cns))));
        make/go(<↓0>); put(Fo(snow), Ty(es)); go(<↑0>);
        make/go(<↓1>); put(Fo(λx.λy.λz.(z, z⊆y&y<x)), Ty(es→(es→(es→cns))))))
ELSE    ABORT

```

where β ∈ {ε, ι}

(A.28) Complementiser: *to* (Sect. 4.2, Chap. 3)

```

IF      Fo(φ), Ty(t)
THEN    make/go(<↑0>); put(?Ty(e→(es→t)));
        make/go(<↑1>); put(?Ty(es→t));
        make/go(<↑1>); put(?Ty(t), <↓1><↓1><↓0>Fo(φ))
ELSE    ABORT

```

(A.29) Clause-linking suffix: *i* (Sect. 4.3.1, Chap. 4)

```

IF      Ty(t)
THEN    IF      <↓*>(Fo(α))
        THEN    make/go(<L>); put(?Ty(t), ?<↓*>(Fo(α), Ty(e)))
        ELSE    ABORT
ELSE    ABORT

```

(A.30) Nominaliser: *no* (Sect. 3.2, Chap. 4)

```

IF      Ty(t)
THEN    IF      Fo(φE[α])
        THEN    make/go(<L-1>); put(Fo(α), Ty(e))
        ELSE    ABORT
ELSE    ABORT

```

(A.31) Cleft-marker: *no-wa* (= *no* + topic particle) (Sect. 3.2, Chap. 5)

```

IF      Fo(φ[(ε, x, P(x))]), Ty(t)
THEN    make/go(<L-1>); put(?Ty(t), ?<↓><D>(Fo(x)))
ELSE    ABORT

```

(A.32) HIR-marker: *no* + case particle (Sect. 4.2, Chap. 6)

```

IF      Ty(t)
THEN    IF      Fo( $\varphi_E[(Fo(\alpha), Ty(e_s)), (Fo(\beta), Ty(e))]$ )
        THEN    make/go( $\langle L^{-1} \rangle$ ); put( $?Ty(t)$ );
            make/go( $\langle \downarrow_0 \rangle$ ); put( $Fo(U), ?\exists x.Fo(x) \& [\alpha \star x], Ty(e_s)$ ); go( $\langle \uparrow_0 \rangle$ );
            make/go( $\mu$ ); put( $Fo(\beta), Ty(e)$ )
        ELSE    ABORT
ELSE    ABORT

```

where $\mu \in \{ \langle \downarrow_1 \rangle \langle \downarrow_0 \rangle, \langle \downarrow_1 \rangle \langle \downarrow_1 \rangle \langle \downarrow_0 \rangle, \langle \downarrow_1 \rangle \langle \downarrow_1 \rangle \langle \downarrow_1 \rangle \langle \downarrow_0 \rangle, \dots \}$

3.4 Pragmatic Actions

(A.33) SUBSTITUTION (for node-address resolution) (Sect. 3.1, Chap. 3)

```

IF      Tn(U),  $?\exists x.Tn(x)$ 
THEN    put( $Tn(\alpha)$ )
ELSE    ABORT      where  $\alpha \in \{0, 00, 01, 000, 001, 010, 011, \dots\}$ 

```

(A.34) SUBSTITUTION (for anaphora/ellipsis resolution) (Sect. 6, Chap. 3)

```

IF      Ty(X),  $?\exists x.Fo(x)$ 
THEN    IF       $\langle T, W, A \rangle \in C, \{Ty(X), Fo(Y)\} \in T$ 
        THEN    put( $Fo(Y)$ )
        ELSE    ABORT
ELSE    ABORT

```

(A.35) REGENERATION (Sect. 6, Chap. 3)¹⁰

