Available online at www.sciencedirect.com
ScienceDirect
Information & Computation 00 (2018) 1–36
Information & Computation

Game Semantics for Dependent Types

Matthijs Vákár[a,*], Radha Jagadeesan[b], Samson Abramsky[a]

[a] *Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK*
[b] *School of CTI, DePaul University, 243 S. Wabash Avenue Chicago, Illinois 60604-2287, USA*

Abstract

We present a model of dependent type theory (DTT) with Π-, 1-, Σ- and intensional Id-types, which is based on a slight variation of the (call-by-name) category of AJM-games and history-free winning well-bracketed strategies. The model satisfies Streicher's criteria of intensionality and refutes function extensionality. The principle of uniqueness of identity proofs is satisfied.

We show it contains a submodel as a full subcategory which gives a faithful interpretation of DTT with Π-, 1-, Σ- and intensional Id-types and, additionally, finite inductive type families. This smaller model is fully (and faithfully) complete with respect to the syntax at the type hierarchy built without Id-types, as well as at the more general class of types where we allow for one strictly positive occurrence of an Id-type. Definability for the full type hierarchy with Id-types remains to be investigated.

Keywords: game semantics, dependent type theory, intensionality

1. Introduction

Dependent Type theory (DTT) can be seen as the extension of the simple λ-calculus along the Curry-Howard correspondence from a proof calculus for (intuitionistic) propositional logic to one for predicate logic. It forms the basis of many proof assistants, like NuPRL, LEGO and Coq, and is increasingly being considered as a more expressive type system for programming, as implemented in e.g. ATS, Cayenne, Epigram, Agda and Idris [1] and with even Haskell slowly approaching its expressive power with the addition of GADTs [2]. A recent source of enthusiasm in this field is homotopy type theory (HoTT), which refers to an interpretation of DTT into abstract homotopy theory [3] or, conversely, an extension of DTT that is sufficient to reproduce significant results of homotopy theory [4]. In practice, the latter means DTT with Σ-, Π-, Id-types (corresponding to existential and universal quantifiers and identity predicates, respectively, through the Curry-Howard correspondence), a universe (roughly, a type of types) satisfying the **univalence axiom**, and certain higher inductive types (playing the role of ground types whose towers of iterated identity types behave like the homotopy types of certain spaces). The univalence axiom is an extensionality principle which implies, in particular, the axiom of function extensionality [4].

Game semantics provides a unified framework for intensional, computational semantics of various type theories, ranging from pure logics [5] to programming languages [6, 7, 8, 9] with a variety of features (e.g. non-local control [10], state [11, 12, 13], non-determinism [14], probability [15], dynamically generated local names [16]) and evaluation strategies [17]. A game semantics for DTT has, surprisingly, so far been absent. Our hope is that such a semantics will provide an alternative analysis of the implications of the subtle shades of intensionality that arise in the analysis

*Corresponding author
Email addresses: matthijs.vakar@cs.ox.ac.uk (Matthijs Vákár), rjagadeesan@cs.depaul.edu (Radha Jagadeesan), samson.abramsky@cs.ox.ac.uk (Samson Abramsky)

of DTT [18, 19]. Moreover, the game semantics of DTT is based on very different, one might say orthogonal intuitions to those of the homotopical models: temporal rather than spatial, and directly reflecting the structure of computational processes. One goal, to which we hope this work will be a stepping stone, is a game semantics of HoTT doing justice to both the spatial and temporal aspects of identity types. Indeed, such an investigation might even lead to a computational interpretation of the univalence axiom which has long been missing, although a significant step in this direction was recently taken by the constructive cubical sets model of HoTT [20].

Our game theoretic model of DTT is inspired in part by the domain model of DTT [21]. This model views a type family as a continuous function to a domain of domains, a witness of a Π-type $\Pi_{x:A}B$ as a continuous (set theoretic) dependent function and interprets identity types via a kind of intersection. We follow this recipe for modelling type families and identity types. We adapt the viewpoint of the game semantics of system F of [9] to describe the Π-type to capture the intuitive idea that the specialisation of a term at type $\Pi_{x:A}B$ to a specific instance $B[a/x]$ is the responsibility solely of the context that provides the argument a of type A; in contrast, any valid term of $\Pi_{x:A}B$ has to operate within the constraints enforced by the context. Our definition draws its power from the fact that in a game semantics, these constraints are enforced not only on completed computations, but also on the incomplete computations that arise when a term interacts with its context. Thus, while we follow some of the formal recipes of [21], the temporal character of game semantics results in strikingly different properties of the resulting model.

In this paper, we describe a game theoretic model of DTT with 1-, Σ-, Π- and intensional Id-types, where (lists of dependent) (call-by-name) AJM-games interpret types and (lists of) history-free winning well-bracketed strategies on games of dependent functions interpret terms. We next specialize to the semantic type hierarchy formed by the 1-, Σ-, Π-, and Id-constructions and substitution over a set of finite dependent games. We show that this gives a model of DTT which additionally supports finite inductive type families. Our two models have the following key properties.

- The place of the Id-types in the intensionality spectrum (in either model) compares as follows with the domain semantics with totality and with HoTT.

	Domains	HoTT	Games
Failure of Equality Reflection	✓	✓	✓
Streicher [18] Intensionality Criteria ($I1$) and ($I2$)	✓	✓	✓
Streicher [18] Intensionality Criterion ($I3$)	✗	✗	✓
Failure of Function Extensionality (FunExt)	✗	✗	✓
Failure of Uniqueness of Identity Proofs (UIP)	✗	✓	✗

- We show that the smaller model faithfully interprets a version of DTT with 1-, Σ-, Π- and Id-types and finite inductive type families. Moreover, it is fully complete at the types which do not involve Id in their construction or which involve one strictly positive Id-type as a subformula. Full completeness for the full type hierarchy remains to be investigated. In contrast, the domain theoretic model of [21] is not (fully) complete or faithful.

Related Work

We are not aware of any published account of a game semantics for dependent type theory. Closest in spirit to the current work, however, are the domain model of dependent type theory [21] and the version of game semantics for system F of [9]. Our model may be contrasted with the homotopy interpretations of dependent type theory [22, 3, 4]. Rather, it seems to bear more resemblance to the (modified) realizability models of [18].

Outline

In section 2, we recall the syntax of both a simple total type theory STT over finite ground types and of a corresponding dependent type theory DTT over finite inductive type families which serve as ground types. A translation $(-)^T$ lets us transfer the equational theory of STT to DTT, which spares us from writing down all appropriate dependently typed commutative conversions. Next, in section 3, we recall the basic setting of AJM-style game semantics for STT and its completeness results. In section 4, we introduce a notion of dependent game and dependently typed strategy, together with a semantic equivalent $☺(-)$ of $(-)^T$: a translation to simply typed game semantics. Treating Σ-types formally, we construct an interpretation of DTT in sections 5, 6 and 7, in the form of a category with families with Σ-, Π- and Id-types and finite inductive type families. Section 6 further characterises various intensionality

properties of the Id-types. Soundness and faithfulness of the interpretation of DTT are finally proved in section 8, as the interpretation factors faithfully over the faithful sound games interpretation of STT, as well as full completeness results which are obtained by a dependently typed modification of the definability proofs of [8, 23]. We end on a discussion of future work in section 9.

2. Type Theoretic Preliminaries

2.1. Dependently Typed Equational Logic

In this section, we briefly recall the framework of dependently typed equational logic (sometimes called generalised algebraic theories [24]), which will serve as the structural core type theory, on top of which we consider two theories: a flavour of simple type theory (STT) and a flavour of dependent type theory (DTT). This framework puts both flavours of type theory on an equal footing and allows us to better study their relationship. We go into this level of precision in our specification of the syntax we are modelling, in order to accurately state the appropriate completeness results in section 8. Although much more informal, our treatment is close in spirit to those of [25] and [19], to which we refer the interested reader for more background and where the reader can find details on delicate topics like pre-syntax, α-conversion, variable binding and capture-avoiding substitution.

Judgements

The language of dependently typed equational logic expresses **judgements** of the following six forms. We present the various kinds of judgements in our formal language and their intended meaning.

Judgement	**Intended meaning**
$\vdash \Gamma$ ctxt	Γ is a valid context
$\Gamma \vdash A$ type	A is a type in context Γ
$\Gamma \vdash a : A$	a is a term of type A in context Γ
$\vdash \Gamma \equiv \Gamma'$ ctxt	Γ and Γ' are judgementally equal contexts
$\Gamma \vdash A \equiv A'$ type	A and A' are judgementally equal types in context Γ
$\Gamma \vdash a \equiv a' : A$	a and a' are judgementally equal terms of type A in context Γ

Figure 1. Judgements of DTT.

Here, $\Gamma, \Gamma', A, , A', a$ and a' are all symbolic expressions from a set Expr, built from an alphabet Sym, in which we have countably infinite designated subsets Var of variables and Cons of constants. As usual, we distinguish between the free and bound variables occurring in an expression $\mathcal{J}$ and we consider expressions $\mathcal{J}$ up to α-equivalence, or up to permutations of Var fixing the free variables of $\mathcal{J}$. We denote the syntactic metaoperation of capture-avoiding substitution of an expression a for all occurrences of a free variable x in an expression $\mathcal{J}$ by $\mathcal{J}[a/x]$.

Structural Rules and Theories

Dependently typed equational logic has the following structural rules, which will be shared by STT and DTT.

$$\frac{\Gamma, \Gamma' \vdash \mathcal{J} \qquad \vdash \Gamma, x : A, \Gamma' \text{ ctxt}}{\Gamma, x : A, \Gamma' \vdash \mathcal{J}} \text{ Weak} \qquad \frac{\Gamma, x : A, \Gamma' \vdash \mathcal{J} \qquad \Gamma \vdash a : A}{\Gamma, \Gamma'[a/x] \vdash \mathcal{J}[a/x]} \text{ Subst}$$

Figure 2. Weakening and substitution rules. Here, $\mathcal{J}$ represents a statement of the form B type, $B \equiv B'$, $b : B$, or $b \equiv b' : B$.

$$\frac{}{\vdash \cdot \text{ ctxt}} \text{ C-Emp} \qquad \frac{\vdash \Gamma \text{ ctxt} \qquad \Gamma \vdash A \text{ type} \qquad x \text{ is fresh for } \Gamma \text{ and } A}{\vdash \Gamma, x : A \text{ ctxt}} \text{ C-Ext} \qquad \frac{\vdash \Gamma, x : A, \Gamma' \text{ ctxt}}{\Gamma, x : A, \Gamma' \vdash x : A} \text{ Var}$$

$$\frac{\Gamma \equiv \Gamma' \text{ ctxt} \qquad \Gamma \vdash A \equiv B \text{ type} \qquad \vdash \Gamma, x : A \text{ ctxt} \qquad \vdash \Gamma', x : B \text{ ctxt}}{\vdash \Gamma, x : A \equiv \Gamma', x : B \text{ ctxt}} \text{ C-Ext-Eq}$$

Figure 3. Context formation and variable declaration rules.

$$\frac{\vdash \Gamma \ \mathsf{ctxt}}{\vdash \Gamma \equiv \Gamma \ \mathsf{ctxt}}\ \text{C-Eq-R} \qquad \frac{\vdash \Gamma \equiv \Gamma' \ \mathsf{ctxt}}{\vdash \Gamma' \equiv \Gamma \ \mathsf{ctxt}}\ \text{C-Eq-S} \qquad \frac{\vdash \Gamma \equiv \Gamma' \ \mathsf{ctxt} \qquad \vdash \Gamma' \equiv \Gamma'' \ \mathsf{ctxt}}{\vdash \Gamma \equiv \Gamma'' \ \mathsf{ctxt}}\ \text{C-Eq-T}$$

$$\frac{\Gamma \vdash A \ \mathsf{type}}{\Gamma \vdash A \equiv A \ \mathsf{type}}\ \text{Ty-Eq-R} \qquad \frac{\Gamma \vdash A \equiv A' \ \mathsf{type}}{\Gamma \vdash A' \equiv A \ \mathsf{type}}\ \text{Ty-Eq-S} \qquad \frac{\Gamma \vdash A \equiv A' \ \mathsf{type} \qquad \Gamma \vdash A' \equiv A'' \ \mathsf{type}}{\Gamma \vdash A \equiv A'' \ \mathsf{type}}\ \text{Ty-Eq-T}$$

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a \equiv a : A}\ \text{Tm-Eq-R} \qquad \frac{\Gamma \vdash a \equiv a' : A}{\Gamma \vdash a' \equiv a : A}\ \text{Tm-Eq-S} \qquad \frac{\Gamma \vdash a \equiv a' : A \qquad \Gamma \vdash a' \equiv a'' : A}{\Gamma \vdash a \equiv a'' : A}\ \text{Tm-Eq-T}$$

$$\frac{\Gamma \vdash A \ \mathsf{type} \qquad \vdash \Gamma \equiv \Gamma' \ \mathsf{ctxt}}{\Gamma' \vdash A \ \mathsf{type}}\ \text{Ty-Conv} \qquad \frac{\Gamma \vdash a : A \qquad \vdash \Gamma \equiv \Gamma \ \mathsf{ctxt} \qquad \Gamma; \cdot \vdash A \equiv A' \ \mathsf{type}}{\Gamma' \vdash a : A'}\ \text{Tm-Conv}$$

Figure 4. Rules for judgemental equality, making it an equivalence relation, compatible with typing.

We can use our framework to talk about various type theories. By a **theory**, we mean a set $\mathbb{T}$ of judgements which is closed under the structural rules above, in the sense that if their hypotheses (written above the horizontal line) are in $\mathbb{T}$, then so are their conclusions (written under the line). Usually, we specify a theory by a set of **axioms**, a set of judgements which can be inductively closed under the structural rules to obtain a theory.

2.2. Simple Type Theory (STT)

The simple type theory we use is a variant STT of the simply typed λ-calculus with finite product types and finite inductive types $\{a_i \mid i\}$ for any finite set of **distinct** constants $a_1, \ldots, a_n$, with β- and η-rules and the appropriate commutative conversions for the corresponding case-constructs - essentially the PCF commutative conversions. We are considering a total finitary PCF, if you will. Specifically, with STT, we are referring to the theory in dependently typed equational logic generated by the following rules together with the obvious congruence rules which state that all (type and) term formers respect judgemental equality.

$$\frac{}{\vdash 1 \ \mathsf{type}}\ 1\text{-F} \qquad \frac{\vdash \Gamma \ \mathsf{ctxt}}{\Gamma \vdash \langle\rangle : 1}\ 1\text{-I} \qquad \frac{\Gamma \vdash t : 1}{\Gamma \vdash t \equiv \langle\rangle : 1}\ 1\text{-}\eta$$

$$\frac{\Gamma \vdash B \ \mathsf{type} \qquad \Gamma \vdash C \ \mathsf{type}}{\Gamma \vdash B \times C \ \mathsf{type}}\ \times\text{-F} \qquad \frac{\Gamma \vdash b : B \qquad \Gamma \vdash c : C}{\Gamma \vdash \langle b, c\rangle : B \times C}\ \times\text{-I} \qquad \frac{\Gamma \vdash d : B \times C}{\Gamma \vdash \mathsf{fst}(d) : B}\ \times\text{-E1} \qquad \frac{\Gamma \vdash d : B \times C}{\Gamma \vdash \mathsf{snd}(d) : C}\ \times\text{-E2}$$

$$\frac{\Gamma \vdash \mathsf{fst}(\langle b, c\rangle) : B}{\Gamma \vdash \mathsf{fst}(\langle b, c\rangle) \equiv b : B}\ \times\text{-}\beta 1 \qquad \frac{\Gamma \vdash \mathsf{snd}(\langle b, c\rangle) : C}{\Gamma \vdash \mathsf{snd}(\langle b, c\rangle) \equiv c : C}\ \times\text{-}\beta 2 \qquad \frac{\Gamma \vdash \langle \mathsf{fst}(d), \mathsf{snd}(d)\rangle : B \times C}{\Gamma \vdash \langle \mathsf{fst}(d), \mathsf{snd}(d)\rangle \equiv d : B \times C}\ \times\text{-}\eta$$

$$\frac{\Gamma \vdash B \ \mathsf{type} \qquad \Gamma \vdash C \ \mathsf{type}}{\Gamma \vdash B \Rightarrow C \ \mathsf{type}}\ \Rightarrow\text{-F} \qquad \frac{\Gamma, x : B \vdash c : C}{\Gamma \vdash \lambda_{x:B} c : B \Rightarrow C}\ \Rightarrow\text{-I} \qquad \frac{\Gamma \vdash f : B \Rightarrow C \qquad \Gamma \vdash b : B}{\Gamma \vdash f(b) : C}\ \Rightarrow\text{-E}$$