```

IF      Ty(X),  $?\exists x.Fo(x)$ 
THEN    IF       $\langle T, W, A \rangle \in C, \langle a_i, \dots, a_{i+n} \rangle \sqsubseteq A,$ 
                 $a_i = \langle \text{IF } \varphi_1, \text{ THEN } \varphi_2, \text{ ELSE ABORT} \rangle, ?Ty(X) \in \varphi_1$ 
        THEN    do( $\langle a_i, \dots, a_{i+n} \rangle$ )
        ELSE    ABORT
ELSE    ABORT

```

¹⁰ The symbol $\sqsubseteq$ is used to represent the sub-sequence relation (Purver et al. 2006: endnote 28).

References

- Akmajian, A. 1970. On deriving cleft sentences from pseudo-cleft sentences. *Linguistic Inquiry* **1** (1), 149-68.
- Asudeh, A. 2006. Direct compositionality and the architecture of LFG. In Butt, M. et al. (eds.) *Intelligent Linguistic Architectures*. Stanford: CSLI.
- Asudeh, A., Dalrymple, M., and Toivonen, I. 2013. Constructions with lexical integrity. *Journal of Language Modelling* **1** (1), 1-53.
- Atlas, J. & Levinson, S. 1981. *It*-clefts, informativeness, and logical form. In Cole, P. (ed.) *Radical Pragmatics*. New York: Academic Press.
- Barwise, J. 1981. Scenes and other situations. *Journal of Philosophy* **78** (7), 369-97.
- Blackburn, P. & Meyer-Viol, W. 1994. Linguistics, logic, and finite trees. *Bulletin of Interest Group of Pure and Applied Logics* **2**, 2-39.
- Boas, H. & Sag, I. (eds.) 2013. *Sign-Based Construction Grammar*. Stanford: CSLI.
- Bouzouita, M. 2008. At the syntax-pragmatics interface. In Cooper, R. & Kempson, R. (eds.) *Language in Flux*. London: College Publications.
- Bresnan, J. 2001. *Lexical-Functional Syntax*. Oxford: Blackwell.
- Butt, M. & King, T. H. 2000. Null elements in discourse structure. In Subbarao, K. V. (ed.) *Papers from the NULLS Seminar*. Delhi: Motilal Banarasidas.
- Cann, R. 2006. Semantic underspecification and the pragmatic interpretation of *be*. In von Stechow, P. & Turner, K. (eds.) *Where Semantics Meets Pragmatics*. Oxford: Elsevier.
- Cann, R. 2011. Towards an account of the auxiliary system in English. In Kempson, R. et al. (eds.) *The Dynamics of Lexical Interfaces*. Stanford: CSLI.
- Cann, R. & Kempson, R. 2008. Production pressures, syntactic change and the emergence of clitic pronouns. In Cooper, R. & Kempson, R. (eds.) *Language in Flux*. London: College Publications.
- Cann, R., Kempson, R., and Marten, L. 2005. *The Dynamics of Language*. Oxford: Elsevier.

- Cann, R., Kempson, R., Marten, L., and Otsuka, M. 2005. Right node raising, coordination and the dynamics of language processing. *Lingua* **115** (4), 502-25.
- Cann, R., Kempson, R., and Purver, M. 2007. Context-dependent well-formedness. *Research on Language and Computation* **5** (3), 333-58.
- Cann, R. & Wu, Y. 2011. The *bei* construction in Chinese. In Kempson, R. et al. (eds.) *The Dynamics of Lexical Interfaces*. Stanford: CSLI.
- Carston, R. 2002. *Thoughts and Utterances*. Oxford: Blackwell.
- Charnavel, I. 2010. The problems of quantifiers in object position. In Otsu, Y. (ed.) *Proceedings of the Tenth Tokyo Conference on Psycholinguistics*. Tokyo: Hituzi.
- Chatzikyriakidis, S. 2009. Clitics in Grecia Salentina Greek. *Lingua* **119** (12), 1939-68.
- Chierchia, G. & McConnell-Ginet, S. 2000. *Meaning and Grammar*, 2nd edition. Cambridge, MA: MIT Press.
- Cho, S., Whitman, J., and Yanagida, Y. 2008. Clefts in Japanese and Korean. In Bane, M. et al. (eds.) *Proceedings from the Annual Meeting of Chicago Linguistics Society* **44** (1). Chicago: Chicago Linguistics Society.
- Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton.
- Chomsky, N. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.
- Chomsky, N. 1977. On *wh*-movement. In Culicover, P. et al. (eds.) *Formal Syntax*. New York: Academic Press.
- Chomsky, N. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.
- Chomsky, N. 1986. *Knowledge of Language*. New York: Praeger.
- Chomsky, N. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.
- Chomsky, N. 2007. Approaching UG from below. In Sauerland, U. & Gärtner, H. M. (eds.) *Interfaces + Recursion = Language?*. New York: Mouton de Gruyter.
- Chung, C. & Kim, J.-B. 2003. Differences between externally and internally headed relative clause constructions. In Kim, J.-B. & Wechsler, S. (eds.) *Proceedings of the 9th International Conference on Head-driven Phrase Structure Grammar*. Stanford: CSLI.
- Chung, S., Ladusaw, W., and McCloskey, J. 1995. Sluicing and logical form. *Natural Language Semantics* **3** (3), 1-44.
- Clark, A. & Lappin, S. 2011. *Linguistic Nativism and the Poverty of the Stimulus*. Oxford: Wiley-Blackwell.
- Cole, P. 1987. The structure of internally headed relative clauses. *Natural Language and Linguistic Theory* **5** (2), 277-302.

- Comrie, B. & Thompson, S. 2007. Lexical nominalization. In Shopen, T. (ed.) *Language Typology and Syntactic Description*, Vol. 3. Cambridge: Cambridge University Press.
- Cooper, R. 1979. The interpretation of pronouns. In Henry, F. & Schnelle, H. (eds.) *Syntax and Semantics*, Vol. 10. New York: Academic Press.
- Croft, W. 2001. *Radical Construction Grammar*. Oxford: Oxford University Press.
- Culicover, P. & Jackendoff, R. 2005. *Simpler Syntax*. Oxford: Oxford University Press.
- Dalrymple, M. 2001. *Lexical Functional Grammar*. New York: Academic Press.
- Davidson, D. 1967. The logical form of action sentences. In Rescher, N. (ed.) *The Logic of Decision and Action*. Pittsburgh: University of Pittsburgh Press. Reprinted in Davidson, D. 1980. *Essays on Actions and Events*. Oxford: Oxford University Press.
- Declerck, R. 1984. The pragmatics of it-clefts and WH-clefts. *Lingua* **64** (4), 251-89.
- Declerck, R. 1988. *Studies on Copular Sentences, Clefts, and Pseudo-Clefts*. Dordrecht: Foris.
- Dryer, M. S. 2011. Order of relative clause and noun. In Dryer, M. S. & Haspelmath, M. (eds.) *The World Atlas of Language Structures Online*. Munich: Max Planck Digital Library, Chapter 90. Available online: <http://wals.info/chapter/90> Accessed on 2013-06-24.
- Erlewine, M. Y. & Gould, I. in press. Interpreting Japanese head-internal relative clauses. *Proceedings of the 31th West Coast Conference on Formal Linguistics*. Somerville, MA: Cascadilla Press.
- Erteschik-Shir, N. 2007. *Information Structure*. Oxford: Oxford University Press.
- Eshghi, A., Purver, M., and Hough, J. 2011. DyLan: Parser for Dynamic Syntax. *Technical Report EECSRR-11-05*. London: School of Electronic Engineering and Computer Science, Queen Mary University of London.
- Eshghi, A., Purver, M., Hough, J., and Sato, Y. 2013. Probabilistic grammar induction in an incremental semantic framework. In Duchier, D. & Parmentier, Y. (eds.) *Constraint Solving and Language Processing (Lecture Notes in Computer Science 8114)*. Berlin: Springer-Verlag.
- Evans, G. 1977. Pronouns, quantifiers, and relative clauses (1). *Canadian Journal of Philosophy* **7** (3), 467-536.
- Evans, G. 1980. Pronouns. *Linguistic Inquiry* **11** (2), 337-62.
- Falk, Y. 2001a. *Lexical-Functional Grammar*. Stanford: CSLI.
- Falk, Y. 2001b. Constituent structure and grammatical functions in the Hebrew action nominal. In Butt, M. & King, T. H. (eds.) *Proceedings of the LFG 2001 Conference*. Stanford: CSLI.

- Fiengo, R. 1977. On trace theory. *Linguistic Inquiry* 8 (1), 35-62.
- Fillmore, C. J. 1985. Syntactic intrusions and the notion of grammatical construction. *Proceedings of the Annual Meeting of the Berkeley Linguistics Society* 11, 73-86.
- Fillmore, C. J., Kay, P., and O'Connor, M. C. 1988. Regularity and idiomaticity in grammatical constructions. *Language* 64 (3), 501-38.
- Fodor, J. 1983. *The Modularity of Mind*. Cambridge, MA: MIT Press.
- Frazier, L. & Fodor, J. D. 1978. The sausage machine. *Cognition* 6 (4), 291-325.
- Frellesvig, B. 2010. *A History of the Japanese Language*. Cambridge: Cambridge University Press.
- Frellesvig, B. & Whitman, J. 2011. Prenominal complementizers and the derivation of complex NPs in Japanese and Korean. In McClure, W. & Den Dikken, M. (eds.) *Japanese/Korean Linguistics* 18. Stanford: CSLI.
- Fujita, Y. & Yamazaki, M. (eds.) 2006. *Fukugouji Kenkyuu no Genzai*. (Current Topics in the Study of Japanese Compound Particles) Osaka: Izumi Shoin.
- Fukaya, T. 2003. Island (in)sensitivity in Japanese sluicing and stripping and some implications. In Garding, G & Tsujimura, M. (eds.) *Proceedings of the 22nd West Coast Conference on Formal Linguistics*. Somerville, MA: Cascadilla Press.
- Fukaya, T. 2007. *Stripping and Sluicing in Japanese and some Implications*. Ph.D. thesis, University of Southern California.
- Fukaya, T. 2012. Island sensitivity in Japanese sluicing and some implications. In Merchant, J. & Simpson, A. (eds.) *Sluicing in Cross-linguistic Perspective*. Oxford: Oxford University Press.
- Fukaya, T. & Hoji, H. 1999. Stripping and sluicing in Japanese and some implications. In Bird, S. et al. (eds.) *Proceedings of the 18th West Coast Conference on Formal Linguistics*. Somerville, MA: Cascadilla Press.
- Fukui, N. & Sakai, H. 2003. The visibility guideline for functional categories. *Lingua* 113 (4/6), 321-75.
- Ginzburg, J. & Cooper, R. 2004. Clarification, ellipsis, and the nature of contextual updates in dialogue. *Linguistics and Philosophy* 27 (3), 297-365.
- Goldberg, A. E. 1995. *Constructions*. Chicago: University of Chicago Press.
- Goldberg, A. E. & Casenhiser, M. 2007. English constructions. In Aarts, B. & McMahon, A. (eds.) *The Handbook of English Linguistics*. Oxford: Wiley-Blackwell.
- Graff, P. & van Urk, C. (eds.) 2012. *Chomsky's Linguistics*. Cambridge, MA: MIT Press.

- Gregoromichelaki, E. 2006. *Conditionals in Dynamic Syntax*. Ph.D. thesis, King's College London.
- Gregoromichelaki, E. 2011. Conditionals in Dynamic Syntax. In Kempson, R. et al. (eds.) *The Dynamics of Lexical Interfaces*. Stanford: CSLI.
- Gregoromichelaki, E., Kempson, R., Purver, M., Mills, G., Cann, R., Meyer-Viol, W., and Healey, P. 2011. Incrementality and intention-recognition in utterance processing. *Dialogue and Discourse* **2** (1), 199-233.
- Grice, P. 1975. Logic and conversation. In Cole, P. & Morgan, J. (eds.) *Syntax and Semantics*, Vol. 3. New York: Academic Press. Reprinted in Grice (1989).
- Grice, P. 1978. Further notes on logic and conversation. In Cole, P. (ed.) *Syntax and Semantics*, Vol. 9. New York: Academic Press. Reprinted in Grice (1989).
- Grice, P. 1989. *Studies in the Way of Words*. Cambridge, MA: Harvard University Press.
- Grosu, A. 2010. The status of the internally-headed relatives of Japanese/Korean within the typology of *definite* relatives. *Journal of East Asian Linguistics* **19** (3), 231-74.
- Grosu, A. & Landman, F. 1998. Strange relatives of the third kind. *Natural Language Semantics* **6** (2), 125-70.
- Grosu, A. & Landman, F. 2012. A quantificational disclosure approach to Japanese and Korean internally headed relatives. *Journal of East Asian Linguistics* **21** (2), 159-96.
- Gunji, T. 2007. Japanese pseudo cleft sentences and discourse representation semantics. *Theoretical and Applied Linguistics at Kobe Shoin* **10**, 17-32.
- Hankamer, J. & Sag, I. 1976. Deep and surface anaphora. *Linguistic Inquiry* **7** (3), 391-426.
- Harada, S.-I. 1971. Ga-no conversion and idiolectal variations in Japanese. *Gengo Kenkyuu* **60**, 25-38.
- Harada, S.-I. 1973. Counter equi NP deletion. *Annual Bulletin of Research Institute of Logopedics and Phoniatrics* **7**. Tokyo: University of Tokyo.