$$\frac{\Gamma \vdash (\lambda_{x:B} c)(b) : C}{\Gamma \vdash (\lambda_{x:B} c)(b) \equiv c[b/x] : C}\ \Rightarrow\text{-}\beta \qquad \frac{\Gamma \vdash \lambda_{x:B} f(x) : B \Rightarrow C}{\Gamma \vdash \lambda_{x:B} f(x) \equiv f : B \Rightarrow C}\ \Rightarrow\text{-}\eta$$

$$\frac{a_i, \quad 1 \le i \le n, \quad \text{distinct constants}}{\vdash \{a_i \mid i\} \ \mathsf{type}}\ \{a_i \mid i\}\text{-F} \qquad \frac{}{\vdash a_j : \{a_i \mid i\}}\ \{a_i \mid i\}\text{-I}_j \qquad \frac{\vdash C \ \mathsf{type}}{x : \{a_i \mid i\}, z_1 : C, \ldots, z_n : C \vdash \mathsf{case}_{\{a_i \mid i\},C}(x, \{z_i\}_i) : C}\ \{a_i \mid i\}\text{-E}$$

$$\frac{z_1 : C, \ldots, z_n : C \vdash \mathsf{case}_{\{a_i \mid i\},C}(a_j, \{z_i\}_i) : C}{z_1 : C, \ldots, z_n : C \vdash \mathsf{case}_{\{a_i \mid i\},C}(a_j, \{z_i\}_i) \equiv z_j : C}\ \{a_i \mid i\}\text{-}\beta_j \qquad \frac{x : \{a_i \mid i\} \vdash \mathsf{case}_{\{a_i \mid i\},\{a_i \mid i\}}(x, \{a_i\}_i) : \{a_i \mid i\}}{x : \{a_i \mid i\} \vdash \mathsf{case}_{\{a_i \mid i\},\{a_i \mid i\}}(x, \{a_i\}_i) \equiv x : \{a_i \mid i\}}\ \{a_i \mid i\}\text{-}\eta$$

$$\frac{\Gamma \vdash \mathsf{case}_{\{a_i \mid i\},B\times C}(x, \{d_i\}_i) : B \times C}{\Gamma \vdash \mathsf{case}_{\{a_i \mid i\},B\times C}(x, \{d_i\}_i) \equiv \langle \mathsf{case}_{\{a_i \mid i\},B}(x, \{\mathsf{fst}(d_i)\}_i), \mathsf{case}_{\{a_i \mid i\},C}(x, \{\mathsf{snd}(d_i)\}_i)\rangle : B \times C}\ \{a_i \mid i\}\text{-Comm-}\langle -, -\rangle$$

$$\frac{\Gamma \vdash \mathsf{case}_{\{a_i \mid i\},B\Rightarrow C}(x, \{f_i\}_i) : B \Rightarrow C}{\Gamma \vdash \mathsf{case}_{\{a_i \mid i\},B\Rightarrow C}(x, \{f_i\}_i) \equiv \lambda_{y:B} \mathsf{case}_{\{a_i \mid i\},C}(x, \{f_i(y)\}_i) : B \Rightarrow C}\ \{a_i \mid i\}\text{-Comm-}\lambda$$

$$\frac{\Gamma \vdash \mathsf{case}_{\{b_j \mid j\},C}(\mathsf{case}_{\{a_i \mid i\},\{b_j \mid j\}}(x, \{b'_i\}), \{c_j\}_j) : C}{\Gamma \vdash \mathsf{case}_{\{b_j \mid j\},C}(\mathsf{case}_{\{a_i \mid i\},\{b_j \mid j\}}(x, \{b'_i\}), \{c_j\}_j) \equiv \mathsf{case}_{\{a_i \mid i\},\{b_j \mid j\}}(x, \{\mathsf{case}_{\{b_j \mid j\},C}(b'_i, c_j)\}_j) : C}\ \{a_i \mid i\}\text{-Comm-case}$$

Figure 5. The rules generating the axioms for STT. For $\Rightarrow -\eta$, we demand the usual side condition that x not free in f.

2.3. Dependent Type Theory (DTT−)

Similarly, we can present our preferred variant DTT of dependent type theory as a theory in dependently typed equational logic. First, we present a smaller theory DTT−, which does not yet include the β- and η-rules and commutative conversions of DTT, but rather only consists of its F-, I- and E-rules. Later, DTT is obtained by adding to DTT− the equational theory that is obtained from that of STT, under a syntactic translation to STT.

Firstly, we have Σ-, Π- and Id-types.

$$\frac{}{\vdash 1\ \mathsf{type}}\ 1\text{-F} \qquad \frac{\vdash \Gamma\ \mathsf{ctxt}}{\Gamma \vdash \langle\rangle : 1}\ 1\text{-I}$$

$$\frac{\Gamma, x : A \vdash B\ \mathsf{type}}{\Gamma \vdash \Sigma_{x:A} B\ \mathsf{type}}\ \Sigma\text{-F} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B[a/x]}{\Gamma \vdash \langle a, b\rangle : \Sigma_{x:A} B}\ \Sigma\text{-I} \qquad \frac{\Gamma \vdash t : \Sigma_{x:A} B}{\Gamma \vdash \mathsf{fst}(t) : A}\ \Sigma\text{-E1} \qquad \frac{\Gamma \vdash t : \Sigma_{x:A} B}{\Gamma \vdash \mathsf{snd}(t) : B[\mathsf{fst}(t)/x]}\ \Sigma\text{-E2}$$

$$\frac{\Gamma, x : A \vdash B\ \mathsf{type}}{\Gamma \vdash \Pi_{x:A} B\ \mathsf{type}}\ \Pi\text{-F} \qquad \frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \lambda_{x:A} b : \Pi_{x:A} B}\ \Pi\text{-I} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash f : \Pi_{x:A} B}{\Gamma \vdash f(a) : B[a/x]}\ \Pi\text{-E}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash a' : A}{\Gamma \vdash \mathsf{Id}_A(a, a')\ \mathsf{type}}\ \mathsf{Id}\text{-F} \qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{refl}_a : \mathsf{Id}_A(a, a)}\ \mathsf{Id}\text{-I}$$

$$\frac{\begin{array}{l}\Gamma \vdash a : A \\ \Gamma \vdash a' : A \qquad \Gamma, x : A, x' : A, y : \mathsf{Id}_A(x, x') \vdash D\ \mathsf{type} \\ \Gamma \vdash p : \mathsf{Id}_A(a, a') \qquad \Gamma, z : A \vdash d : D[z/x, z/x', \mathsf{refl}_z/y]\end{array}}{\Gamma \vdash \mathsf{let}\ p\ \mathsf{be}\ \mathsf{refl}_z\ \mathsf{in}\ d : D[a/x, a'/x', p/y]}\ \mathsf{Id}\text{-E}$$

Figure 6. Rules for 1-, Σ-, Π-, and Id-types. In case x is not free in B, we sometimes write $A \Rightarrow B$ for $\Pi_{x:A} B$ and $A \times B$ for $\Sigma_{x:A} B$.

Secondly, we have a mechanism for forming finite inductive type families, which play the rôle of ground types. Let A be a type formed without using Π-type constructors. Then, we specify a finite inductive definition of a type family $x : A \vdash (a_i \mapsto_i \{b_{i,j} \mid j\})(x)\ \mathsf{type}$ by specifying finitely many closed terms $a_1, \ldots, a_n : A$ and distinct symbols $b_{i,j}$, $1 \le i \le n$, $1 \le j \le m_i$. The idea is that $B = (a_i \mapsto_i \{b_{i,j} \mid j\})(x)$ is a type family, such that $(a_i \mapsto_i \{b_{i,j} \mid j\})(a_i)$ contains precisely the distinct closed terms $b_{i,1}, \ldots, b_{i,m_i}$. These type families are more limited than general inductive definitions as they are freely generated by **closed** terms, while one would allow open terms in the general case [26]. This means that we precisely get the inductive type families with finitely many non-empty fibres which are all finite types. An example the reader may want to keep in mind is given by calendars in format dd-mm (for 1984, for instance): here $A = \mathsf{mm} := \{01, \ldots, 12\}$ and $B = \mathsf{dd} - \mathsf{mm} := (i \mapsto_i \{01\text{-}i, \ldots, N_i\text{-}i\})(x)$, where N_i is 29, 30, or 31, depending on the number of days the month in question has.

$$\frac{\begin{array}{c}\vdash a_1 : A \qquad \ldots \qquad \vdash a_n : A \\ b_{i,j},\quad 1 \le i \le n,\ 1 \le j \le m_i,\ \text{distinct constants}\end{array}}{x : A \vdash (a_i \mapsto_i \{b_{i,j} \mid j\})(x)\ \mathsf{type}}\ (a_i \mapsto_i \{b_{i,j} \mid j\})(x)\text{-F} \qquad \frac{}{\vdash b_{i,j} : (a_i \mapsto_i \{b_{i,j} \mid j\})(a_i)}\ (a_i \mapsto_i \{b_{i,j} \mid j\})(x)\text{-}I_{i,j}$$

$$\frac{x : A, y : (a_i \mapsto_i \{b_{i,j} \mid j\})(x) \vdash C\ \mathsf{type}}{x : A, y : (a_i \mapsto_i \{b_{i,j} \mid j\})(x), z_{1,1} : C[a_1/x, b_{1,1}/y], \ldots, z_{n,m_n} : C[a_n/x, b_{n,m_n}/y] \vdash \mathsf{case}_{(a_i \mapsto_i \{b_{i,j} \mid j\})(x), C}(y, \{z_{i,j}\}_{i,j}) : C}\ (a_i \mapsto_i \{b_{i,j} \mid j\})(x)\text{-}E$$

$$\frac{\begin{array}{ll}\Gamma \vdash a : A & \Gamma, y : (a_i \mapsto_i \{b_{i,j} \mid j\})(a) \vdash C\ \mathsf{type} \\ \Gamma \vdash b : (a_i \mapsto_i \{b_{i,j} \mid j\})(a) & \{\Gamma, p_{i,j} : \mathsf{Id}_A(a_i, a), q_{i,j} : \mathsf{Id}_{(a_i \mapsto_i \{b_{i,j} \mid j\})(a)}(\mathsf{subst}(p, b_{i,j}), b) \vdash c_{ij} : C[b/y]\}_{i,j}\end{array}}{\Gamma \vdash \mathsf{case}^{p,q}_{(a_i \mapsto_i \{b_{i,j} \mid j\})(a), C}(b, \{c_{i,j}\}_{i,j}) : C[b/y]}\ (a_i \mapsto_i \{b_{i,j} \mid j\})(x)\text{-}E'$$

$$\frac{\vdash \mathsf{Id}_C(c, c')\ \mathsf{type} \qquad \text{where} \qquad (i, j) \neq (i', j')}{p : \mathsf{Id}_A(a_i, a_{i'}), q : \mathsf{Id}_{(a_i \mapsto_i \{b_{i,j} \mid j\})(a_{i'})}(\mathsf{subst}(p, b_{i,j}), b_{i',j'}) \vdash \mathsf{exfalso} : \mathsf{Id}_C(c, c')}\ (a_i \mapsto_i \{b_{i,j} \mid j\})(x)\text{-Discr}$$

Figure 7. Rules for a finite inductive type family $x : A \vdash (a_i \mapsto_i \{b_{i,j} \mid j\})(x)\ \mathsf{type}$, generated by $\vdash b_{i,1}, \ldots, b_{i,m_i} : (a_i \mapsto_i \{b_{i,j} \mid j\})(x)[a_i/x]$ for $\vdash a_1, \ldots, a_n : A$.

We interpret such a definition as specifying F-, I- and E-rules for $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$, as well as an ex falso quodlibet eliminator for identity types of distinct constructors[1]. In fact, instead of $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$-$E$, we may

[1] Note that this rule is derivable in presence of a universe $\mathcal{U}$. Indeed, then, using the $\mathsf{case}_{(a_i \mapsto_i \{b_{i,j} \mid j\})(x), \mathcal{U}}$-construct for $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$,

equivalently specify an alternative elimination rule $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$-$E'$. While the former is the usual elimination rule for finite inductive type families, the latter is closer, in a sense, to the intuition of our model and arises naturally in the completeness proofs later in this paper. Here, we write subst for the following principle of indiscernability of identicals.

$$\frac{\Gamma, x : A \vdash B \ \mathsf{type} \qquad \Gamma, x, x' : A, p : \mathsf{Id}_A(x, x') \vdash x, x' : A \qquad \Gamma, x, x' : A, p : \mathsf{Id}_A(x, x') \vdash p : \mathsf{Id}_A(x, x')}{\Gamma, x : A \vdash \lambda_{y:B} y : \Pi_{y:B} B \qquad \Gamma, x, x' : A, p : \mathsf{Id}_A(x, x') \vdash \mathsf{subst}(p, -) : \Pi_B B[x'/x]}\ \mathsf{Id}\text{-}E$$

More generally, for a context $\Gamma, x : A, y_1 : B_1, \dots, y_n : B_n$, we can inductively define

$$\Gamma, x, x' : A, p : \mathsf{Id}_A(x, x'), y_1 : B_1, \dots, y_{n-1} : B_{n-1} \vdash \mathsf{subst}(p, -) : \Pi_{y_n:B_n} B_n[x'/x, \mathsf{subst}(p, y_1)/y_1, \dots, \mathsf{subst}(p, y_{n-1})/y_{n-1}].$$

We note that $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$-$E$ and $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$-$E'$ really are equivalent in a precise sense.

Theorem 2.1. *We have translations between $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$-$E$ and $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$-$E'$. These become mutually inverse in the equational theory of DTT.*

Proof. Let us write B for $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$. In presence of B-E', we can define B-E by noting that

$$\frac{\frac{x : A, y : B, z_{1,1} : C[a_1/x, b_{1,1}/y], \dots, z_{n,m_n} : C[a_n/x, b_{n,m_n}/y] \vdash z_{i,j} : C[a_i/x, b_{i,j}/y]}{x : A, y : B, z_{1,1} : C[a_1/x, b_{1,1}/y], \dots, z_{n,m_n} : C[a_n/x, b_{n,m_n}/y], p_{i,j} : \mathsf{Id}_A(a_i, x), q : \mathsf{Id}_B(\mathsf{subst}(p_{i,j}, b_{i,j}), y) \vdash z_{i,j} : C[a_i/x, b_{i,j}/y]}}{x : A, y : B, z_{1,1} : C[a_1/x, b_{1,1}/y], \dots, z_{n,m_n} : C[a_n/x, b_{n,m_n}/y], p_{i,j} : \mathsf{Id}_A(a_i, x), q_{i,j} : \mathsf{Id}_B(\mathsf{subst}(p_{i,j}, b_{i,j}), y) \vdash \mathsf{subst}(q_{i,j}, \mathsf{subst}(p_{i,j}, z_{i,j})) : C}$$

and applying B-E' with $A' = \Sigma_{x:A}\Sigma_{y:B}\Sigma_{z_{1,1}:C[a_1/x,b_{1,1}/y]} \cdots \Sigma_{z_{n,m_{n-1}}:C[a_n/x,b_{n,m_n-1}/y]} C[a_n/x, b_{n,m_n}/y]$, $a = x$ (the projection to A), $b = y$ (the projection to B) and the from $z_{i,j}$ derived expression above for $c_{i,j}$ to derive B-E'.

Conversely, in presence of B-E, we derive B-E':

$$\frac{\frac{\frac{\{x' : A', p_{i,j} : \mathsf{Id}_A(a_i, a), q_{i,j} : \mathsf{Id}_{B[a/x]}(\mathsf{subst}(p_{i,j}, b_{i,j}), b) \vdash c_{i,j} : C[b/y]\}_{i,j}}{\{\vdash \lambda_{x':A'} \lambda_{p_{i,j}:\mathsf{Id}_A(a_i,a)} \lambda_{q_{i,j}:\mathsf{Id}_{B[a/x]}(\mathsf{subst}(p_{i,j},b_{i,j}),b)} c_{i,j} : \Pi_{x':A'} \Pi_{p_{i,j}:\mathsf{Id}_A(a_i,a)} \Pi_{q_{i,j}:\mathsf{Id}_{B[a/x]}(\mathsf{subst}(p_{i,j},b_{i,j}),b)} C[b/y]\}_{i,j}}}{x : A, y' : B \vdash \mathsf{case}_{B, \Pi_{x':A'} \Pi_{p:\mathsf{Id}_A(x,a)} \Pi_{q:\mathsf{Id}_{B[a/x]}(\mathsf{subst}(p,y'),b)} C[b/y]}(y', \{\lambda_{x':A'} \lambda_{p_{i,j}:\mathsf{Id}_A(a_i,a)} \lambda_{q_{i,j}:\mathsf{Id}_{B[a/x]}(\mathsf{subst}(p_{i,j},b_{i,j}),b)} c_{i,j}\}) : \Pi_{x':A'} \Pi_{p:\mathsf{Id}_A(x,a)} \Pi_{q:\mathsf{Id}_{B[a/x]}(\mathsf{subst}(p,y'),b)} C[b/y]}}{x' : A' \vdash \mathsf{case}_{B, \Pi_{x':A'} \Pi_{p:\mathsf{Id}_A(x,a)} \Pi_{q:\mathsf{Id}_{B[a/x]}(\mathsf{subst}(p,y'),b)} C[b/y]}(y', \{\lambda_{x':A'} \lambda_{p_{i,j}:\mathsf{Id}_A(a_i,a)} \lambda_{q_{i,j}:\mathsf{Id}_{B[a/x]}(\mathsf{subst}(p_{i,j},b_{i,j}),b)} c_{i,j}\})[a/x, b/y'](x', \mathsf{refl}_a, \mathsf{refl}_b) : C[b/y]}$$

These translations are easily seen to be mutually inverse in their translation to STT, which we define in the next section, due to the $\{b_{i,j} \mid i, j\}$-Comm-λ-rule. Therefore, they are mutually inverse in DTT. □

We conclude that case and $\mathsf{case}^{p,q}$ are equivalent. We prefer to use the latter as the default, as it naturally arises in the completeness proofs later in this paper. For the purposes of proof theory, however, the rules of figure 7 may be the preferred choice, as the metatheory of the resulting system is known to be well-behaved (at least in absence of the commutative conversions).