- Harbour, D. 2008. Klivaj predika, or predicate clefts in Haitian. *Lingua* **118** (7), 853-71.
- Hasebe, Y. 2007. Eigo to Nihongo no bunretsubun. (Cleft sentences in English and Japanese) In Yamanashi, M. (ed.) *Ninchi Gengogaku Ronkou* **6**. Tokyo: Hituzi.
- Hasegawa, N. 1988. Remarks on 'zero-pronominals.' In Tawa, W. & Nakayama, M. (eds.) *Proceedings of Japanese Syntax Workshop*. CT: Connecticut College.
- Hasegawa, N. 1997. A copula-based analysis of Japanese clefts. In Inoue, K. (ed.) *Grant-in-Aid for COE Research Report (1)*. Chiba: Center for Language Sciences, Kanda University of International Studies.

- Hasegawa, N. 2002. Shuyoubu naizaigata kankeisetsu. (Head-internal relatives) *Scientific Approaches to Language* **1**, 1-33.
- Heim, I. 1990. E-type pronouns and donkey anaphora. *Linguistics and Philosophy* **13** (2), 137-77.
- Heim, I. & Kratzer, A. 1998. *Semantics in Generative Grammar*. Oxford: Blackwell.
- von Heusinger, K. & Egli, U. (eds.) 2000. *Reference and Anaphoric Relations*. Dordrecht: Kluwer.
- Heycock, C. 2008. Japanese *wa*, *ga*, and information structure. In Miyagawa, S. & Saito, M. (eds) *The Oxford Handbook of Japanese Linguistics*. Oxford: Oxford University Press.
- Heycock, C. & Kroch, A. 1999. Pseudocleft connectivity. *Linguistic Inquiry* **30** (3), 365-97.
- Higginbotham, J. 1985. On semantics. *Linguistics Inquiry* **16** (4), 547-93.
- Higginbotham, J. 2000. On events in linguistic semantics. In Higginbotham, J. et al. (eds.) *Speaking of Events*. Oxford: Oxford University Press.
- Hilbert, D. & Bernays, P. 1939. *Grundlagen der Mathematik II*. Berlin: Julius Springer.
- Hiraiwa, K. 2012. The mechanism of inverted relativization in Japanese. *Journal of Linguistics* **48** (2), 345-88.
- Hiraiwa, K. & Ishihara, S. 2002. Missing links. In Ionin, T. et al. (eds.) *MIT Working Papers in Linguistics* **43**. Cambridge, MA: MIT Press.
- Hiraiwa, K. & Ishihara, S. 2012. Syntactic metamorphosis. *Syntax* **15** (2), 142-80.
- Hoffmann, T. & Trousdale, G. (eds.) 2013. *The Oxford Handbook of Construction Grammar*. Oxford: Oxford University Press.
- Hoji, H. 1990. *Theories of Anaphora and Aspects of Japanese Syntax*. Ms., University of Southern California.
- Hoji, H. 2003. Falsifiability and repeatability in generative grammar. *Lingua* **113** (4-6), 377-446.
- Horie, K. 1998. On the polyfunctionality of the Japanese particle *no*. In Ohori, T. (ed.) *Studies in Japanese Grammaticalization*. Tokyo: Kuroshio.
- Horie, K. 2008. The grammaticalization of nominalizers in Japanese and Korean. In López-Couso, M. J. & Seoane, E. (eds.) *Rethinking Grammaticalization*. Amsterdam: John Benjamins.
- Horn, L. 1981. Exhaustiveness and the semantics of clefts. *Proceedings of the Eleventh Annual New England Linguistics Society Conference*. Amherst, MA: University of Massachusetts.

- Horn, L. 1984. Towards a new taxonomy for pragmatic inference. In Schiffrin, D. (ed.) *Meaning, Form, and Use in Context*. Washington, DC.: Georgetown University Press.
- Hoshi, K. 1995. *Structural and Interpretive Aspects of Head-Internal and Head-External Relative Clauses*. Ph.D. thesis, University of Rochester.
- Hoshi, K. 1996. Multiple case-checking in the head-internal relativization in Japanese. In Koizumi, M. et al. (eds.) *MIT Working Papers in Linguistics* **29**. Cambridge, MA: MIT Press.
- Hoshi, K. 2002. The nominalist approach to the head-internal relative clause in Japanese revisited. *Language, Culture and Communication* **28**, 1-32.
- Howes, C. 2012. *Coordinating in Dialogue*. Ph.D. thesis, Queen Mary University of London.
- Ikawa, H. 2012. *Ibento Imiron to Nichieigo no Koubun*. (Event Semantics and Constructions in Japanese and English) Tokyo: Kuroshio.
- Imada, M. 2009. *Nihongo Meishiku no Imironteki Kinouronteki Bunseki*. (Semantical and Functional Analyses of Noun Phrases in Japanese) Ph.D. thesis, University of Tsukuba.
- Inoue, K. 1976. *Henkeibunpou to Nihongo*. (Transformational Grammar and Japanese) Tokyo: Taishukan.
- Ishii, Y. 1989. Head-internal relative clauses in Japanese. *ESCOL* '88, 234-45.
- Ishii, Y. 1990. 'Half-relatives' and syntactic operator movement in Japanese. In Green, T. & Uziel, S. (eds.) *MIT Working Papers in Linguistics* **12**. Cambridge, MA: MIT Press.
- Ito, J. 1986. Head-movement at LF and PF. *University of Massachusetts Occasional Papers in Linguistics* **11**, 109-38.
- Kahraman, B. 2011. *Nihongo oyobi Torukogo niokeru 'Kuusho to Maigo no Izonkankei' no Shori*. (Processing 'Gap-filler Dependencies' in Japanese and Turkish) Ph.D. thesis, Hiroshima University.
- Kahraman, B., Sato, A., Ono, H., and Sakai, H. 2011a. Incremental processing of gap-filler dependencies. In Otsu, Y. (ed.) *Proceedings of the Twelfth Tokyo Conference on Psycholinguistics*. Tokyo: Hituzi.
- Kahraman, B., Sato, A., Ono, H., and Sakai, H. 2011b. Why object clefts are easier to process than subject clefts in Japanese? *Technical Report of IECIE* **111** (170), 67-72.
- Kamide, Y. 2006. Incrementality in Japanese sentence processing. In Nakayama, M. et al. (eds.) *The Handbook of East Asian Psycholinguistics*, Vol. 2. Cambridge: Cambridge University Press.
- Kamio, A. 1983. Meishiku no kouzou. (Structures of noun phrases) In Inoue, K. (ed.) *Nihongo no Kouzou*. Tokyo: Sanseido.

- Kamp, H. & Reyle, U. 2003. *From Discourse to Logic*. Dordrecht: Kluwer.
- Kang, B. 2006. Some peculiarities of Korean *kes* cleft constructions. *Studia Linguistica* **60** (3), 251-81.
- Kaplan, R. M. & Zaenen, A. 1989. Long-distance dependencies, constituent structure, and functional uncertainty. In Baltin, M. & Kroch, A. (eds.) *Alternative Conceptions of Phrase Structure*. Chicago, IL: University of Chicago Press.
- Kaplan, T. & Whitman, J. 1995. The category of relative clauses in Japanese, with reference to Korean. *Journal of East Asian Linguistics* **4** (1), 29-58.
- Kempson, R., Cann, R., and Kiaer, J. 2006. Topic, focus and the structural dynamics of language. In Molnár, V. & Winkler, S. (eds.) *The Architecture of Focus*. Berlin: Mouton de Gruyter.
- Kempson, R., Eshghi, A., Cann, R., Gregoromichelaki, E., and Purver, M. to appear. Ellipsis. In Lappin, S. & Fox, C. (eds.) *The Handbook of Contemporary Semantic Theory*, 2nd edition. Oxford: Wiley-Blackwell.
- Kempson, R. & Gabbay, D. 1998. Crossover. *Journal of Linguistics* **34** (1), 73-124.
- Kempson, R., Gregoromichelaki, E., and Howes, C. (eds.) 2011. *The Dynamics of Lexical Interfaces*. Stanford: CSLI.
- Kempson, R. & Kiaer, J. 2010. Multiple long-distance scrambling. *Journal of Linguistics* **46** (1), 127-92.
- Kempson, R. & Kurosawa, A. 2009. At the syntax-pragmatics interface. In Hoshi, H. (ed.) *The Dynamics and Mechanism of Language*. Tokyo: Kuroshio.
- Kempson, R., Meyer-Viol, W., and Gabbay, D. 1999. VP ellipsis. In Lappin, S. & Benmamoun, E. (eds.) *Fragments*. Oxford: Blackwell.
- Kempson, R., Meyer-Viol, W., and Gabbay, D. 2001. *Dynamic Syntax*. Oxford: Blackwell.
- Kempson, R., Thwala, N., and Marten, L. 2011. SiSwati clefts. In Kempson, R. et al. (eds.) *The Dynamics of Lexical Interfaces*. Stanford: CSLI.
- Kiaer, J. in press. *Pragmatic Syntax*. London: Bloomsbury Publishing.
- Kim, J. & Lee, C. 2008. Why multiple clefts are disallowed. In Chang, C. B. & Haynie, H. J. (eds.) *Proceedings of the 26th West Coast Conference on Formal Linguistics*. Somerville, MA: Cascadilla Press.
- Kim, M. J. 2007. Formal linking in internally headed relatives. *Natural Language Semantics* **15** (4), 279-315.
- Kim, M. J. 2008a. Relevance of grammar and pragmatics to the Relevancy Condition. *Language Research* **44** (1), 95-120.

- Kim, M. J. 2008b. *Event Structure and Internally-headed Relative Clauses*. Saarbrücken: VDM Verlag Dr. Müller Aktiengesellschaft & Co. KG.
- Kim, M. J. 2009. E-type anaphora and three types of *kes*-construction in Korean. *Natural Language and Linguistic Theory* **27** (2), 345-77.
- Kimball, J. 1973. Seven principles of surface structure parsing in natural language. *Cognition* **2** (1), 15-47.
- King, T. H. 1997. Focus domains and information structure. In Butt, M. & King, T. H. (eds.) *Proceedings of the LFG 1997 Conference*. Stanford: CSLI.
- Kinsui, S. 1995. Nihongo-kara mita joshi. (Particles from a historical perspective) *Gengo* **24**, 78-84.
- Kiss, K. 1998. Identificational focus versus informational focus. *Language* **74** (2), 245-73.
- Kitagawa, C. 2005. Typological variations of head-internal relatives in Japanese. *Lingua* **115** (9), 1243-76.
- Kitagawa, C. & Ross, C. 1982. Prenominal modification in Chinese and Japanese. *Linguistic Analysis* **9** (1), 19-53.
- Kizu, M. 2005. *Cleft Constructions in Japanese Syntax*. New York: Palgrave.
- Koizumi, M. 1994. Secondary predicates. *Journal of East Asian Linguistics* **3** (1), 25-79.
- Koizumi, M. 1995. *Phrase Structure in Minimalist Syntax*. Ph.D. thesis, MIT. Published as Koizumi, M. 1999. *Phrase Structure in Minimalist Syntax*. Tokyo: Hituzi.
- Koizumi, M. 2000. String vacuous overt verb raising. *Journal of East Asian Linguistics* **9** (3), 227-85.
- Kondo, Y. 2000. *Nihongo Kijutsu Bunpou no Riron*. (Theory of Japanese Descriptive Grammar) Tokyo: Hituzi.
- Kratzer, A. 1995. Stage-level and individual-level predicates. In Carlson, G. & Pelletier, J. (eds.) *The Generic Book*. Chicago: University of Chicago Press.
- Kratzer, A. 1998. More structural analogies between pronouns and tenses. In Strolavitch, D. & Lawson, A. (eds.) *Proceedings of the 8th Semantics and Linguistics Theory Conference*. Ithaca, NY: Cornell University.
- Kratzer, A. 2011. Situations in natural language semantics. In Zalta, N. E. (ed.) *The Stanford Encyclopedia of Philosophy* (Fall 2011 Edition). Available at: <http://plato.stanford.edu/archives/fall2011/entries/situations-semantics/>
- Kubota, Y. & Smith, E. A. 2007. The Japanese internally headed relative clause is not an E-type pronoun. In Miyamoto, Y. & Ochi, M. (eds.) *MIT Working Papers in Linguistics* **55**. Cambridge, MA: MIT Press.

- Kumamoto, C. 1989. Nichi/Eigo no bunretsubun ni tsuite. (English and Japanese cleft sentences) *Saga Daigaku Eibungaku Kenkyuu* 17, 11-34.
- Kuno, S. 1973a. *The Structure of the Japanese Language*. Cambridge, MA: MIT Press.
- Kuno, S. 1973b. *Nihon Bunpou Kenkyuu*. (A Study of Japanese Grammar) Tokyo: Taishukan.
- Kuroda, S.-Y. 1970. Remarks on the notion of subject with reference to words like *also*, *even* or *only*, Part II. *Annual Bulletin of Research Institute of Logopedics and Phoniatorics* 4. Tokyo: University of Tokyo. Reprinted in Kuroda (1992).
- Kuroda, S.-Y. 1974. Pivot-independent relative clauses in Japanese, I. *Papers in Japanese Linguistics* 3, 59-93. Reprinted in Kuroda (1992).
- Kuroda, S.-Y. 1975-76. Pivot-independent relative clauses in Japanese, II. *Papers in Japanese Linguistics* 4, 85-96. Reprinted in Kuroda (1992).
- Kuroda, S.-Y. 1976-77. Pivot-independent relative clauses in Japanese, III. *Papers in Japanese Linguistics* 5, 157-79. Reprinted in Kuroda (1992).
- Kuroda, S.-Y. 1986. Movement of noun phrases in Japanese. In Imai, T. & Saito, M. (eds.) *Issues in Japanese Linguistics*. Dordrecht: Foris. Reprinted in Kuroda (1992).
- Kuroda, S.-Y. 1992. *Japanese Syntax and Semantics*. Dordrecht: Kluwer.
- Kuroda, S.-Y. 1998. Shubunaizaigata kankeisetsu. (Head-internal relative clauses.) In Hirano, H. & Nakamura, M. (eds.) *Gengo no Naizai to Gaizai*. Reprinted in Kuroda (2005).