Remark 2.2. *We have restricted finite inductive type families to dependency on types A that are built from the 1-type, finite inductive type families and their Σ- and Id-types. This restriction precisely makes sure that* ***all finite inductive type families respect observational equivalence***[2]*, which is essential for our definability proofs to carry through later. For instance, a type family $z : \{\mathsf{tt}, \mathsf{ff}\} \Rightarrow \{\mathsf{tt}, \mathsf{ff}\} \vdash A \ \mathsf{type}$ will have $\vdash A[\lambda_{x:\{\mathsf{tt},\mathsf{ff}\}} \mathsf{tt}/z] = A[\lambda_{x:\{\mathsf{tt},\mathsf{ff}\}} \mathsf{case}_{\{\mathsf{tt},\mathsf{ff}\},\{\mathsf{tt},\mathsf{ff}\}}(x, \mathsf{tt}, \mathsf{tt})/z]$ because $\lambda_{x:\{\mathsf{tt},\mathsf{ff}\}} \mathsf{tt}$ and $\lambda_{x:\{\mathsf{tt},\mathsf{ff}\}} \mathsf{case}_{\{\mathsf{tt},\mathsf{ff}\},\{\mathsf{tt},\mathsf{ff}\}}(x, \mathsf{tt}, \mathsf{tt})$ are observationally equivalent: both compute the constant function which returns tt. We note that in the setting where one has a universe and finite inductive types, rather than an explicit mechanism for generating finite inductive type families, the finite type families we can build automatically have this property.*

---

we can define a suitable type family D over $\Sigma_{x:A}(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$ that has fibre $\{\}$ over $\langle a_{i'}, b_{i',j'} \rangle$ and 1 over $\langle a_i, b_{i,j} \rangle$ to get $p : \mathsf{Id}_A(a_i, a_{i'}), q : \mathsf{Id}_{(a_i \mapsto_i \{b_{i,j} \mid j\})(a_{i'})}(\mathsf{subst}(p, b_{i,j}), b_{i',j'}) \vdash \mathsf{subst}(q, \mathsf{subst}(p, *)) : \{\}$.

[2] We call terms $\vdash a, a' : A$ observationally equivalent if for all contexts $C[-]$ of type $\{\mathsf{tt}, \mathsf{ff}\}$, we have that $\vdash C[a] = C[a'] : \{\mathsf{tt}, \mathsf{ff}\}$. This captures the idea that a and a' are extensionally equal, in a sense.

2.4. A Syntactic Translation from DTT to STT

Morally, DTT should describe the same algorithms as STT (at least at the type hierarchy over finite types), possibly assigning them a more precise type. Formally, this idea is captured by the existence of a syntactic translation from DTT(−) into STT. By noting that it is compositional and faithful on all term constructors, we note that we can add to DTT− the equational theory of STT under this translation. We refer to the theory we obtain as DTT. Some examples of equations this implies are the usual β- and η-laws for 1-, Σ- and Π-types and finite inductive type families, β-laws for Id-types and commutative conversions for the case-constructs. We note that we then have a faithful translation from DTT to STT.

This translation will later suggest our game theoretic interpretation of dependent type theory, by demanding that it agrees with (a total, finitary equivalent of) the usual PCF game semantics [8] after transforming the syntax. A semantically inclined reader may want to think about the translation we define as a faithful non-full functor $(-)^T$ from the syntactic category (or, category of contexts) $\mathsf{Syntax}_{\mathsf{DTT}}$ of DTT to the syntactic category $\mathsf{Syntax}_{\mathsf{STT}}$ of STT.

The translation $(-)^T$ is inductively defined on types and terms through the following schema.

$\vdash b_{i,j} : (a_i \mapsto_i \{b_{i,j} \mid j\})(a_i)$	$\mapsto$	$\vdash b_{i,j} : \{b_{i,j} \mid i, j\}$
$x : A, y : B, z_{1,1} : C[a_1/x, b_{1,1}/y], \dots, z_{n,m_n} : C[a_n/x, b_{n,m_n}/y] \vdash \mathsf{case}_{B,C}(y, \{z_{i,j}\}_{i,j}) : C$	$\mapsto$	$x : A^T, y : B^T, z_{1,1} : C^T, \dots, z_{n,m_n} : C^T \vdash \mathsf{case}_{B^T,C^T}(y, \{z_{i,j}\}_{i,j}) : C^T$
$x' : A' \vdash \mathsf{case}^{p,q}_{B[a/x],C}(b, \{c_{i,j}\}_{i,j}) : C[b/y]$	$\mapsto$	$x' : (A')^T \vdash \mathsf{case}_{B^T,C^T}(b^T, \{c^T_{i,j}[a^T/p_{i,j}, b^T/q_{i,j}]\}_{i,j}) : C^T$
$x : A \vdash \langle\rangle : 1$	$\mapsto$	$x : A^T \vdash \langle\rangle : 1$
$x : A \vdash \langle b, c\rangle : \Sigma_{y:B} C$	$\mapsto$	$x : A^T \vdash \langle b^T, c^T\rangle : B^T \times C^T$
$x : A \vdash \mathsf{fst}(d) : B$	$\mapsto$	$x : A^T \vdash \mathsf{fst}(d^T) : B^T$
$x : A \vdash \mathsf{snd}(d) : C[\mathsf{fst}(d)/y]$	$\mapsto$	$x : A^T \vdash \mathsf{snd}(d^T) : C^T$
$x : A \vdash \lambda_{y:B} c : \Pi_{y:B} C$	$\mapsto$	$x : A^T \vdash \lambda_{y:B^T} c^T : B^T \Rightarrow C^T$
$x : A \vdash f(b) : C[b/y]$	$\mapsto$	$x : A^T \vdash f^T(b^T) : C^T$
$x : A \vdash \mathsf{refl}_b : \mathsf{Id}_B(b, b)$	$\mapsto$	$x : A^T \vdash b^T : B^T$
$x : A \vdash \mathsf{let}\ p\ \mathsf{be}\ \mathsf{refl}_z\ \mathsf{in}\ d : D[b/y, b'/y', p/w]$	$\mapsto$	$x : A^T \vdash d^T[p^T/z] : D^T$
$p : \mathsf{Id}_A(a_i, a_{i'}), q : \mathsf{Id}_{B[a_{i'}/x]}(\mathsf{subst}(p, b_{i,j}), b_{i',j'}) \vdash \mathsf{exfalso} : \mathsf{Id}_C(c, c')$	$\mapsto$	$p : A^T, q : B^T \vdash c^T : C^T$
$\Gamma \vdash c[b/y] : C[b/y]$	$\mapsto$	$\Gamma^T \vdash c^T[b^T/y] : C^T$
$\Gamma, x : A, \Delta \vdash x : A$	$\mapsto$	$\Gamma^T, x : A^T, \Delta^T \vdash x : A^T$

Figure 8. A syntactic translation from DTT− into STT.

Finally, we define DTT as the theory generated by the rules of DTT− together with the following rule which says that DTT inherits the judgemental equalities of STT and the obvious congruence rules which state that all type and term formers respect judgemental equality. This gives us a concise way of equipping DTT with the appropriate β- and η-rules for its type formers as well as all necessary commutative conversions.

$$\frac{\Gamma \vdash_{\mathsf{DTT}} a : A \qquad \Gamma \vdash_{\mathsf{DTT}} b : A \qquad \Gamma^T \vdash_{\mathsf{STT}} a^T \equiv b^T : A^T}{\Gamma \vdash_{\mathsf{DTT}} a \equiv b : A}\ \mathsf{DTT\text{-}Eq}$$

Figure 9. The final rule which DTT has on top of DTT−, letting it inherit the equational theory of STT.

We note that, by induction, $(-)^T$ respects the judgemental equalities introduced by the rule above, meaning that $(-)^T$ defines a translation from DTT to STT. This lets us conclude the following.

Corollary 2.3. *The translation $(-)^T$ defined above defines a faithful translation from DTT to STT.*

We observe that we have defined a flavour of intensional type theory.

Remark 2.4. *We easily see that DTT refutes the principle of equality reflection,*

$$\frac{\Gamma \vdash p : \mathsf{Id}_A(f, g)}{\Gamma \vdash f \equiv g : A}\ \text{Equality Reflection,}$$

as, for instance, for $\Gamma = x : \{a\}$ and $A = \{a\}$, $f = \mathsf{case}_{\{a\},\{a\}}(x, a)$ and $g = a$, we do not have that $x : \{a\} \vdash f \equiv g : \{a\}$, while we do have $x : \{a\} \vdash \mathsf{case}_{\{a\},\mathsf{Id}_{\{a\}}(f,g)}(x, \mathsf{refl}_a) : \mathsf{Id}_{\{a\}}(f, g)$.

3. A Category of Games

The idea behind game semantics is to model a computation by an alternating sequence of interactions (the play) between a program (Player) and its environment (Opponent), following some rules specified by its datatype (the game). In this translation, programs become Player strategies, while termination corresponds to a strategy being winning or beating all Opponents. The charm of this interpretation is that it not only fully captures the intensional aspects of a program but that it combines this with the structural clarity of a categorical model, thus interpolating between traditional operational and denotational semantics.

We assume the reader has some familiarity with the basics of categories of AJM-games and ($\approx$-saturated[3]) strategies, as described in [27], and will only briefly recall the definitions. We define a category **Game** which has as objects AJM-games.

Definition 3.1 (Game)**.** *A **game** A is a tuple $(M_A, \lambda_A, P_A, \approx_A, W_A)$, where*

- *M_A is a countable set of **moves**;*
- *$M_A \xrightarrow{\lambda_A = \langle \lambda_A^{OP}, \lambda_A^{QA} \rangle} \{O,P\} \times \{Q,A\}$ is a function which indicates if a move is made by **Opponent** (O) **or Player** (P) and if it is a **Question** (Q) or an **Answer** (A), for which we write $\overline{O} = P$, $\overline{P} = O$ and $M_A^O := {\lambda_A^{OP}}^{-1}(O)$, $M_A^P := {\lambda_A^{OP}}^{-1}(P)$, $M_A^Q := {\lambda_A^{QA}}^{-1}(Q)$ and $M_A^A := {\lambda_A^{QA}}^{-1}(A)$;*
- *$P_A \subseteq M_A^{\circledast}$ is a non-empty prefix-closed set of **plays**, where $M_A^{\circledast}$ is the set of finite sequences of moves, with the properties*

  (p1) $s = at \Rightarrow a \in M_A^O$;

  (p2) $\forall_i \lambda_A^{OP}(s_{i+1}) = \overline{\lambda_A^{OP}(s_i)}$, *where we write s_i for the i-th move in s;*

  (p3) $\forall_{t \leq s} |t \upharpoonright_{M_A^A}| \leq |t \upharpoonright_{M_A^Q}|$.

  *Here, $\leq$ denotes the prefix order and $|s|$ the length of a sequence. For an answer $a \in M_A^A$ given after a play s, let us write $\mathrm{j}_A(sa) \in M_A^Q$ for the last unanswered question preceding it in sa (the pending question), which we say a answers. Defining $\mathrm{j}_A(sq) := \epsilon$ for a question $q \in M_A^Q$, we inductively define $\mathrm{j}_A^*(sa) := \mathrm{j}_A^*(s)\,\mathrm{j}_A(sa)$. j_A will be used to enforce **stack discipline**.*

- *$\approx_A$ is an equivalence relation on P_A, satisfying*

  (e1) $s \approx_A t \Rightarrow \lambda_A^*(s) = \lambda_A^*(t)$;

  (e2) $s \approx_A t \ \wedge\ s' \leq s \ \wedge\ t' \leq t \ \wedge\ |s'| = |t'| \ \Rightarrow s' \approx_A t'$;

  (e3) $s \approx_A t \wedge sa \in P_A \Rightarrow \exists_b sa \approx_A tb$.

  Here, λ_A^* is the extension of λ_A to sequences. The intuition is that $\approx_A$-equivalent plays represent the same computation performed using different threads.

- *$W_A \subseteq P_A^{\infty}$ is a set of **winning plays**, where P_A^{∞} is the set of infinite plays, i.e. infinite sequences of moves such that all their finite prefixes are in P_A, such that W_A is closed under $\approx_A$ in the sense that*

$$(s \in W_A \wedge t \notin W_A) \Rightarrow \exists_{s_0 \leq s, t_0 \leq t} |s_0| = |t_0| \wedge s_0 \not\approx_A t_0.$$

  The intuition is that Opponent is the one who caused interactions in W_A to be infinite.

Our notion of morphism will be defined in terms of strategies on games.

Definition 3.2 (Strategy)**.** *A **strategy on** A is a non-empty subset $\sigma \subseteq P_A^{\mathsf{even}}$ satisfying*

[3] Note that this is a mild technical difference from the formalism of [8], where strategies are what we call skeletons, here, which are considered up to a partial equivalence relation induced by $\approx$. Both formalisms are equivalent as a class of skeletons up to this partial equivalence relation can precisely be identified with the unique strategy obtained by closing the plays of the skeleton under $\approx$.

(Causal Consistency): $sab \in \sigma \Rightarrow s \in \sigma$;

(Representation Independence): $s \in \sigma \ \wedge \ s \approx_A t \Rightarrow t \in \sigma$;

(Determinacy): $sab, ta'b' \in \sigma \ \wedge \ sa \approx_A ta' \Rightarrow sab \approx_A ta'b'$.

We sometimes identify σ with the subset of P_A that is obtained as its prefix closure. We restrict from the (history-sensitive) strategies defined so far to history-free strategies, as we are modelling computation without mutable state.

Definition 3.3 (History-Free Strategy)**.** *We call a strategy* $\sigma \in \mathsf{str}(A)$ ***history-free****, if there exists a non-empty causally consistent subset* $\phi \subseteq \sigma$ *(called a* ***history-free skeleton****) such that*

(Uniformization): $\forall_{sab \in \sigma} s \in \phi \Rightarrow \exists!_{b'} sab' \in \phi$;

(History-Freeness 1): $sab, tac \in \phi \Rightarrow b = c$;

(History-Freeness 2): $(sab, t \in \phi \ \wedge \ ta \in P_A) \Rightarrow tab \in \phi$.

Then, ϕ is induced by a partial function on moves and $\sigma = \{t \mid \exists_{s \in \phi} t \approx_A s\}$. From now on, we assume strategies to be history-free, unless explicitly stated otherwise. We write $\mathsf{str}(A)$ for the cpo of (history-free) strategies on A ordered under inclusion and write $\bot_A$ or simply $\bot$ for the empty strategy on A. Winning conditions give rise to the notion of a winning strategy, the semantic equivalent of a normalising or total term. A winning strategy always has a response to any valid O-move. Furthermore, if the result of the interaction between a winning strategy and any (possibly history-sensitive) Opponent is an infinite play, then this is a member of the set of winning plays, capturing the idea that the infinite interaction is Opponent's fault.

Definition 3.4 (Winning Strategy)**.** *A strategy* $\sigma \in \mathsf{str}(A)$ *is* ***winning*** *if it satisfies*

(Finite Wins): If s *is* $\leq$*-maximal in* σ*, then* s *is* $\leq$*-maximal in* P_A.

(Infinite Wins): If $s_0 \leq s_1 \leq \ldots$ *is an infinite chain in* σ*, then* $\bigcup_i s_i \in W_A$.

We write $\mathsf{wstr}(A)$ for the set of winning (history-free) strategies on A. Next, we define some constructions on games, starting with their symmetric monoidal closed structure.

Definition 3.5 (Tensor Unit)**.** *We define the game* $I := (\emptyset, \emptyset, \{\epsilon\}, \{(\epsilon, \epsilon)\}, \emptyset)$.