- Kuroda, S.-Y. 1999. Tokoro setsu. (Tokoro clauses.) In Kuroda, S.-Y. & Nakamura, M. (eds.) *Kotoba no Kaku to Shuuhen*. Tokyo: Kuroshio. Reprinted in Kuroda (2005).
- Kuroda, S.-Y. 2005. *Nihongo-kara Mita Seisei Bunpou*. (Generative Grammar from the Perspective of Japanese) Tokyo: Iwanami.
- Kurosawa, A. 2003. *On the Interaction of Syntax and Pragmatics*. Ph.D. thesis, King's College London.
- Kuwabara, K. 1996. Multiple *wh*-phrases in elliptical clauses and some aspects of clefts with multiple foci. In Koizumi, M. et al. (eds.) *MIT Working Papers in Linguistics* 29. Cambridge, MA: MIT Press.
- Lakoff, G. 1987. *Women, Fire, and Dangerous Things*. Chicago: University of Chicago Press.
- Lambrecht, K. 1996. *Information Structure and Sentence Form*. Cambridge: Cambridge University Press.
- Langacker, R. 1987. *Foundations of Cognitive Linguistics*, Vol. 1. Stanford: Stanford University Press.

- Langacker, R. 1991. *Foundations of Cognitive Linguistics*, Vol. 2. Stanford: Stanford University Press.
- Langacker, R. 2000. A dynamic usage-based model. In Barlow, M. & Kemmer, S. (eds.) *Usage-based Models of Language*. Stanford: CSLI.
- Langacker, R. 2003. Constructions in Cognitive Grammar. *English Linguistics* **20** (1), 41-83. Reprinted in Langacker, R. 2009. *Investigations in Cognitive Grammar*. New York/Berlin: Mouton de Gruyter.
- Levinson, S. 2000. *Presumptive Meanings*. Cambridge, MA: MIT Press.
- Li, C. & Thompson, S. A. 1989. *Mandarin Chinese*. Berkeley: University of California Press.
- Li, K. 2008. Contrastive focus structure in Mandarin Chinese. *Proceedings of the 20th North American Conference on Chinese Linguistics*, 759-774.
- Lichtenberk, F. 2011. Nominalizations in Toqabaqita and closely related languages. In Yap, F. H. et al. (eds.) *Nominalization in Asian Languages*. Amsterdam: John Benjamins.
- Link, G. 1983. The logical analysis of plurals and mass terms. In Bauerle, R. et al. (eds.) *Meaning, Use, and the Interpretation of Language*. Berlin: Walter de Gruyter.
- Lyons, J. 1977. *Semantics*, Vol. 2. Cambridge: Cambridge University Press.
- Makino, S. 1968. *Some Aspects of Japanese Nominalization*. Kanagawa: Tokai University Press.
- Marten, L. 2002. *At the Syntax-Pragmatics Interface*. Oxford: Oxford University Press.
- Martin, S. E. 1975. *A Reference Grammar of Japanese*. New Haven and London: Yale University Press.
- Matsuda, Y. 1997. A syntactic analysis of focus sentences in Japanese. In Bruening, B. (ed.) *MIT Working Papers in Linguistics* **31**. Cambridge, MA: MIT Press.
- Matsuda, Y. 2000. An asymmetry in copula sentences. *Gengo Kenkyuu* **117**, 3-36.
- Matsuda, Y. 2002. Event sensitivity of head-internal relatives in Japanese. Akatsuka, N. et al. (eds.) *Japanese/Korean Linguistics* **10**. Stanford: CSLI.
- McGloin, N. 1985. No-pronominalization in Japanese. *Papers in Japanese Linguistics* **10**, 1-15.
- Merchant, J. 2001. *The Syntax of Silence*. Oxford: Oxford University Press.
- Michaelis, L. 2009. Sign-Based Construction Grammar. In Heine, B. & Narrog, H. (eds.) *The Oxford Handbook of Linguistic Analysis*. Oxford: Oxford University Press.
- Mihara, K. 1992. *Jiseikaishaku to Tougogenshou*. (Interpretation of Tense and Grammatical Phenomena) Tokyo: Kuroshio.

- Mihara, K. 1994. *Nihongo no Tougo Kouzou*. (Syntactic Structure of Japanese) Tokyo: Shohakusha.
- Mihara, K. & Hiraiwa, K. 2006. *Shin Nihongo no Tougo Kouzou*. (Syntactic Structure of Japanese, New edition.) Tokyo: Shohakusha.
- Miyagawa, S. 1989. *Structure and Case Marking in Japanese*. New York: Academic Press.
- Miyamoto, E. 2002. Case markers as clause boundary inducers in Japanese. *Journal of Psycholinguistic Research* **31** (4), 307-47.
- Miyamoto, E. 2008. Processing sentences in Japanese. In Miyagawa, S. & Saito, M. (eds.) *The Oxford Handbook of Japanese Linguistics*. Oxford: Oxford University Press.
- Montague, R. 1973. The proper treatment of quantification in ordinary English. In Hintikka, I. et al. (eds.) *Approaches to Natural Language*. Dordrecht: Reidel.
- Morey, S. 2011. Nominalization in Numhpuk Singpho. In Yap, F. H. et al. (eds.) *Nominalization in Asian Languages*. Amsterdam: John Benjamins.
- Mulders, I. 2005. Transparent parsing. In McGinnis, M. & Richards, N. (eds.) *MIT Working Papers in Linguistics* **49**. Cambridge, MA: MIT Press.
- Murasugi, K. 1991. *Noun Phrases in Japanese and English*. Ph.D. thesis, University of Connecticut.
- Murasugi, K. 1994. Head internal relative clauses as adjunct pure complex NPs. In Nakano, H. et al. (eds.) *Synchronic and Diachronic Approaches to Language*. Tokyo: Liber Press.
- Murasugi, K. 1996. Head-internal relative clauses and the pro-drop parameter. *Kinjo Daigaku Ronshuu* **37**, 327-50.
- Nakau, M. 1973. *Sentential Complementation in Japanese*. Tokyo: Kaitakusha.
- Nishi, Y. 2006. The emergence of the complementizer *no* in Japanese revisited. In Vance, T. & Jones, K. (eds.) *Japanese/Korean Linguistics* **14**. Stanford: CSLI.
- Nishina, A. & Yoshimura, N. 2005. Hobunhyoushiki no shutsugen. (Emergence of complementizer) *Kokusai Kankei Hikaku Bunka Kenkyuu* **3**, 267-81.
- Nishiyama, K., Whitman, J., and Yi, E.-Y. 1996. Syntactic movement of overt *wh*-phrases in Japanese and Korean. In Akatsuka, N. et al. (eds.) *Japanese/Korean Linguistics* **5**. Stanford: CSLI.
- Nishiyama, Y. 2003. *Nihongo Meishiku no Imiron to Goyouron*. (Semantics and Pragmatics of Noun Phrases in Japanese) Tokyo: Hituzi.
- Noda, T. 1996. *Wa to Ga*. ('Wa' and 'Ga') Tokyo: Kuroshio.
- Ogihara, T. 1999. The semantics of tense and aspect in Japanese. In Tsujimura, N. (ed.) *The Handbook of Japanese Linguistics*. Oxford: Blackwell.