Definition 3.6 (Tensor)**.** *Given games A and B, we define the game* $A \otimes B$ *by*

- $M_{A \otimes B} := M_A + M_B = \Sigma_{i \in \{A,B\}} M_i$; *we write* fst *for the first projection of this* Σ*-type;*
- $\lambda_{A \otimes B} := [\lambda_A, \lambda_B]$;
- $P_{A \otimes B} := \{s \in M^{\circledast}_{A \otimes B} \mid s \restriction_A \in P_A \wedge s \restriction_B \in P_B \wedge \mathsf{fst}^*(\mathsf{j}^*_{A \otimes B}(s)) = \mathsf{fst}^*(s \restriction_{M^A_{A \otimes B}})\}$;
- $s \approx_{A \otimes B} t := s \restriction_A \approx_A t \restriction_A \ \wedge \ s \restriction_B \approx_B t \restriction_B \ \wedge \forall_{1 \leq i \leq |s|} (s_i \in M_A \Leftrightarrow t_i \in M_A)$;
- $W_{A \otimes B} := \left\{ s \in P^{\infty}_{A \otimes B} \mid \left(s \restriction_A \in P^{\infty}_A \Rightarrow s \restriction_A \in W_A \right) \wedge \left(s \restriction_B \in P^{\infty}_B \Rightarrow s \restriction_B \in W_B \right) \right\}$.

Definition 3.7 (Linear Implication)**.** *Given games A and B, we define the game* $A \multimap B$ *by*

- $M_{A \multimap B} := M_A + M_B = \Sigma_{i \in \{A,B\}} M_i$;
- $\lambda_{A \multimap B} := [\overline{\lambda_A}, \lambda_B]$, *where* $\overline{\lambda_A}(m) := \overline{\lambda_A(m)}$;
- $P_{A \multimap B} := \{s \in M^{\circledast}_{A \multimap B} \mid s \restriction_A \in P_A \wedge s \restriction_B \in P_B \wedge \mathsf{fst}^*(\mathsf{j}^*_{A \multimap B}(s)) = \mathsf{fst}^*(s \restriction_{M^A_{A \multimap B}})\}$;
- $s \approx_{A \multimap B} t := s \restriction_A \approx_A t \restriction_A \ \wedge \ s \restriction_B \approx_B t \restriction_B \ \wedge \forall_{1 \leq i \leq |s|} (s_i \in M_A \Leftrightarrow t_i \in M_A)$;
- $W_{A \multimap B} := \{s \in P^{\infty}_{A \multimap B} \mid s \restriction_A \in W_A \Rightarrow s \restriction_B \in W_B\}$.

I is the unique game whose only play has length 0. Both $A \otimes B$ and $A \multimap B$ are obtained by playing A and B in parallel by interleaving. Note that the definitions of $P_{\_}$ and $\lambda_{\_}$ imply that in $A \otimes B$ only Opponent can switch between A and B, while in $A \multimap B$ only Player can. The stack discipline in both cases implies that a question is answered in the game where it was asked.

These definitions on objects extend to strategies, e.g. for (winning) strategies $\sigma \in \mathsf{str}(A), \tau \in \mathsf{str}(B)$, we can define a (winning) strategy $\sigma \otimes \tau = \{s \in P^{\mathsf{even}}_{A \otimes B} \mid s \upharpoonright_A \in \sigma \ \wedge \ s \upharpoonright_B \in \tau\} \in \mathsf{str}(A \otimes B)$. This gives us a model of multiplicative intuitionistic linear logic, with all structural morphisms consisting of appropriate variants of copycat strategies, which are introduced next.

Theorem 3.8 (Linear Category of Games)**.** *We define a category* **Game** *by*

- $\mathsf{ob}(\mathbf{Game}) := \{A \mid A$ is an AJM-game$\}$;
- $\mathbf{Game}(A, B) := \mathsf{wstr}(A \multimap B)$;
- $\mathsf{id}_A := \{s \in P^{\mathsf{even}}_{A \multimap A} \mid \forall_{s' \in P^{\mathsf{even}}_{A \multimap A}} s' \leq s \Rightarrow s' \upharpoonright_{A^{(1)}} \approx_A s' \upharpoonright_{A^{(2)}}\}$, *the* ***copycat strategy*** *on* A;
- *for* $A \xrightarrow{\sigma} B \xrightarrow{\tau} C$, *the composition (or* ***interaction****)* $A \xrightarrow{\sigma;\tau} C$ *is defined from parallel composition* $\sigma \| \tau := \{s \in M^{\circledast}_{(A \multimap B) \multimap C} \mid s \upharpoonright_{A,B} \in \sigma \ \wedge \ s \upharpoonright_{B,C} \in \tau\}$ *plus hiding:* $\sigma; \tau := \{s \upharpoonright_{A,C} \mid s \in \sigma \| \tau\}$.

Then, $(\mathbf{Game}, I, \otimes, \multimap)$ *is, in fact, a symmetric monoidal closed category.*

To make this into a model of intuitionistic logic, a Cartesian closed category (ccc), through the (first) Girard translation, we need two more constructions on games, to interpret the additive conjunction and exponential, respectively.

Definition 3.9 (With)**.** *Given games A and B, we define the game* $A \& B$ *by*

- $M_{A \& B} := M_A + M_B$;
- $\lambda_{A \& B} := [\lambda_A, \lambda_B]$;
- $P_{A \& B} := P_A + P_B$;
- $\approx_{A \& B} := \approx_A + \approx_B$;
- $W_{A \& B} := W_A + W_B$.

Definition 3.10 (Bang)**.** *Given a game A, we define the game* $!A$ *by*

- $M_{!A} := \mathbb{N} \times M_A$;
- $\lambda_{!A}(i, a) := \lambda_A(a)$;
- $P_{!A} := \{s \in M^{\circledast}_{!A} \mid \forall_{i \in \mathbb{N}} s \upharpoonright_i \in P_A \ \wedge \ \mathsf{fst}^*(\mathsf{j}^*_{!A}(s)) = \mathsf{fst}^*(s \upharpoonright_{M^A_{!A}})\}$;
- $s \approx_{!A} t := \exists_{\pi \in S(\mathbb{N})} \forall_{i \in \mathbb{N}} s \upharpoonright_i \approx_A t \upharpoonright_{\pi(i)} \ \wedge (\mathsf{fst}; \pi)^*(s) = \mathsf{fst}^*(t)$, *writing* $S(\mathbb{N})$ *for the set of permutations of* $\mathbb{N}$;
- $W_{!A} := \{s \in P^{\infty}_{!A} \mid \forall_i s \upharpoonright_i \in P^{\infty}_A \Rightarrow s \upharpoonright_i \in W_A\}$.

A play in $A \& B$ consists of either a play in A or in B, where Opponent chooses which as by our convention Opponent always makes the initial move. A play in $!A$ consists of any number of interleaved **threads** of plays in A. Because of the definition of $\approx_A$, $!A$ behaves as a countably infinite symmetric $\otimes$-product of A with itself. As before, the definition of $\lambda_{!A}$ assures only Opponent can switch games, while the stack discipline ensures that a question is answered in the thread in which it was asked.

Next, we note that ! can be made into a co-monad by defining, for $A \xrightarrow{\sigma} B$,

$$!\sigma := \{s \in P^{\mathsf{even}}_{!A \multimap !B} \mid \exists_{\pi \in S(\mathbb{N})} \forall_{i \in \mathbb{N}} s \upharpoonright_{(\pi(i),A),(i,B)} \in \sigma\},$$

and natural transformations

$$!A \xrightarrow{\mathsf{der}_A} A := \{s \in P^{\mathsf{even}}_{!A \multimap A} \mid \forall_{s' \in P^{\mathsf{even}}_{!A \multimap A}} s' \leq s \Rightarrow \exists_{i \in \mathbb{N}} s' \upharpoonright_{!A} \upharpoonright_i \approx_A s' \upharpoonright_A\} \quad \text{and}$$

$$!A \xrightarrow{\delta_A} !!A := \{s \in P^{\mathsf{even}}_{!A \multimap !!A} \mid \forall_{s' \in P^{\mathsf{even}}_{!A \multimap !!A}} s' \leq s \Rightarrow \exists_{p:\mathbb{N}\times\mathbb{N}\hookrightarrow\mathbb{N}} \forall_{i,j\in\mathbb{N}} s' \restriction_{!A} \restriction_{p(i,j)} \approx_A s' \restriction_{!!A} \restriction_i \restriction_j\}.$$

This allows us to define the co-Kleisli category $\mathbf{Game}_!$, which has the same objects as **Game**, while $\mathbf{Game}_!(A,B) := \mathbf{Game}(!A, B)$. Let us write $\mathsf{dom}(f)$ for the domain of a morphism f. We have a composition $(f,g) \mapsto f^\dagger; g$, where we write $f^\dagger := \delta_{\mathsf{dom}(f)}; !(f)$, for which the strategies der_A serve as identities. We can define nullary and binary products (and from those all finite products) in $\mathbf{Game}_!$ by I and & and write

$$\mathsf{diag}_A := \left\{s \in P^{\mathsf{even}}_{!A \multimap (A\&A)} \mid \forall_{s' \in P^{\mathsf{even}}_{!A\multimap(A\&A)}} s' \leq s \Rightarrow \exists_{i\in\mathbb{N}} (s' = \epsilon) \vee (s' \restriction_{!A} \restriction_i \approx_A s' \restriction_{A^{(1)}} \neq \epsilon) \vee (s' \restriction_{!A} \restriction_i \approx_A s' \restriction_{A^{(2)}} \neq \epsilon)\right\}$$

for the diagonal $!A \longrightarrow A\&A$. Moreover, we have Seely-isomophisms $!I \cong I$ and $!(A\&B) \cong !A \otimes !B$, so we obtain a linear-non-linear adjunction $\mathbf{Game} \leftrightarrows \mathbf{Game}_!$, hence a model of multiplicative exponential intuitionistic linear logic. In particular, by defining $A \Rightarrow B := !A \multimap B$, we make $\mathbf{Game}_!$ into a ccc. We write $\mathsf{comp}_{A,B,C}$ for the internal composition $((A \Rightarrow B)\ \&\ (B \Rightarrow C)) \longrightarrow A \Rightarrow C$ in $\mathbf{Game}_!$.

Theorem 3.11 (Intuitionist Category of Games)**.** $(\mathbf{Game}_!, I, \&, \Rightarrow)$ *is a ccc.*

Remark 3.12. *Note that for the hierarchy of intuitionistic types A that are formed by operations I, & and* $\Rightarrow$ *from finite games (games A with finite* P_A*), winning strategies are the total strategies – strategies which respond to any O-move – for which infinite plays can only occur because Opponent opens infinitely many threads of the same game.*

So far, we have shown that $\mathbf{Game}_!$ provides a model of STT without finite inductive types. It is not clear that inductive types and (weak) co-products are supported in our large world $\mathbf{Game}_!$ of all games[4]. To support these - to be precise, in order to define the appropriate eliminators -, we restrict the games we consider.

We can in fact construct a model of all of STT in a full subcategory $\mathbf{Game}_!^{\mathsf{fin1}\times\Rightarrow}$ of $\mathbf{Game}_!$ by giving a suitable interpretation to finite inductive types, which serve as the ground types for a type hierarchy built with 1, $\times$ and $\Rightarrow$. For a set X, let us define X_* to be a so-called **flat game** with $P_{X_*} = \{\epsilon, *\} \cup \{*x \mid x \in X\}$ and $\approx_X = \{(s,s) \in P_X \times P_X\}$ where the initial move $*$ is a question and the moves from X are answers. Let us interpret a finite inductive type $\{a_i \mid i\}$ as a finite flat game $\{a_i \mid i\}_*$. Let us write $\mathbf{Game}_!^{\mathsf{fin1}\times\Rightarrow}$ for the full subcategory of $\mathbf{Game}_!$ on the objects formed from finite flat games by 1, $\times$ and $\Rightarrow$. Then, $\mathbf{Game}_!^{\mathsf{fin1}\times\Rightarrow}$ is a model of STT.

Indeed, the interpretation of the introduction rule for a_i is the strategy which answers a_i to $*$, while the case-eliminators for finite inductive types are interpreted inductively on the structure of the type C we are eliminating into: the cases where C is not a finite inductive type are defined through $1 - \eta$, $\{a_i \mid i\} - \mathrm{Comm} - \langle -, - \rangle$ and $\{a_i \mid i\} - \mathrm{Comm} - \lambda$. In case C is a finite inductive type $\{c_j \mid j\}$, we interpret $\mathsf{case}_{\{a_i \mid i\},C}$ as the winning history-free strategy on $\{a_i \mid i\}_* \Rightarrow \{c_j \mid j\}_*{}^{(1)} \Rightarrow \ldots \Rightarrow \{c_j \mid j\}_*{}^{(n)} \Rightarrow \{c_j \mid j\}_*$ which is given the $\approx$-closure of the set of traces defined by the following partial function f on moves:

$$*^{(C)} \mapsto (0,*)^{(!\{a_i \mid i\}_*)} \quad (0,a_i) \mapsto (0,*)^{(!C^{(i)})} \quad (0,c_j)^{(!C^{(i)})} \mapsto c_j^{(C)}.$$

One of the interesting aspects of game semantics are the strong correspondences that can often be established with the syntax we are modelling. In this case, we have the following very strong completeness result. Note that fullness of the interpretation is strictly stronger than completeness: it is a notion of completeness with respect to proofs rather than mere provability.

Theorem 3.13. *The interpretation functor* $\mathsf{Syntax}_{STT} \xrightarrow{[\![-]\!]} \mathbf{Game}_!$ *is full and faithful and hence is an equivalence of categories to* $\mathbf{Game}_!^{\mathsf{fin1}\times\Rightarrow}$.

Proof. This is a straightforward finitary total variation on the results of [8]. We note that winning strategies are total and therefore the case of $\perp$ in the decomposition lemma does not occur. Moreover, we note that the iterated decomposition terminates if we start with a winning strategy, because of the proof of lemma 8.3 (essentially because infinite plays are always Opponent's responsibility, we can assign a finite size to a strategy which shrinks under the decomposition). □

[4] In fact, an alternative category of games and innocent strategies is used in [28] which mends precisely this defect.

4. An Indexed Category of Dependent Games

The previous section sketched how $\mathbf{Game}_!$ models simple intuitionistic type theory. Next, we show how we can equip it with a notion of dependent type. This leads to an indexed ccc $\mathbf{DGame}_!$ of dependent games and dependently typed winning strategies.

We define a poset $\mathbf{Game}_{\trianglelefteq}$ of games with order relation $A \trianglelefteq B := (M_A = M_B) \wedge (\lambda_A = \lambda_B) \wedge (P_A \subseteq P_B) \wedge \forall_{s,t\in P_B}(s \approx_A t \Leftrightarrow s \in P_A \wedge s \approx_B t) \wedge (W_A = W_B \cap P_A^\infty)$. Given a game C, we define the complete lattice $\mathsf{Sub}(C)$ as the poset of its $\trianglelefteq$-subgames. We note that, for $A, B \in \mathsf{Sub}(C)$, $A \trianglelefteq B \Leftrightarrow P_A \subseteq P_B$. We make the following simple observation that we shall refer to later.

Theorem 4.1. *We have a functor* $\mathbf{Game}_! \xrightarrow{\mathsf{Sub}} \mathsf{CLat}$ *to the category* CLat *of complete lattices and join-preserving functions.*

Proof. An element of $\mathsf{Sub}(C)$ is precisely specified by a $\approx_C$-closed prefix-closed subset of P_C, so we can compute joins and meets simply by unions and intersections. Given $A \xrightarrow{f} B \in \mathbf{Game}_!$ and $A' \trianglelefteq A$, we define $\mathsf{Sub}(f)(A') := \{s \in P_B \mid \exists_{t\in f} s \leq t \restriction_B \wedge \forall_i t \restriction_{!A} \restriction_i \in A'\}$. The result is clearly prefix-closed and closed under $\approx_B$, as f is closed under $\approx_{A\Rightarrow B}$. $\mathsf{Sub}(f)$ clearly preserves unions. □

This allows us to define a dependent game as follows, where we encourage the reader to think of $☺(B)$ as the semantic counterpart to the syntactic translation B^T of section 2.4.

Definition 4.2 (Dependent game)**.** *For a game A, we define the set* $\mathsf{ob}(\mathbf{DGame}_!(A))$ *of* ***games with dependency on*** *A as the set of continuous*[5] *functions* $\mathsf{str}(A) \xrightarrow{B} \mathsf{Sub}(☺(B))$ *for some other game* $☺(B)$*.*

We note that $\mathsf{ob}(\mathbf{DGame}_!(I))$ is the set of pairs $(A(\bot), ☺(A))$ where $A(\bot) \trianglelefteq ☺(A)$, in which $\mathsf{ob}(\mathbf{Game}_!)$ embeds as the proper subset of diagonal elements (A, A). As the definability results of section 8 illustrate, we need the generality of $\mathsf{ob}(\mathbf{DGame}_!(I))$ to properly capture the notion of closed typed in DTT. Therefore, we define, more generally, for a pair $(A, ☺(A)) \in \mathsf{ob}(\mathbf{DGame}_!(I))$, $\mathsf{ob}(\mathbf{DGame}_!(A(\bot), ☺(A))) := \mathsf{ob}(\mathbf{DGame}_!(☺(A)))$.