- Ohara, K. 1994. An event-reporting relative construction in Japanese. *Proceedings of the Annual Meeting of the Berkeley Linguistics Society* **20** (1), 260-72.
- Ohori, T. 1994-95. Problems of Japanese HIRCs. *Language, Information, Text* **2**, 89-108.
- Okutsu, K. 1974. *Seisei Nihon Bunpouron*. (A Generative Study of the Japanese Grammar) Tokyo: Taishukan.
- Oshima, M. 2010. *Nihongo Rentai Shuushokusetsu Kouzou no Kenkyuu*. (A Study on Japanese Adnominal Clause Constructions) Tokyo: Hituzi.
- Otsuka, M. 1999. *Head-Internal Relatives in Japanese from an LDSNL Perspective*. MA dissertation, University of London.
- Parsons, T. 1990. *Events in the Semantics of English*. Cambridge, MA: MIT Press.
- Partee, B., ter Meulen, A., and Wall, R. 1990. *Mathematical Methods in Linguistics*, 2nd edition. Dordrecht: Kluwer.
- Paul, I. 2001. Concealed pseudo-clefts. *Lingua* **111** (9), 707-27.
- Percus, O. 1997. Prying open the cleft. In Kusumoto, K. (ed.) *Proceedings of NELS 27*. Amherst, MA: GLSA.
- Phillips, C. 2003. Linear order and constituency. *Linguistic Inquiry* **34** (1), 37-90.
- Pickering, M. & Garrod, S. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences* **27** (2), 169-226.
- Prince, E. 1978. A comparison of *wh*-clefts and *it*-clefts in discourse. *Language* **54** (4), 883-906.
- Pritchett, B. L. 1992. *Grammatical Competence and Parsing Performance*. Chicago, IL: University of Chicago Press.
- Pullum, G. K. & Scholz, B. C. 2001. On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In de Groote, P. et al. (eds.) *Logical Aspects of Computational Linguistics (Lecture Notes in Artificial Intelligence 2099)*. Berlin: Springer-Verlag.
- Purver, M., Cann, R., and Kempson, R. 2006. Grammars as parsers. *Research on Language and Computation* **4** (2/3), 289-326.
- Pustet, R. 2003. *Copulas*. Oxford: Oxford University Press.
- Recanati, F. 2003. *Literal Meaning*. Cambridge: Cambridge University Press.
- Reichenbach, H. 1947. *Elements of Symbolic Logic*. London: Macmillan.
- Rizzi, L. 1997. The fine structure of the left periphery. In Haegeman, L. (ed.) *Elements of Grammar*. Dordrecht: Kluwer.

- Ross, J. R. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT. Published as Ross (1986).
- Ross, J. R. 1969. Guess who? In Binnick, A. et al. (eds.) *Papers from the 5th Regional Meeting of the Chicago Linguistic Society*. Chicago, IL: University of Chicago Press.
- Ross, J. R. 1986. *Infinite Syntax!* New Jersey: Ablex Publishing Corporation.
- Russell, B. 1905. On denoting. *Mind* **14** (56), 479-93.
- Sag, I. 2013. Sign-Based Construction Grammar. In Boas, H. & Sag, I. (eds.) *Sign-Based Construction Grammar*. Stanford: CSLI.
- Sag, I., Boas, H., and Kay, P. 2013. Introducing Sing-Based Construction Grammar. In Boas, H. & Sag, I. (eds.) *Sign-Based Construction Grammar*. Stanford: CSLI.
- Sag, I. & Wasow, T. 2011. Performance-compatible competence grammar. In Borsley, R. & Borjars, K. (eds.) *Non-transformational Syntax*. Oxford: Wiley-Blackwell.
- Sag, I., Wasow, T., and Bender, E. 2003. *Syntactic Theory*, 2nd edition. Stanford: CSLI.
- Saiki, M. 1985. On the coordination of gapped constituents in Japanese. In Eilfort, W. H. et al. (eds.) *Papers from the Twenty-first Regional Meeting of the Chicago Linguistics Society*. Chicago, IL: University of Chicago Press.
- Saito, M. 2004. Ellipsis and pronominal reference in Japanese clefts. *Nanzan Linguistics* **1**, 21-50.
- Saito, M., Lin, J. T.-H., and Murasugi, K. 2008. N'-ellipsis and the structure of noun phrases in Chinese and Japanese. *Journal of East Asian Linguistics* **17** (3), 247-71.
- Saito, M. & Murasugi, K. 1990. N'-deletion in Japanese. In Hoji, H. (ed.) *Japanese/Korean Linguistics*. Stanford: CSLI.
- Sakai, H. 1994. Complex NP constraint and case-conversions in Japanese. In Nakamura, M. (ed.) *Current Topics in English and Japanese*. Tokyo: Hituzi.
- Sakai, H. 2000. Predicate ellipsis and nominalization in Japanese. *Proceedings of 2000 Seoul International Conference on Language and Computation*. Seoul: Korea University.
- Sato, Y. 2011. Local ambiguity, search strategies and parsing in Dynamic Syntax. In Kempson, et al. (eds.) *The Dynamics of Lexical Interfaces*. Stanford: CSLI.
- Saussure, F. de. 1916. *Cours de Linguistique Générale*. Paris: Payo.
- Sells, P. 1986. Coreference and bound anaphora. In S. Berman, J.-W. & McDonough, J. (eds.) *Proceedings of NELS 16*. Amherst, MA: GLSA.
- Seraku, T. 2008. *The Explicit/Implicit Distinction*. MA dissertation, University College London.

- Seraku, T. 2011. Clefts and free-relatives in Japanese. In Bailey, D. & Teliga, V. (eds.) *Proceedings of the Thirty-Ninth Western Conference on Linguistics*. CA: Department of Linguistics, California State University, Fresno.
- Seraku, T. 2012a. Multiple foci in Japanese clefts and the growth of semantic representation. In Aloni, M. et al. (eds.) *Logic, Language, and Meaning (Lecture Notes in Computer Science 7218)*. Berlin: Springer-Verlag.
- Seraku, T. 2012b. Two types of nominalization in Japanese as an outcome of semantic tree growth. In Manurung, R. & Bond, F. (eds.) *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*. Faculty of Computer Science, Universitas Indonesia.
- Seraku, T. 2013a. Towards an LFG account of the compound particle *toiuno* in Japanese. The 11th South of England LFG Meeting. SOAS (University of London), UK.
- Seraku, T. 2013b. Multiple foci in Japanese clefts revisited. *Lingua* **137**, 145-71.
- Seraku, T. in press a. Structural underspecification and resolution within a processing-oriented grammar formalism. *International Journal of Computational Linguistics and Applications*.
- Seraku, T. in press b. Head-internal relatives in Japanese as rich context-setters. *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation*.
- Seraku, T. in press c. An incremental semantics account of the particle *no* in Japanese. *Proceedings of the Forties Western Conference on Linguistics*. CA: Department of Linguistics, California State University, Fresno.
- Shibatani, M. 1990. *The Languages of Japan*. Cambridge: Cambridge University Press.
- Shibatani, M. 2009. Elements of complex structures, where recursion isn't. In Givon, T. & Shibatani, M. (eds.) *Syntactic Complexity*. Amsterdam: John Benjamins.
- Shibatani, M. 2013. What can Japanese dialects tell us about the function and development of the nominalization particle *no*? In Frellesvig, B. & Sells, P. (eds.) *Japanese/Korean Linguistics* **20**. Stanford: CSLI.
- Shibatani, M. & Bin Makhshen, K. A. 2009. Nominalization in Soqotri, a South Arabian language of Yemen. In Wetzels, L. (ed.) *The Linguistics of Endangered Languages*. Leiden: Brill.
- Shimoyama, J. 1999. Internally headed relative clauses in Japanese and E-type anaphora. *Journal of East Asian Linguistics* **8** (2), 147-82.
- Shimoyama, J. 2001. *Wh-Constructions in Japanese*. Ph.D. thesis, University of Massachusetts, Amherst.

- Simpson, A. & Wu, X.-Z. 2001. The grammaticalization of formal nouns and nominalizers in Chinese, Japanese and Korean. In McAuley, T. E. (ed.) *Language Change in East Asia*. Surrey: Curzon Press.
- Soga, M. & Fujimura, T. 1978. A note on noun modification in Japanese. *Journal of the Association of Teachers of Japanese* **13** (1), 41-51.
- Sperber, D. & Wilson, D. 1995. *Relevance*, 2nd edition. Oxford: Blackwell.
- Sunakawa, Y. 1995. Nihongo niokeru bunretsubun no kinou to gojun no genri. (Function of Japanese cleft sentences and the principle of word order) In Nitta, Y. (ed.) *Fukubun no Kenkyuu*. Tokyo: Kuroshio.
- Sunakawa, Y. 2005. *Bunpou to Danwa no Setten*. (The Interface between Grammar and Discourse) Tokyo: Kuroshio.
- Takahashi, D. 1994. Sluicing in Japanese. *Journal of East Asian Linguistics* **3** (2), 265-300.
- Takano, Y. 2002. Surprising constituents. *Journal of East Asian Linguistics* **11** (3), 243-301.
- Tamura, S. 2009. *Gendai Nihongo niokeru Meishibun no Jikanhyougen*. (Time Expressions of Noun-Predicate Sentences in Modern Japanese) Hiroshima: Keisuisha.
- Tomasello, M. 2003. *Constructing a Language*. Cambridge, MA: Harvard University Press.
- Tonoike, S. 1990. *No no ronrikeishiki*. (LF representation of *no*) *Meijigakuin Ronsou* **467**, 69-99.
- Tonosaki, S. 1996. Change of state head-internal relative clauses in Japanese. *Studies in Language Sciences* **2**, 31-47.
- Tonosaki, S. 1998. Change-relatives in Japanese. *Journal of Japanese Linguistics* **16**, 143-60.
- Tsubomoto, A. 1995. Bunrenketsu to ninchi zushiki. (Sentence linkage and cognitive diagram) *Nihongogaku* **14** (3), 79-91.
- Tsujimura, N. 2007. *An Introduction to Japanese Linguistics*, 2nd edition. Oxford: Blackwell.
- Watanabe, A. 1992. Subjacency and s-structure movement of *wh*-in-situ. *Journal of East Asian Linguistics* **1** (3), 255-91.
- Watanabe, A. 2003. Wh and operator constructions in Japanese. *Lingua* **113** (4), 519-58.
- Watanabe, Y. 2008. *Bunhogohyoushiki 'Koto' 'No' no Imitekisoui nikansuru Kenkyuu*. (Study of Semantic Differences between Complementizer 'Koto' and 'No.') Hiroshima: Keisuisha.
- Wedgwood, D. 2005. *Shifting the Focus*. Oxford: Elsevier.

- Wei, L. & Kempson, R. 2011. Chinese cleft structures and the dynamics of processing. Ms., Beijing Jiaotong University, King's College London.
- Williams, E. 1980. Predication. *Linguistic Inquiry* **11** (1), 203-38.
- Wilson, D. & Sperber, D. 2012. *Meaning and Relevance*. Cambridge: Cambridge University Press.
- Wrona, J. 2005. Specificational pseudoclefts in old Japanese. *Folia Linguistica Historica* **26** (1/2), 139-57.
- Wrona, J. 2012. The early history of *no* as a nominalizer. In Frellesvig, B. et al. (eds.) *Studies in Japanese and Korean Linguistics*. London: Lincom.
- Yamada, M. 2004. Shouten idou toshitenno kakimazebun to bunretsubun. (Scrambling and clefting as focus movement) In Sato, S. et al. (eds.) *Taishou Gengogaku no Shintenkai*. Tokyo: Hituzi.
- Yanagida, S. 1993. *No no tenkai*, Kodai-Nihongo-kara Kindaigo-e no. (Development of *no* from Old Japanese to Modern Japanese) *Nihongogaku* **12** (11), 15-22.
- Yap, F. H., Grunow-Harsta, K., and Wrona, J. (eds.) 2011. *Nominalization in Asian Languages*. Amsterdam: John Benjamins.
- Yap, F. H., Grunow-Harsta, K., and Wrona, J. 2011. Nominalization strategies in Asian languages. In Yap, F. H. et al. (eds.) *Nominalization in Asian Languages*. Amsterdam: John Benjamins.
- Yap, F. H. & Matthews, S. 2008. The development of nominalizers in East Asian and Tibeto-Burman languages. In López-Couso, M. & Seoane, E. (eds.) *Rethinking Grammaticalization*. Amsterdam: John Benjamins.
- Yap, F. H., Matthews, S., and Horie, K. 2004. From pronominalizer to pragmatic marker. In Fischer, O. et al. (eds.) *Up and Down the Cline*. Amsterdam: John Benjamins.
- Yoshimura, N. 2001. Bunretsubun-o Yatsushirohougen-kara saguru. (A dialectal study of Japanese cleft constructions) *Kotoba to Bunka* **4**, 67-84.
- Yoshimura, N. 2005. Gengo no henka to kakutoku. (Change and acquisition of language) *Kotoba to Bunka* **8**, 55-73.
- Yoshimura, N. 2010. Dialectal perspectives on the emergence of Japanese complementizer *no*. In Iwasaki, S. (ed.) *Japanese/Korean Linguistics* **17**. Stanford: CSLI.
- Yoshimura, N. & Nishina, A. 2004. Bunretsubun no imi to kouzou. (Meaning and structure of cleft sentences) *Kotoba to Bunka* **7**, 55-72.