As an example, let us write $x : \mathsf{mm} \vdash \mathsf{dd} - \mathsf{mm}(x)$ for the (finite inductive) type family encoding the calendar of the year 1984 in dd-mm format (for instance, $\mathsf{dd} - \mathsf{mm}(02)$ has constructors 01-02,...,29-02). In this case, we note that, in the syntax of the type theory, the closed type $\mathsf{dd} - \mathsf{mm}(02)$ will behave differently from the closed (inductive) type $\{01\text{-}02, \ldots, 29\text{-}02\}$. Indeed, when eliminating from the former, our case analysis contains (redundant) additional information on how to handle the all other days of the year as well. This example shows that for a substituted type like $\mathsf{dd} - \mathsf{mm}(02)$ the type theory still remembers information about the whole type family $\mathsf{dd} - \mathsf{mm}$ (like the constructors outside the particular fibre under consideration), hence our interpretation of closed types as pairs $A(\bot) \trianglelefteq ☺(A)$ of games rather than as single games. From now on, we write A for the pair $(A(\bot), ☺(A)) \in \mathsf{ob}(\mathbf{DGame}_!(I))$ and, more generally and slightly ambiguously, B for the pair $(B, ☺(B)) \in \mathsf{ob}(\mathbf{DGame}_!(A))$.

Remark 4.3. *Alternatively, the reader may want to think of a game B with dependency on a game A as a continuous function* $\mathsf{str}(A)^\top \xrightarrow{B} \mathbf{Game}_{\trianglelefteq}$*, where we adjoin a top element $\top$ to* $\mathsf{str}(A)$*. In this view, $☺(B)$ corresponds to $B(\top)$. $\top$ may be viewed as an overdetermined element that is passed on under substitutions in the sense that a strategy f on $A' \Rightarrow A$ determines through post-composition the function* $\mathsf{str}(A')^\top \longrightarrow \mathsf{str}(A)^\top$ *which sends $\top$ to $\top$.*

Let us write $s \mapsto \overline{s}$ for the function from $P_{!☺(A)}$ to the power set $\mathcal{P}P_{☺(A)}$, inductively defined on the empty play, Opponent moves and Player moves, respectively, as $\epsilon \mapsto \emptyset$, $s(i,a) \mapsto \overline{s}$, $s(i,a)(i,b) \mapsto \overline{s(i,a)} \cup \{t \mid \exists_{s'\in\overline{s}} t \approx_{☺(A)} s'ab\}$. Morally, $\overline{s}$ represents the smallest (history-sensitive) strategy consistent with s. We then define the Π-game as follows.

[5] In fact, continuity and even monotonicity is not necessary for any of the proofs in this paper to go through. One might also restrict the domain to just $\mathsf{wstr}(A)$. If the model were to be extended with an interpretation of type universes, however, one would expect games with dependency to be embodied by certain strategies from A to a universe $\mathcal{U}$. We would expect these to induce functions $\mathsf{str}(A) \xrightarrow{B} \mathsf{Sub}(☺(B))$ that are not merely continuous but even sequential. Our definition of game with dependency can be seen as a first step towards this ultimate very intensional notion.

Definition 4.4 (Π-Game)**.** *Given* $A \in \mathsf{ob}(\mathbf{DGame}_!(I))$ *and* $B \in \mathsf{ob}(\mathbf{DGame}_!(A))$*, we define* $\Pi_A B \in \mathsf{ob}(\mathbf{DGame}_!(I))$ *with* $☺(\Pi_A B) := ☺(A) \Rightarrow ☺(B)$ *and* $(\Pi_A B)(\bot)$ *carved out in* $☺(\Pi_A B)$ *as follows*

$$\begin{aligned} P_{(\Pi_A B)(\bot)} :=& \{\epsilon\} \bigcup \\ & \{sa \in P^{\mathsf{odd}}_{☺(A)\Rightarrow ☺(B)} \mid s \in P^{\mathsf{even}}_{(\Pi_A B)(\bot)} \ \wedge \ \exists_{\overline{sa\restriction_{!☺(A)}} \subseteq \tau \in \mathsf{wstr}(A(\bot))} sa \in P_{A(\bot)\Rightarrow B(\tau)} \} \bigcup \\ & \{sab \in P^{\mathsf{even}}_{☺(A)\Rightarrow ☺(B)} \mid sa \in P^{\mathsf{odd}}_{(\Pi_A B)(\bot)} \ \wedge \ \forall_{\overline{sab\restriction_{!☺(A)}} \subseteq \tau \in \mathsf{wstr}(A(\bot))} sa \in P_{A(\bot)\Rightarrow B(\tau)} \Rightarrow sab \in P_{A(\bot)\Rightarrow B(\tau)} \}. \end{aligned}$$

We note that we can make $\mathbf{DGame}_!(A)$ into a ccc[6] by defining I and & pointwise on dependent games B, while also performing the operation on $☺(B)$, and by defining $☺(B \Rightarrow C) := ☺(B) \Rightarrow ☺(C)$ and $P_{(B\Rightarrow C)(\sigma)} := \{s \in P_{B(\sigma)\Rightarrow C(\sigma)} \mid \exists_{\tau \in \mathsf{wstr}(B(\sigma))} \overline{s\restriction_{B(\sigma)}} \subseteq \tau \}$. This lets us define $\mathbf{DGame}_!(A)(B,C) := \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_A(B \Rightarrow C)))$ with the obvious identity morphisms and composition, which we discuss later. Here, $\mathsf{ob}(\mathbf{DGame}_!(I)) \xrightarrow{\mathsf{O\text{-}sat}} \mathsf{ob}(\mathbf{Game}_!)$, sends $(A(\bot), ☺(A))$ to the game in which Opponent can play freely in $☺(A)$ and Player has to respect the rules of the more restrictive game $A(\bot)$ as long as Opponent does:

$$\begin{aligned} P_{\mathsf{O\text{-}sat}(A(\bot),☺(A))} :=& \{\epsilon\} \bigcup \\ & \{sa \in P^{\mathsf{odd}}_{☺(A)} \mid s \in P^{\mathsf{even}}_{\mathsf{O\text{-}sat}(A(\bot),☺(A))} \} \bigcup \\ & \{sab \in P^{\mathsf{even}}_{☺(A)} \mid sa \in P^{\mathsf{odd}}_{\mathsf{O\text{-}sat}(A(\bot),☺(A))} \ \wedge \ (sa \in P_{A(\bot)} \Rightarrow sab \in P_{A(\bot)})\}. \end{aligned}$$

Remark 4.5. *Note that, explicitly, the* ***game of dependent functions from*** *A* ***to*** *B,* $\mathsf{O\text{-}sat}(\Pi_A B)$*, is carved out in* $☺(A) \Rightarrow ☺(B)$*, as*

$$\begin{aligned} P_{\mathsf{O\text{-}sat}(\Pi_A B)} :=& \{\epsilon\} \bigcup \\ & \{sa \in P^{\mathsf{odd}}_{☺(A)\Rightarrow ☺(B)} \mid s \in P^{\mathsf{even}}_{\mathsf{O\text{-}sat}(\Pi_A B)} \} \bigcup \\ & \{sab \in P^{\mathsf{even}}_{☺(A)\Rightarrow ☺(B)} \mid sa \in P^{\mathsf{odd}}_{\mathsf{O\text{-}sat}(\Pi_A B)} \ \wedge \ \forall_{\overline{sab\restriction_{!☺(A)}} \subseteq \tau \in \mathsf{wstr}(A(\bot))} sa \in P_{A(\bot)\Rightarrow B(\tau)} \Rightarrow sab \in P_{A(\bot)\Rightarrow B(\tau)} \}. \end{aligned}$$

Indeed, this follows as $\overline{sab\restriction_{!☺(A)}} = \overline{sa\restriction_{!☺(A)}}$*. An explicit proof is given for the more general claim of theorem 5.3.*

Recall that we would like $☺(-)$ *to define a faithful functor to the world of simply typed games, being the semantic equivalent of* $(-)^T$*. It is for this reason that the game of dependent functions from A to B is saturated under all O-moves in* $☺(A) \Rightarrow ☺(B)$*. We present* $\mathsf{O\text{-}sat}$ *as a separate operation as this presentation will simplify the treatment of higher order dependent functions in section 5.*

Following the mantra of game semantics for quantifiers [9], in $\mathsf{O\text{-}sat}(\Pi_A B)$, Opponent can choose a winning strategy τ on $A(\bot)$ while Player has to play in a way that is compatible with all choices of τ that have not yet been excluded. Similarly to the approach taken in the game semantics for polymorphism [9], we do not specify all of τ in one go, as this would violate "Scott's axiom" of continuity of computation. Instead, τ is gradually revealed, explicitly so by playing in $!☺(A)$ and implicitly by playing in $☺(B)$. That is, unless Opponent behaves naughtily, in the sense that there is no winning history-free strategy τ on $A(\bot)$ which is consistent with her behaviour while $s\restriction_{☺(B)}$ obeys the rules of $B(\tau)$. In case of such a naughty Opponent, any further play in $☺(A) \Rightarrow ☺(B)$ is permitted. Summarising, Opponent does not need to respect the type dependency at all; Playes does and, in fact, has to play conservatively to be compatible with all possible fibres of B that Opponent might choose.

Remark 4.6. *In particular,* $\mathbf{DGame}_!(I)$ *is a ccc which has* $\mathbf{Game}_!$ *as the proper full subcategory on the objects of the form* (A, A)*. Note that the morphisms from A to B consist of the strategies on* $☺(A) \Rightarrow ☺(B)$ *for which Player plays along the rules of* $A(\bot) \Rightarrow B(\bot)$ *as long as Opponent does so and as long as there is a winning strategy on* $A(\bot)$ *which is consistent with her play.*

[6] Perhaps a more insightful way to think of this is as $\mathbf{DGame}_!(A)$ being obtained as a co-Kleisli category for a linear exponential co-monad ! on a symmetric monoidal closed category $\mathbf{DGame}(A)$. Here, $\mathbf{DGame}(A)$ has the same objects as $\mathbf{DGame}_!(A)$ on which we define operations I, $\otimes$, $\multimap$ pointwise, while also performing the operation on $☺(B)$, and $☺(!B) := !☺(B)$ while $(!B)(\sigma) := \{s \in P_{!(B(\sigma))} \mid \exists_{\tau \in \mathsf{wstr}(B(\sigma))} \overline{s} \subseteq \tau \}$. We define $\mathbf{DGame}(A)(B,C) := \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_A(B \multimap C))$ with the obvious identity morphisms and composition.

For a function $Y \xrightarrow{X} \mathsf{Set}$ to the class Set of sets, we define $☺(X_*) := (\bigcup_{y\in Y} X(y))_*$ and $X_*(y) := X(y)_*$. For an example of non-constant type dependency, write $\mathsf{days}(n) := \{m \mid \text{there are} > m \text{ days in the year } n\}$ and define $\mathsf{days}(\bot) = \emptyset$, $\mathsf{days}_*(n) := \mathsf{days}(n)_*$ to obtain a game depending on $\mathbb{N}_*$ (with $\mathsf{days}_*(n) = \mathbb{N}_{<365*}$ or $\mathbb{N}_{<366*}$). (Note that this will not correspond to a finite inductive type family as the fibres of the type are not disjoint.) Then, figure 10 gives four examples of valid dependently typed strategies.

$\mathbb{N}_*$	days_*
	*
	364

$!\mathbb{N}_*$	days_*
	*
$(i,*)$	
$(i,1984)$	
	365

$!\mathbb{N}_*$	days_*
	*
$(i,*)$	
$(i,1985)$	
$(i+1,*)$	
$(i+1,1986)$	
	365

$!\mathbb{N}_*$	$!\mathsf{days}_*$	days_*	
		*	O
	$(i,*)$		P
	(i,m)		O
		m	P
			O
			P

Figure 10. Three plays in $\mathsf{O\text{-}sat}(\Pi_{\mathbb{N}_*}\mathsf{days}_*)$ and one in $\mathsf{O\text{-}sat}(\Pi_{\mathbb{N}_*}(\mathsf{days}_* \Rightarrow \mathsf{days}_*))$. The first as all years have > 364 days, the second as 1984 was a leap year, the third as Player can play any move in $☺(\mathsf{days}_*) = \mathbb{N}_{<366*}$ after Opponent has not played along a (history-free) strategy on $\mathbb{N}_*$ and the fourth as Opponent makes the move m first, after which Player can safely copy it. In the paired moves, Player chooses an (irrelevant) index i. For an interpretation of the plays in $\mathsf{O\text{-}sat}(\Pi_{\mathbb{N}_*}\mathsf{days}_*)$, imagine them as a dialogue between a departmental education manager (Opponent) and an academic (Player) where Player gets to choose for every year the date that she promises to have marked the students' end-of-year exam. A cheeky academic might try to suggest that she'll return the marked exams every year on the 366th day of the year without asking the manager for which year he wants to know the date. Clearly the manager should not accept this. This corresponds to the play $*365$, which is illegal as, by making the move 365, Player would exclude certain fibres (the non-leap years), which is a privilege only Opponent has.

The fourth example is especially important, as it generalises to a (derelicted) B-copycat on $\mathsf{O\text{-}sat}(\Pi_A(B \Rightarrow B))$ for arbitrary B, denoted $\mathbf{v}_{[A],[B]}$ in section 5. This motivates why Opponent can narrow down the fibre of B freely, while Player can only play without narrowing down the fibre further. To see that Player should not be able to narrow down the fibre of B, note that we do not want $f := \{\epsilon, *365\}$ to define a strategy on $\mathsf{O\text{-}sat}(\Pi_{\mathbb{N}_*}\mathsf{days}_*)$, as $1983; f = \{\epsilon, *365\} \notin \mathsf{str}(\mathsf{days}_*(1983))$.

Theorem 4.7. *We obtain a strict indexed ccc*[7] $\mathbf{DGame}_!(I)^{op} \xrightarrow{(\mathbf{DGame}_!, -\{-\})} \mathsf{CCCat}$ *of dependent games, if we define*

- *fibrewise objects* $\mathsf{ob}(\mathbf{DGame}_!(A)) := \{\mathsf{str}(☺(A)) \xrightarrow{B} \mathsf{Sub}(☺(B)) \mid ☺(B) \in \mathsf{ob}(\mathbf{Game}_!) \ \wedge \ B \text{ continuous } \}$*;*
- *fibrewise hom-sets* $\mathbf{DGame}_!(A)(B,C) := \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_A(B \Rightarrow C)))$*;*
- *fibrewise identities* $\mathsf{der}_B := \{s \in P^{\mathsf{even}}_{\mathsf{O\text{-}sat}(\Pi_A(B\Rightarrow B))} \mid \forall_{s'\in P^{\mathsf{even}}_{\mathsf{O\text{-}sat}(\Pi_A(B\Rightarrow B))}} s' \leq s \Rightarrow \exists_i s' \restriction_{!☺(B)} \restriction_i \approx_{☺(B)} s' \restriction_{☺(B)}\}$*;*
- *for* $B \xrightarrow{\tau} C \xrightarrow{\tau'} D \in \mathbf{DGame}_!(A)$*, we define the fibrewise composition* $B \xrightarrow{\tau^\dagger;_A\tau'} D \in \mathbf{DGame}_!(A)$ *as* $\tau^\dagger;_A \tau' := \mathsf{diag}^\dagger_A; (\tau^\dagger \otimes \tau'); \mathsf{comp}_{☺(B),☺(C),☺(D)}$*;*
- *given* $f \in \mathbf{Game}_!(A', A)$*, we define the change of base functor* $-\{f\}$*:* $B\{f\} \in \mathsf{ob}(\mathbf{DGame}_!(A'))$ *where* $B\{f\}(\sigma) := B(!(\sigma); f)$ *and* $☺(B\{f\}) := ☺(B)$ *and* $\tau\{f\} := f^\dagger; \tau$*.*

Proof. We prove a more general result in theorem 5.6. □

Seeing that $\mathbf{DGame}_!(I)$ additionally has a terminal object I to interpret the empty context, we are well on our way to producing a model of dependent type theory [29]: we only need to interpret context extension. This takes the form of the comprehension axiom for $\mathbf{DGame}_!$, which states that for each $A \in \mathsf{ob}(\mathbf{DGame}_!(I))$ and $B \in \mathsf{ob}(\mathbf{DGame}_!(A))$ the following presheaf is representable

$$x \mapsto \mathbf{DGame}_!(\mathsf{dom}(x))(I, B\{x\}) : (\mathbf{DGame}_!(I)/A)^{op} \longrightarrow \mathsf{Set}.$$

Unfortunately, this fails, as $\mathbf{DGame}_!(I)$ does not yield a sound interpretation of dependent contexts. Essentially, the problem is that we do not have **additive Σ-types**, appropriate generalisations $\Sigma^{\&}_A B$ of & to interpret dependent context extension in $\mathbf{DGame}_!(I)$.

[7] That is, a functor from $\mathbf{DGame}_!(I)^{op}$ to the 1-category CCCat of cartesian closed categories and strong cartesian closed functors.

To get some intuition of why such objects may be problematic to interpret (in addition to the formal proof below), note that the usual product $A\&B$ in $\mathbf{Game}_!$ has an additive reading: we either have (a play of) A or B and Opponent chooses which. This means that we would expect a $\Sigma^{\&}_A B$-game, which should be a dependent generalisation of the ordinary product, to have an additive reading too. However, B represents a predicate on A, so we are in the situation that Opponent chooses whether we have a play embodying an object a of type A or a play embodying a property b (of type B) of a. This is like the paradoxical Cheshire cat of Alice in Wonderland of figure 11: let A be the type of cats and let B be the predicate "is grinning".

Figure 11. We encourage the reader to compare the idea of Σ-types in $\mathbf{Game}_!$ with Lewis Carroll's invention of the Cheshire cat. *"Well! I've often seen a cat without a grin," thought Alice; "but a grin without a cat! It's the most curious thing I ever saw in all my life."* [30]

Theorem 4.8. $\mathbf{DGame}_!$ *does not satisfy the comprehension axiom.*

Proof. Let us write $\mathbb{B} := \{\mathsf{tt}, \mathsf{ff}\}$. Then, $\mathbb{B}_*$ is the usual flat game of Booleans. We can define a dependent game just_* over $\mathbb{B}_*$, where $☺(\mathsf{just}_*) := \mathbb{B}_*$, $\mathsf{just}_*(\bot) = \emptyset_*$, $\mathsf{just}_*(\mathsf{ff}) = \{\mathsf{ff}\}_*$ and $\mathsf{just}_*(\mathsf{tt}) = \{\mathsf{tt}\}_*$.

Then, note that the comprehension axiom (supposing that it holds) implies that, for any $C \in \mathsf{ob}(\mathbf{DGame}_!(\mathbb{B}_*))$, $\mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{\mathbb{B}_*}(\mathsf{just}_* \Rightarrow C))) = \mathbf{DGame}_!(\mathbb{B}_*)(\mathsf{just}_*, C) = \mathbf{DGame}_!(\mathbb{B}_*)(I, \mathsf{just}_* \Rightarrow C) \cong \mathbf{DGame}_!(\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*)(I\{\mathbf{p}_{\mathbb{B}_*,\mathsf{just}_*}\}, C\{\mathbf{p}_{\mathbb{B}_*,\mathsf{just}_*}\}) = \mathbf{DGame}_!(\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*)(I, C\{\mathbf{p}_{\mathbb{B}_*,\mathsf{just}_*}\}) = \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*} C))$, where the third isomorphism follows from the defining equation of $\Rightarrow$ as a special case of a Π-type and where $\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_* \overset{\mathbf{p}_{\mathbb{B}_*,\mathsf{just}_*}}{\longrightarrow} \mathbb{B}_*$ is the representing object above for $A = \mathbb{B}_*$ and $B = \mathsf{just}_*$.

Now, taking $C(\tau) = I$ for all τ and $☺(C) = D$ for some game D, implies that $☺(\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*) \cong \mathbb{B}_*\&\mathbb{B}_*$. Indeed, we have a natural bijection $\mathbf{Game}_!(☺(\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*), D) = \mathsf{wstr}(☺(\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*) \Rightarrow D) = \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*} C)) = \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{\mathbb{B}_*}\mathsf{just}_* \Rightarrow C)) = \mathsf{wstr}(\mathbb{B}_* \Rightarrow \mathbb{B}_* \Rightarrow D) = \mathbf{Game}_!(\mathbb{B}_*, \mathbb{B}_* \Rightarrow D) \cong \mathbf{Game}_!(\mathbb{B}_*\&\mathbb{B}_*, D)$, which according to the Yoneda lemma is induced by an isomorphism $☺(\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*) \cong \mathbb{B}_*\&\mathbb{B}_*$ in $\mathbf{Game}_!$. (The crucial observation is that Opponent's first move immediately has to break the type dependency, as $C(\tau) = I$ for all τ.) According to theorem 4.1 this induces an isomorphism $\mathsf{Sub}(☺(\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*)) \cong \mathsf{Sub}(\mathbb{B}_*\&\mathbb{B}_*)$.

Therefore, symmetry of just in tt and ff implies that there are only nine options for $(\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*)(\bot)$: $I\&I$, $I\&\emptyset_*$, $\emptyset_*\&I$, $\emptyset_*\&\emptyset_*$, $I\&\mathbb{B}_*$, $\mathbb{B}_*\&I$, $\emptyset_*\&\mathbb{B}_*$, $\mathbb{B}_*\&\emptyset_*$ and $\mathbb{B}_*\&\mathbb{B}_*$. We take $C = \mathsf{just}_*$ in the bijection implied by the comprehension axiom above, to obtain $\mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{\mathbb{B}_*}(\mathsf{just}_* \Rightarrow \mathsf{just}_*))) \cong \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{\Sigma^{\&}_{\mathbb{B}_*}\mathsf{just}_*}\mathsf{just}_*))$. We see that none of the nine options is satisfactory. Indeed, $I\&I$, $I\&\emptyset_*$, $\emptyset_*\&I$, $\emptyset_*\&\emptyset_*$, $\mathbb{B}_*\&I$ and $\mathbb{B}_*\&\emptyset_*$ would imply that the negation between the two copies of just_* is a member of the right hand side, but not the left hand side, which is a contradiction. Similarly, $I\&I$, $I\&\emptyset_*$, $\emptyset_*\&I$, $\emptyset_*\&\emptyset_*$, $I\&\mathbb{B}_*$ and $\emptyset_*\&\mathbb{B}_*$ would imply that the negation between $\mathbb{B}_*$ and the second copy of just_*

is a member of the right hand side, but not the left hand side, which is a contradiction. The last case of $\mathbb{B}_*\&\mathbb{B}_*$ also leads to a contradiction as it would restrict members of the right hand side to output tt in response to having having been supplied with arguments tt and ff to the function upon request, while members of the left hand side would also be free to answer ff. □

5. A Category with Families of Context Games

All is not lost, however. In fact, we have almost translated the syntax of dependently typed equational logic into the world of games and strategies. The remaining generalisation, necessitated by the lack of additive Σ-types, is to dependent games depending on multiple (mutually dependent) games. We can produce a categorical model of DTT out of the resulting structure by applying a so-called **category of contexts** (Ctxt) **construction**, which is precisely how one builds a categorical model from the syntax of dependent type theory [19, 25]. This construction can be seen as a way of making our indexed category satisfy the comprehension axiom, extending its base category by (inductively) adjoining (strong) Σ-types formally, analogous to the Fam-construction of [17] which adds formal co-products.

The problem which needs to be addressed is how to interpret dependent types and dependent functions of more variables. This is done through a notion of context game and a generalisation of the Π-game construction from the previous section.

Definition 5.1 (Context Game)**.** *We define a **context game** to be a (finite) list $[X_i]_{1\le i\le n}$ where X_i is a **game with dependency on** $[X_j]_{j<i}$, i.e. a continuous function* $\mathsf{str}(☺(X_1)) \times \cdots \times \mathsf{str}(☺(X_{i-1})) \cong \mathsf{str}(☺(X_1)\& \cdots \& ☺(X_{i-1})) \xrightarrow{X_i} \mathsf{Sub}(☺(X_i))$ *for some game $☺(X_i)$.*

Definition 5.2 (Dependent Π-game)**.** *For a game X_{n+1} depending on $[X_i]_{i\le n}$, we define the game $\Pi_{X_n}X_{n+1}$ depending on $[X_i]_{i\le n-1}$ by $☺(\Pi_{X_n}X_{n+1}) := ☺(X_n) \Rightarrow ☺(X_{n+1})$ from which $(\Pi_{X_n}X_{n+1})(\sigma_1,\dots,\sigma_{n-1})$ is carved out as*

$$
\begin{aligned}
P_{(\Pi_{X_n}X_{n+1})(\sigma_1,\dots,\sigma_{n-1})} = &\{\epsilon\} \bigcup \\
&\{sa \mid s \in P^{\mathsf{even}}_{(\Pi_{X_n}X_{n+1})(\sigma_1,\dots,\sigma_{n-1})} \ \wedge\ \exists_{\overline{sa\restriction_{!☺(X_n)}}\subseteq\tau\in\mathsf{wstr}(X_n(\sigma_1,\dots,\sigma_{n-1}))} sa \in P_{X_n(\sigma_1,\dots,\sigma_{n-1})\Rightarrow X_{n+1}(\sigma_1,\dots,\sigma_{n-1},\tau)} \} \bigcup \\
&\{sab \mid sa \in P^{\mathsf{odd}}_{(\Pi_{X_n}X_{n+1})(\sigma_1,\dots,\sigma_{n-1})} \wedge \forall_{\overline{sab\restriction_{!☺(X_n)}}\subseteq\tau\in\mathsf{wstr}(X_n(\sigma_1,\dots,\sigma_{n-1}))} sa \in P_{X_n(\sigma_1,\dots,\sigma_{n-1})\Rightarrow X_{n+1}(\sigma_1,\dots,\sigma_{n-1},\tau)} \Rightarrow \\
&sab \in P_{X_n(\sigma_1,\dots,\sigma_{n-1})\Rightarrow X_{n+1}(\sigma_1,\dots,\sigma_{n-1},\tau)} \}.
\end{aligned}
$$

The following explicit characterisation of **the game** $\mathsf{O}\text{-}\mathsf{sat}(\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1})$ **of dependent functions of multiple arguments** will be useful later.

Theorem 5.3. *Explicitly, we have that $(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\dots,\sigma_{k-1})$ can be inductively defined as the following subset of the plays of $☺(X_k) \Rightarrow \cdots \Rightarrow ☺(X_{n+1})$:*

$$
\begin{aligned}
P_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\dots,\sigma_{k-1})} = &\{\epsilon\} \bigcup \\
&\{sa \mid s \in P^{\mathsf{even}}_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\dots,\sigma_{k-1})} \ \wedge\ \exists_{\overline{sa\restriction_{!☺(X_k)}}\subseteq\sigma_k\in\mathsf{wstr}(X_k(\sigma_1,\dots,\sigma_{k-1}))} \cdots \exists_{\overline{sa\restriction_{!☺(X_n)}}\subseteq\sigma_n\in\mathsf{wstr}(X_n(\sigma_1,\dots,\sigma_{n-1}))} \\
&sa \in P_{X_k(\sigma_1,\dots,\sigma_{k-1})\Rightarrow\cdots\Rightarrow X_{n+1}(\sigma_1,\dots,\sigma_n)} \} \bigcup \\
&\{sab \mid sa \in P^{\mathsf{odd}}_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\dots,\sigma_{k-1})} \wedge \forall_{\overline{sab\restriction_{!☺(X_k)}}\subseteq\sigma_k\in\mathsf{wstr}(X_k(\sigma_1,\dots,\sigma_{k-1}))} \cdots \forall_{\overline{sab\restriction_{!☺(X_n)}}\subseteq\sigma_n\in\mathsf{wstr}(X_n(\sigma_1,\dots,\sigma_{n-1}))} \\
&sa \in P_{X_k(\sigma_1,\dots,\sigma_{k-1})\Rightarrow\cdots\Rightarrow X_{n+1}(\sigma_1,\dots,\sigma_n)} \Rightarrow sab \in P_{X_k(\sigma_1,\dots,\sigma_{k-1})\Rightarrow\cdots\Rightarrow X_{n+1}(\sigma_1,\dots,\sigma_n)} \}.
\end{aligned}
$$

As a consequence, the game of dependent functions $\mathsf{O}\text{-}\mathsf{sat}(\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1})$ *is carved out in $☺(X_1) \Rightarrow \cdots \Rightarrow ☺(X_n) \Rightarrow ☺(X_{n+1})$ as*

$$\begin{aligned}P_{\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1})} =& \{\epsilon\} \bigcup \\ & \{sa \mid s \in P^{\mathsf{even}}_{\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1})}\} \bigcup \\ & \{sab \mid sa \in P^{\mathsf{odd}}_{\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1})} \wedge \forall_{\overline{sab\restriction_{!\☺(X_1)}}\subseteq\tau_1\in\mathsf{wstr}(X_1())}\cdots\forall_{\overline{sab\restriction_{!\☺(X_n)}}\subseteq\tau_n\in\mathsf{wstr}(X_n(\tau_1,\ldots,\tau_{n-1}))} \\ & sa \in P_{X_1()\Rightarrow\cdots\Rightarrow X_n(\tau_1,\ldots,\tau_{n-1})\Rightarrow X_{n+1}(\tau_1,\ldots,\tau_n)} \Rightarrow sab \in P_{X_1()\Rightarrow\cdots\Rightarrow X_n(\tau_1,\ldots,\tau_{n-1})\Rightarrow X_{n+1}(\tau_1,\ldots,\tau_n)}\}.\end{aligned}$$

That is, the set of plays where Opponent can do whatever she pleases, while Player can only move without further determining the fibre of any of $X_1,\ldots,X_{n+1}$ as long as Opponent plays along compatible history-free strategies $\sigma_1,\ldots,\sigma_n$ on $\☺(X_1),\ldots,\☺(X_n)$, in the sense that they extend to $\langle\tau_1,\ldots,\tau_n\rangle \in \Sigma(\mathsf{wstr}(X_1),\ldots,\mathsf{wstr}(X_n)) := \{\langle\tau_1,\ldots,\tau_n\rangle \mid \tau_1 \in \mathsf{wstr}(X_1()) \wedge \cdots \wedge \tau_n \in \mathsf{wstr}(X_n(\tau_1,\ldots,\tau_{n-1}))\}$ such that the current play obeys the rules of $X_1() \Rightarrow \cdots \Rightarrow X_n(\tau_1,\ldots,\tau_{n-1}) \Rightarrow X_{n+1}(\tau_1,\ldots,\tau_n)$.

Proof. We first note that the second claim follows straightforwardly from the first. Clearly, the proposed description of Opponent moves in the second claim is correct as, by definition of O-sat, Opponent is free to move in $\☺(X_1) \Rightarrow \cdots \Rightarrow \☺(X_n) \Rightarrow \☺(X_{n+1})$ in $\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1})$. For Player moves, note that $\overline{sab\restriction_{!\☺(X_i)}} = \overline{sa\restriction_{!\☺(X_i)}}$ for all $1 \leq i \leq n$. Therefore, assuming the first claim holds, it follows that we are in one of two cases:

- Opponent has been naughty and has broken the rules of $\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1}$. In this case, there are no $\overline{sab\restriction_{!\☺(X_1)}} \subseteq \tau_1 \in \mathsf{wstr}(X_1()),\ldots,\overline{sab\restriction_{!\☺(X_n)}} \subseteq \tau_n \in \mathsf{wstr}(X_n(\tau_1,\ldots,\tau_{n-1}))$ such that $sa \in X_1() \Rightarrow \cdots \Rightarrow X_{n+1}(\tau_1,\ldots,\tau_n)$. In this case, Player is allowed to make any valid move in $\☺(X_1) \Rightarrow \cdots \Rightarrow \☺(X_n) \Rightarrow \☺(X_{n+1})$ according to our proposed description of the second claim as the hypotheses of the implication defining the incremental condition on Player moves are false. This matches, of course, the definition of $\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1})$ from the description of $\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1}$ of the first claim.
- Opponent has been nice and has followed the rules of $\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1}$ (i.e. there are in fact such $\tau_1,\ldots,\tau_n$). In this case, Player has to keep obeying the rules of $\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1}$ as well according to the definition of $\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}X_{n+1})$. This matches our proposed description of the second claim.

For the first claim, the idea is that Opponent has to play precisely such that there is **some** compatible assignment of winning strategies $\sigma_k,\ldots,\sigma_n$ on $X_k,\ldots,X_n$ while Player has to play such that she does not exclude **any** such compatible assignment of winning strategies. Formally, we prove by induction that the proposed description of plays in $(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})$ coincides with its definition

$$\begin{aligned}P_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})} =& \{\epsilon\} \bigcup \\ & \{sa \mid s \in P^{\mathsf{even}}_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})} \wedge \exists_{\overline{sa\restriction_{!\☺(X_k)}}\subseteq\sigma_k\in\mathsf{wstr}(X_k(\sigma_1,\ldots,\sigma_{k-1}))} s\restriction_{!\☺(X_{k+1}),\ldots,\☺(X_{n+1})} \in P_{(\Pi_{X_{k+1}}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_k)} \\ & \wedge \exists_{\overline{sa\restriction_{!\☺(X_{k+1})}}\subseteq\sigma_{k+1}\in\mathsf{wstr}(X_{k+1}(\sigma_1,\ldots,\sigma_k))} s\restriction_{!\☺(X_{k+2}),\ldots,\☺(X_{n+1})} \in P_{(\Pi_{X_{k+2}}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k+1})} \wedge \cdots \\ & \wedge \exists_{\overline{sa\restriction_{!\☺(X_n)}}\subseteq\sigma_n\in\mathsf{wstr}(X_{k+1}(\sigma_1,\ldots,\sigma_{n-1}))} sa\restriction_{\☺(X_{n+1})} \in P_{X_{n+1}(\sigma_1,\ldots,\sigma_n)}\} \bigcup \\ & \{sab \mid sa \in P^{\mathsf{odd}}_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})} \wedge \forall_{\overline{sab\restriction_{!\☺(X_k)}}\subseteq\sigma_k\in\mathsf{wstr}(X_k(\sigma_1,\ldots,\sigma_{k-1}))}(sa \in P_{X_k(\sigma_1,\ldots,\sigma_{k-1})\Rightarrow(\Pi_{X_{k+1}}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_k)} \\ & \Rightarrow sab\restriction_{!\☺(X_k)} \in P_{X_k(\sigma_1,\ldots,\sigma_{k-1})}) \wedge \forall_{\overline{sab\restriction_{!\☺(X_{k+1})}}\subseteq\sigma_{k+1}\in\mathsf{wstr}(X_{k+1}(\sigma_1,\ldots,\sigma_k))}(sa\restriction_{!\☺(X_{k+1}),\ldots,\☺(X_{n+1})} \in P_{X_{k+1}(\sigma_1,\ldots,\sigma_k)\Rightarrow(\Pi_{X_{k+2}}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k+1})} \\ & \Rightarrow sab\restriction_{!\☺(X_{k+1})} \in P_{X_{k+1}(\sigma_1,\ldots,\sigma_k)}) \wedge \cdots \wedge \forall_{\overline{sab\restriction_{!\☺(X_n)}}\subseteq\sigma_n\in\mathsf{wstr}(X_n(\sigma_1,\ldots,\sigma_{n-1}))} \\ & (sa\restriction_{!\☺(X_n),\☺(X_{n+1})} \in P_{(\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{n-1})} \Rightarrow sab\restriction_{!\☺(X_n)} \in P_{X_n(\sigma_1,\ldots,\sigma_{n-1})} \wedge sab\restriction_{\☺(X_{n+1})} \in P_{X_{n+1}(\sigma_1,\ldots,\sigma_n)})\}.\end{aligned}$$

We note that the proposed description is valid for ϵ. Let us suppose it is valid for s. Note that all conjuncts involving s (rather than sa) in the incremental condition on Opponent moves then already follow from the condition that $s \in P^{\mathsf{even}}_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})}$. Rearranging the incremental condition on Opponent moves now gives us a description in which we have obtained the required incremental condition on Opponent moves, but not yet on Player moves - in particular, our proposed description is now valid for sa:

$$
\begin{aligned}
P_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})} = &\{\epsilon\} \bigcup \\
&\{sa \mid s \in P^{\mathsf{even}}_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})} \ \wedge\ \exists_{\overline{sa\restriction_{!\odot(X_k)}}\subseteq\sigma_k\in\mathsf{wstr}(X_k(\sigma_1,\ldots,\sigma_{k-1}))}\cdots\exists_{\overline{sa\restriction_{!\odot(X_n)}}\subseteq\sigma_n\in\mathsf{wstr}(X_n(\sigma_1,\ldots,\sigma_{n-1}))} \\
&sa \in P_{X_k(\sigma_1,\ldots,\sigma_{k-1})\Rightarrow\cdots\Rightarrow X_{n+1}(\sigma_1,\ldots,\sigma_n)}\ \} \bigcup \\
&\{sab \mid sa \in P^{\mathsf{odd}}_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})} \wedge \forall_{\overline{sab\restriction_{!\odot(X_k)}}\subseteq\sigma_k\in\mathsf{wstr}(X_k(\sigma_1,\ldots,\sigma_{k-1}))}(sa \in P_{X_k(\sigma_1,\ldots,\sigma_{k-1})\Rightarrow(\Pi_{X_{k+1}}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_k)} \\
&\Rightarrow\ sab\restriction_{!\odot(X_k)} \in P_{X_k(\sigma_1,\ldots,\sigma_{k-1})})\ \wedge\ \forall_{\overline{sab\restriction_{!\odot(X_{k+1})}}\subseteq\sigma_{k+1}\in\mathsf{wstr}(X_{k+1}(\sigma_1,\ldots,\sigma_k))}(sa\restriction_{!\odot(X_{k+1}),\ldots,\odot(X_{n+1})} \in P_{X_{k+1}(\sigma_1,\ldots,\sigma_k)\Rightarrow(\Pi_{X_{k+2}}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k+1})} \\
&\Rightarrow\ sab\restriction_{!\odot(X_{k+1})} \in P_{X_{k+1}(\sigma_1,\ldots,\sigma_k)})\ \wedge\ \cdots\ \wedge\ \forall_{\overline{sab\restriction_{!\odot(X_n)}}\subseteq\sigma_n\in\mathsf{wstr}(X_n(\sigma_1,\ldots,\sigma_{n-1}))} \\
&(sa\restriction_{!\odot(X_n),\odot(X_{n+1})} \in P_{(\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{n-1})} \Rightarrow sab\restriction_{!\odot(X_n)} \in P_{X_n(\sigma_1,\ldots,\sigma_{n-1})}\ \wedge\ sab\restriction_{\odot(X_{n+1})} \in P_{X_{n+1}(\sigma_1,\ldots,\sigma_n)})\ \}.
\end{aligned}
$$

Next, noting that our proposed description holds for sa, we note that the conjuncts

$$sa\restriction_{!\odot(X_m),\ldots,\odot(X_{n+1})} \in P_{X_m(\sigma_1,\ldots,\sigma_{m-1})\Rightarrow(\Pi_{X_{m+1}}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_m)}$$

in the incremental condition on P-moves can be replaced by the conditions

$$\exists_{\overline{sa\restriction_{!\odot(X_{m+1})}}\subseteq\sigma_{m+1}\in\mathsf{wstr}(X_{m+1}(\sigma_1,\ldots,\sigma_m))}\cdots\exists_{\overline{sa\restriction_{!\odot(X_n)}}\subseteq\sigma_n\in\mathsf{wstr}(X_n(\sigma_1,\ldots,\sigma_{n-1}))} sa\restriction_{!\odot(X_m),\ldots,\odot(X_{n+1})} \in P_{X_m(\sigma_1,\ldots,\sigma_{m-1})\Rightarrow\cdots\Rightarrow X_{n+1}(\sigma_1,\ldots,\sigma_n)}.$$

Now (again noting that $\overline{sa\restriction_{!\odot(X_l)}} = \overline{sab\restriction_{!\odot(X_l)}}$, this means that all universal quantifiers in the incremental condition on P-moves range over a non-empty domain, so we might as well move them to the front of our formula, seeing that they do not bind any more variables:

$$
\begin{aligned}
P_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})} = &\{\epsilon\} \bigcup \\
&\{sa \mid s \in P^{\mathsf{even}}_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})} \ \wedge\ \exists_{\overline{sa\restriction_{!\odot(X_k)}}\subseteq\sigma_k\in\mathsf{wstr}(X_k(\sigma_1,\ldots,\sigma_{k-1}))}\cdots\exists_{\overline{sa\restriction_{!\odot(X_n)}}\subseteq\sigma_n\in\mathsf{wstr}(X_n(\sigma_1,\ldots,\sigma_{n-1}))} \\
&sa \in P_{X_k(\sigma_1,\ldots,\sigma_{k-1})\Rightarrow\cdots\Rightarrow X_{n+1}(\sigma_1,\ldots,\sigma_n)}\ \} \bigcup \\
&\{sab \mid sa \in P^{\mathsf{odd}}_{(\Pi_{X_k}\cdots\Pi_{X_n}X_{n+1})(\sigma_1,\ldots,\sigma_{k-1})} \wedge \forall_{\overline{sab\restriction_{!\odot(X_k)}}\subseteq\sigma_k\in\mathsf{wstr}(X_k(\sigma_1,\ldots,\sigma_{k-1}))}\cdots\forall_{\overline{sab\restriction_{!\odot(X_n)}}\subseteq\sigma_n\in\mathsf{wstr}(X_n(\sigma_1,\ldots,\sigma_{n-1}))} \\
&sa \in P_{X_k(\sigma_1,\ldots,\sigma_{k-1})\Rightarrow\cdots\Rightarrow X_{n+1}(\sigma_1,\ldots,\sigma_n)} \Rightarrow sab\restriction_{!\odot(X_k)} \in P_{X_k(\sigma_1,\ldots,\sigma_{k-1})}\ \wedge\ \cdots\ \wedge \\
&sab\restriction_{!\odot(X_n)} \in P_{X_n(\sigma_1,\ldots,\sigma_{n-1})}\ \wedge\ sab\restriction_{\odot(X_{n+1})} \in P_{X_{n+1}(\sigma_1,\ldots,\sigma_n)}\ \},
\end{aligned}
$$

which is clearly carves out the same plays in $P_{\odot(X_1)\Rightarrow\cdots\Rightarrow\odot(X_n)\Rightarrow\odot(X_{n+1})}$ as our proposed description. □

Remark 5.4 (Logical Predicates/Realizability?)**.** *Note that these games of dependent functions lead to quite a non-trivial notion of dependently typed strategy. Indeed, we can send a game B with dependency on A to a function* $\mathsf{str}(\odot(A)) \longrightarrow \mathcal{P}(\mathsf{str}(\odot(B)))$ *which assigns a set of consistent strategies* $\sigma \mapsto \mathsf{cstr}(B(\sigma)) := \mathsf{str}(\mathsf{O\text{-}sat}(B(\sigma))) \subseteq \mathsf{str}(\odot(B))$*. One might wonder if this description is enough to recover our model from and if the model can be recast into a realizability style model [31]. In particular, this would mean that we send a pair* $(A(\bot), \odot(A))$ *to the pair* $(\odot(A), \mathsf{cstr}(A) \subseteq \mathsf{str}(\odot(A)))$*. In a realizability model, one would expect this class* $\mathsf{cstr}(A)$ *of consistent strategies to behave as a logical predicate. In particular, given* $\mathsf{just}_*(\mathsf{tt}) = (\{\mathsf{tt}\}_*, \mathbb{B}_*)$*, if* cstr *were a logical predicate, we would have that* $\mathsf{cstr}((\mathsf{just}_*(\mathsf{tt}) \Rightarrow \mathbb{B}_*) \Rightarrow \mathbb{B}_*) = \mathsf{cstr}((\mathbb{B}_* \Rightarrow \mathbb{B}_*) \Rightarrow \mathbb{B}_*) = \mathsf{str}((\mathbb{B}_* \Rightarrow \mathbb{B}_*) \Rightarrow \mathbb{B}_*)$ *as* $\mathsf{cstr}(\mathbb{B}_*) = \mathsf{str}(\mathbb{B}_*)$*. However, we have that* $\mathsf{cstr}((\mathsf{just}_*(\mathsf{tt}) \Rightarrow \mathbb{B}_*) \Rightarrow \mathbb{B}_*) := \mathsf{str}(\mathsf{O\text{-}sat}((\mathsf{just}_*(\mathsf{tt}) \Rightarrow \mathbb{B}_*) \Rightarrow \mathbb{B}_*)) \subsetneq \mathsf{str}((\mathbb{B}_* \Rightarrow \mathbb{B}_*) \Rightarrow \mathbb{B}_*)$*. Indeed, a consistent strategy on* $(\mathsf{just}_*(\mathsf{tt}) \Rightarrow \mathbb{B}_*) \Rightarrow \mathbb{B}_*$ *cannot play* ff *in* $\mathsf{just}_*(\mathsf{tt}) \trianglelefteq \mathbb{B}_*$*. We see that our notion of consistent strategy does not behave as a logical predicate. We get a more non-trivial notion of higher order dependent function.*

For illustration, define a game RA_* depending on the context game $[\mathbb{N}_*, \mathsf{days}_*]$ by

$$
\begin{aligned}
&\mathsf{RA}(n, m) := \{\text{Rick Astley lyrics from songs released before day } m \text{ of year } n\} \\
&\mathsf{RA}(n, \bot) := \mathsf{RA}(\bot, m) := \mathsf{RA}(\bot, \bot) := \emptyset.
\end{aligned}
$$

Then, the following two strategies illustrate that a dependent function may query its arguments in unexpected order or may not query some at all.

$!\mathbb{N}_*$	$!\mathsf{days}_*$	RA_*		$!\mathbb{N}_*$	$!\mathsf{days}_*$	RA_*	
		$*$	O			$*$	O
	$(i, *)$		P	$(i, *)$			P
	$(i, m > 206)$		O	$(i, n > 1987)$			O
$(j, *)$			P			Never Gonna Let You Down	P
$(j, 1987)$			O				O
		Never Gonna Give You Up	P				P

Figure 12. Two examples of (partial) strategies on the game $\mathsf{O\text{-}sat}(\Pi_{\mathbb{N}_*}\Pi_{\mathsf{days}_*}\mathsf{RA}_*)$, defining dependent functions of two arguments. Note that these lyrics come from a song released on day 207 of the year 1987, so Player does not limit the fibre anywhere.

To illustrate the subtle nature of higher order dependent functions with an example, define the game $\mathsf{holidays}_*$ depending on the context game $[\mathbb{N}_*, \mathsf{days}_*]$ by

$$\mathsf{holidays}(n, m) := \{\text{holidays that are celebrated on day } m \text{ of year } n\}$$
$$\mathsf{holidays}(n, \bot) := \mathsf{holidays}(\bot, m) := \mathsf{holidays}(\bot, \bot) := \emptyset.$$

The following figure illustrates how Player is in charge of providing certain arguments (the positive ones) of dependent games and can therefore choose the fibre in some cases. (Opponent controls the negative arguments to dependent games.) In the figure below, Player controls the arguments of type days_* and $\mathsf{holidays}_*$, while Opponent is in charge of the type of years $\mathbb{N}_*$. We stress again that although Player has to play in accordance with any choice of year that Opponent could make, the converse is not true: Opponent can do what she likes and does not have to respect Player's choices of day and holiday.

$!\mathbb{N}_*$	$!!\mathsf{days}_*$	$!!\mathsf{holidays}_*$	$!\mathbb{B}_*$	$\mathbb{B}_*$		$!\mathbb{N}_*$	$!!\mathsf{days}_*$	$!!\mathsf{holidays}_*$	$!\mathbb{B}_*$	$\mathbb{B}_*$	
				$*$	O					$*$	O
			$(0, *)$		P				$(0, *)$		P
		$(0, (0, *))$			O			$(0, (0, *))$			O
		$(0, (0,$ International Talk Like a Pirate Day$))$			P			$(0, (0,$ Holi$))$			P
			$(0,$ ff$)$		O		$(0, (0, *))$				O
				tt	P	$(0, *)$					P
					O	$(0, 2015)$					O
					P		$(0, (0, 65))$				P
					O				$(0,$ tt$)$		O
					P					tt	P

Figure 13. Two plays in $\mathsf{O\text{-}sat}(\Pi_{\mathbb{N}_*}\Pi_{\Pi_{\mathsf{days}_*}\Pi_{\mathsf{holidays}_*}\mathbb{B}_*}\mathbb{B}_*)$. For an interpretation, imagine Player is a PhD-student who is trying to decide if he is going on holidays and ends up asking his supervisor (Opponent) if she's okay with him doing so. The first play can be read as the dialogue where the supervisor asks if the student is planning to take any holidays, the student asks if he's allowed to, the supervisor wants to know what the occasion is, the student admits that his best excuse for wanting time off is International Talk Like a Pirate Day, the supervisor tells the student that he can't have time off and, finally, the student tells his supervisor that he's taking time off anyway for this important occasion. Here, Player can choose the holiday 'International Talk Like a Pirate Day' as it is celebrated each year, meaning that Player does not restrict the year we may be talking about (which, as a negative argument, belongs to Opponent). Note that by choosing this particular holiday, Player automatically fixes the day the holiday falls on, which is fine as the subgame days_* occurs positively in the total game we are playing in, meaning that Player is in charge of determining the corresponding argument. The second play corresponds to a dialogue with a more sensible student who uses the more respectable excuse of celebrating Holi to get time off from work. Here, Player has to let Opponent determine the year first, before she can answer with a date for Holi, as the date of Holi on the Gregorian calendar varies (while it is celebrated every year).

We define a category $\mathsf{Ctxt}(\mathbf{DGame}_!)$ with context games as objects and morphisms which are defined inductively as (dependent) lists of winning strategies on appropriate games of dependent functions. We show that this has the structure of a category with families (CwF) [32, 19], a canonical notion of model of dependently typed equational logic. This gives a more concise presentation of the strict indexed category with comprehension which results from $\mathbf{DGame}_!$ when we add formal Σ-types to both the fibres and the base category.

Definition 5.5 (CwF)**.** *A CwF is a category $\mathcal{C}$ with a terminal object $\cdot$, for all objects Γ a set $\mathsf{Ty}(\Gamma)$, for all $A \in \mathsf{Ty}(\Gamma)$ a set $\mathsf{Tm}(\Gamma, A)$, for all $\Gamma' \xrightarrow{f} \Gamma$ in $\mathcal{C}$ functions $\mathsf{Ty}(\Gamma) \xrightarrow{-\{f\}} \mathsf{Ty}(\Gamma')$ and $\mathsf{Tm}(\Gamma, A) \xrightarrow{-\{f\}} \mathsf{Tm}(\Gamma', A\{f\})$, such that*

$$\begin{array}{llll} A\{\mathsf{id}_\Gamma\} = A & \text{(Ty-Id)} & A\{f;g\} = A\{g\}\{f\} & \text{(Ty-Comp)} \\ t\{\mathsf{id}_\Gamma\} = t & \text{(Tm-Id)} & t\{f;g\} = t\{g\}\{f\} & \text{(Tm-Comp),} \end{array}$$

for $A \in \mathsf{Ty}(\Gamma)$ a morphism $\Gamma.A \xrightarrow{\mathbf{p}_{\Gamma,A}} \Gamma$ of $\mathcal{C}$ and $\mathbf{v}_{\Gamma,A} \in \mathsf{Tm}(\Gamma.A, A\{\mathbf{p}_{\Gamma,A}\})$ and, finally, for all $t \in \mathsf{Tm}(\Gamma', A\{f\})$ a morphism $\Gamma' \xrightarrow{\langle f,t\rangle} \Gamma.A$ such that

$$\begin{array}{llll} \langle f,t\rangle;\mathbf{p}_{\Gamma,A} = f & \text{(Cons-L)} & \mathbf{v}_{\Gamma,A}\{\langle f,t\rangle\} = t & \text{(Cons-R)} \\ \langle \mathbf{p}_{\Gamma,A}, \mathbf{v}_{\Gamma,A}\rangle = \mathsf{id}_{\Gamma.A} & \text{(Cons-Id)} & g;\langle f,t\rangle = \langle g; f, t\{g\}\rangle & \text{(Cons-Nat).} \end{array}$$

Theorem 5.6. *We have a category with families* $(\mathsf{Ctxt}(\mathbf{DGame}_!), \mathsf{Ty}, \mathsf{Tm}, \mathbf{p}, \mathbf{v}, -.-, \langle -,-\rangle)$.

Proof. We define the required structures and verify the required equations.

$\boxed{\mathsf{ob}(\mathcal{C}),\ \mathsf{Ty},\ -.-,\ \cdot}$

We define a category $\mathcal{C} := \mathsf{Ctxt}(\mathbf{DGame}_!)$ with context games as objects. We define $\mathsf{Ty}([X_i]_i)$ as the set of **context games with dependency on** $[X_i]_i$: $[Y_j]_j \in \mathsf{Ty}([X_i]_i)$ iff $[X_i]_i.[Y_j]_j := [X_1, \dots, X_n, Y_1, \dots, Y_m]$ is a context game, while $\cdot := []$ is the terminal object.

$\boxed{\mathsf{mor}(\mathcal{C}),\ -\{-\}_{\mathsf{Ty}}}$

Next, $\mathsf{mor}(\mathcal{C})$ and $-\{-\}_{\mathsf{Ty}}$ are defined (where $[X_i]_{i\le n}, [Y_j]_{j\le m} \in \mathsf{ob}(\mathcal{C})$ and $[Z_k]_{k\le l} \in \mathsf{Ty}([Y_j]_{j\le m})$):

$$\mathsf{Ctxt}(\mathbf{DGame}_!)([X_i]_{i\le n}, [Y_j]_{j\le m}) := \left\{ [f_j]_{j\le m} \mid f_j \in \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{X_1} \dots \Pi_{X_n} Y_j\{[f_k]_{k<j}\})) \right\}$$

$$Z_k\{[f_j]_{j<m}\}(\sigma_1, \dots, \sigma_n, \tau_1, \dots, \tau_{k-1}) := Z_k(\langle\sigma_1, \dots, \sigma_n\rangle^\dagger; f_1, \dots, \langle\sigma_1, \dots, \sigma_n\rangle^\dagger; f_m, \tau_1, \dots, \tau_{k-1}), \quad \☺(Z_k\{[f_j]_{j<m}\}) = \☺(Z_k).$$

Here, $\langle\sigma_1, \dots, \sigma_n\rangle^\dagger; f_j$ is defined as the usual co-Kleisli composition of (winning) strategies on $\☺(X_1)\& \cdots \&\☺(X_n)$ and $\☺(X_1) \Rightarrow \cdots \Rightarrow \☺(X_n) \Rightarrow \☺(Y_j)$. Note that $-\{-\}_{\mathsf{Ty}}$ is well-defined, as post-composition defines a continuous function on domains of strategies.

$\boxed{\mathsf{id},\ \mathbf{p},\ \mathsf{Tm},\ \mathbf{v},\ \langle\cdot,\cdot\rangle,\ \text{(Cons-Id)}}$

The identities are defined as lists of derelicted copycats. Let us define a strategy $\mathsf{der}_{[X_j]_j, X_i}$ which plays the derelicted copycat on all of $\☺(X_i)$: $\mathsf{der}_{[X_j]_j,X_i} := \{s \in P_{\mathsf{O\text{-}sat}(\Pi_{X_1}\dots\Pi_{X_n}X_i)} \mid \forall_{s' \in P^{\mathsf{even}}_{\mathsf{O\text{-}sat}(\Pi_{X_1}\dots\Pi_{X_n}X_i)}} s' \le s \Rightarrow \exists_k s \restriction_{!\☺(X_i)} \restriction_k \approx_{\☺(X_i)} s \restriction_{\☺(X_i)}\}$. We then define $\mathsf{id}_{[X_i]_i} := [\mathsf{der}_{[X_j]_j,X_i}]_i$ and $\mathbf{p}_{[X_i]_i,[Y_j]_j} := [\mathsf{der}_{[X_i]_i.[Y_j]_j,X_k}]_k$. Let us define

$$\mathsf{Tm}([X_i]_{i\le n}, [Y_j]_{j\le m}) := \left\{ [f_j]_{j\le m} \;\middle|\; [\mathsf{der}_{[X_i]_i,X_1}, \dots, \mathsf{der}_{[X_i]_i,X_n}, f_1, \dots, f_m] \in \mathsf{Ctxt}(\mathbf{DGame}_!)([X_i]_i, [X_i]_i.[Y_j]_j) \right\}.$$

Then, we can define $\mathbf{v}_{[X_i]_i,[Y_j]_j} := [\mathsf{der}_{[X_i]_i.[Y_j]_j,Y_k}]_k$. Note that these are well-defined because of the following claim.

Claim. $\mathsf{der}_{[X_j]_j,X_i}$ *defines a winning strategy on* $\mathsf{O\text{-}sat}(\Pi_{X_1} \cdots \Pi_{X_n} X_i\{[\mathsf{der}_{[X_j]_j,X_k}]_{k\le i-1}\})$.

Proof. Note that Opponent makes every move first, so Player can copy it without restricting the fibre of X_i further. □

We define $\langle[f_j]_{j\le m}, [g_k]_{k\le l}\rangle := [f_1, \dots, f_m, g_1, \dots, g_l]$, after which (Cons-Id) follows trivially.

Composition, $-\{-\}_{\mathsf{Tm}}$, (Cons-Nat)

We define the composition of $[X_i]_{i\leq n} \xrightarrow{[f_j]_j} [Y_j]_{j\leq m} \xrightarrow{[g_k]_k} [Z_k]_k$ in $\mathsf{Ctxt}(\mathbf{DGame}_!)$ by

$$[f_j]_j;[g_k]_k := [\langle f_1,\dots,f_m\rangle^\dagger; g_k]_k,$$

using the usual (co-Kleisli) composition of (winning) strategies on $\mathbb{O}(X_1)\Rightarrow\cdots\Rightarrow\mathbb{O}(X_n)\Rightarrow(\mathbb{O}(Y_1)\&\cdots\&\mathbb{O}(Y_m))$ and $\mathbb{O}(Y_1)\Rightarrow\cdots\Rightarrow\mathbb{O}(Y_m)\Rightarrow\mathbb{O}(Z_k)$. We note that we can assign to this composition a more precise dependent function type.

Claim. *The composition* $[f_j]_j;[g_k]_k$ *above defines a morphism in* $\mathsf{Ctxt}(\mathbf{DGame}_!)([X_i]_{i\leq n},[Z_k]_k)$.

Proof. We need to verify that $\langle f_1,\dots,f_m\rangle^\dagger;g_k$ defines a winning strategy on $\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}W\{[f_j]_j\})$, where we write $W := Z_k\{[g_{k'}]_{k'<k}\}$. The winning part of the claim follows trivially from the usual fact that winning strategies compose. What is to be verified is the claim that $\langle f_1,\dots,f_m\rangle^\dagger;g_k$ is a strategy on $\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}W\{[f_j]_j\})$.

Recall that, by assumption, g_k is a strategy on $\mathsf{O\text{-}sat}(\Pi_{Y_1}\cdots\Pi_{Y_m}W)$. Suppose $\langle f_1,\dots,f_m\rangle^\dagger;g_k$ wants to respond with a move b in some X_i after a play sa. Recall that by theorem 5.3, we need to verify that for all $\overline{sab\restriction_{!\mathbb{O}(X_1)}}\subseteq\sigma'_1\in\mathsf{wstr}(X_1()),\dots,\overline{sab\restriction_{!\mathbb{O}(X_n)}}\subseteq\sigma'_n\in\mathsf{wstr}(X_n(\sigma'_1,\dots,\sigma'_{n-1}))$, we have that

$$sab\restriction_{!\mathbb{O}(X_1)}\in P_{!X_1()}\wedge\cdots\wedge sab\restriction_{!\mathbb{O}(X_n)}\in P_{!X_n(\sigma'_1,\dots,\sigma'_{n-1})}\wedge sab\restriction_{\mathbb{O}(W)}\in P_{W\{[f_j]_j\}(\sigma'_1,\dots,\sigma'_n)},$$

provided that already

$$sa\restriction_{!\mathbb{O}(X_1)}\in P_{!X_1()}\wedge\cdots\wedge sa\restriction_{!\mathbb{O}(X_n)}\in P_{!X_n(\sigma'_1,\dots,\sigma'_{n-1})}\wedge sa\restriction_{\mathbb{O}(W)}\in P_{W\{[f_j]_j\}(\sigma'_1,\dots,\sigma'_n)}.$$

Here, all but the last conjunct follow from the fact that the f_k are strategies on $\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}Y_k\{[f_{k'}]_{k'<k}\})$.

What remains to be checked, therefore, is that

$$sab\restriction_{\mathbb{O}(W)}\in P_{W\{[f_j]_j\}(\sigma'_1,\dots,\sigma'_n)},$$

or, equivalently, by definition of $\{-\}_{\mathsf{Ty}}$ and composition,

$$sab\restriction_{\mathbb{O}(W)}\in P_{W([\sigma_i]_{i\leq n};[f_j]_{j\leq m})}.$$

This follows immediately from the fact that g_k is a strategy on $\mathsf{O\text{-}sat}(\Pi_{Y_1}\cdots\Pi_{Y_m}W)$ if we can show that $[\sigma'_i]_{i\leq n};[f_j]_{j\leq m}\in\mathsf{Ctxt}(\mathbf{DGame}_!)([],[Y_j]_{j\leq m})$, which is a special case of our claim when $[X_i]_i=[]$.

That is, we need to show that $\langle\sigma'_1,\dots,\sigma'_n\rangle^\dagger;f_j$ defines a strategy on $\mathsf{O\text{-}sat}(Y_j\{[\sigma'_i]_{i\leq n};[f_j]_j\})$. (Again, it follows trivially that it will be a winning strategy.) We verify that for any play sab in $\langle\sigma'_1,\dots,\sigma'_n\rangle^\dagger\|f_j$, where b is a Player move in $\mathbb{O}(Y_j)$, we have in fact that it is a move in $\mathsf{O\text{-}sat}(Y_j\{[\sigma'_i]_{i\leq n};[f_j]_j\})$. This follows from the fact that f_j is a strategy on $\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}Y_j\{[f_{j'}]_{j'<j}\})$, which according to theorem 5.3 means, in particular, that for all $\overline{sab\restriction_{!\mathbb{O}(X_1)}}\subseteq\sigma'_1\in\mathsf{wstr}(X_1()),\dots\overline{sab\restriction_{!\mathbb{O}(X_n)}}\subseteq\sigma'_n\in\mathsf{wstr}(X_n(\sigma'_1,\dots,\sigma'_{n-1}))$, we have that

$$sab\restriction_{!\mathbb{O}(X_1)}\in P_{!X_1()}\wedge\cdots\wedge sab\restriction_{!\mathbb{O}(X_n)}\in P_{!X_n(\sigma'_1,\dots,\sigma'_{n-1})}\wedge sab\restriction_{\mathbb{O}(Y_j)}\in P_{Y_j\{[f_{j'}]_{j'<j}\}(\sigma'_1,\dots,\sigma'_n)},$$

provided that already

$$sa\restriction_{!\mathbb{O}(X_1)}\in P_{!X_1()}\wedge\cdots\wedge sa\restriction_{!\mathbb{O}(X_n)}\in P_{!X_n(\sigma'_1,\dots,\sigma'_{n-1})}\wedge sa\restriction_{\mathbb{O}(Y_j)}\in P_{Y_j\{[f_{j'}]_{j'<j}\}(\sigma'_1,\dots,\sigma'_n)}.$$

The last conjunct is what we are looking for, or rather its reformulation $sab\restriction_{\mathbb{O}(Y_j)}\in P_{Y_j\{[\sigma'_i];[f_j]_j\}}$.

□

Note that for $[X_i]_i\xrightarrow{[f_j]_j}[Y_j]_j$ and $[Y_j]_j\xrightarrow{\langle[g_k]_k,[h_l]_l\rangle}[Z_k]_k.[W_l]_l$, (Cons-Nat) holds in the sense that

$$[f_j]_j;\langle[[g_k]_k,[h_l]_l\rangle=\langle[f_j]_j;[g_k]_k,[h_l]_l\{[f_j]_j\}\rangle,$$

if we define $-\{-\}_{\mathsf{Tm}}$ by

$$[h_l]_l\{[f_j]_j\} := [\langle f_1, \dots, f_m\rangle^\dagger; h_l]_l,$$

which then automatically type checks because of (Cons-Nat).

Identity Law, Associativity, (Ty-Id), (Tm-Id), (Ty-Comp), (Tm-Comp), (Cons-L), (Cons-R)

All these identities are direct consequences of the identity and associativity laws of the usual composition of strategies in $\mathbf{Game}_!$. □

Remark 5.7. *Note that, in* $\mathsf{Ctxt}(\mathbf{DGame}_!)$, $[A, B] \cong [A\&B]$ *if A and B are games (without mutual dependency) and* $[] \cong [I]$.

6. Semantic Type Formers 1, Σ, Π and Id

We show that our CwF supports 1-, Σ-, Π-, and Id-types. We leave the verification of all term equations (which are inherited from their simply typed equivalents) to section 8. We can note that all type formers are preserved by substitution. We characterise some of the properties of the Id-types, marking their place in the intensionality spectrum.

1-*types.* 1-Types are interpreted by the context game of length 0. $\langle\rangle$ is interpreted by the list of strategies of length 0.

Σ-*Types.* Σ-types are just interpreted by concatenation of lists. For $[Z_k]_{k\le l} \in \mathsf{Ty}([X_i]_{i\le n}.[Y_j]_{j\le m})$, we define a Σ-type

$$\Sigma_{[Y_j]_j}[Z_k]_k := [Y_j]_j.[Z_k]_k \in \mathsf{Ty}([X_i]_{i\le n}).$$

We can interpret $\langle -, -\rangle$ by a concatenation $[\sigma_1, \dots, \sigma_m, \tau_1, \dots, \tau_l]$ of lists of strategies $[\sigma_j]_j$ and $[\tau_k]_k$, while we interpret fst as $[\mathsf{der}_{[X_i]_i.[Y_j]_j.[Z_k]_k, Y_{j'}}]_{j'}$ and snd as $[\mathsf{der}_{[X_i]_i.[Y_j]_j.[Z_k]_k, Z_{k'}}]_{k'}$.

Π-*Types.* We have already seen Π-types

$$\Pi_{[Y_j]_{j\le m}}[Z] := [\Pi_{Y_1} \cdots \Pi_{Y_m} Z] \in \mathsf{Ty}([X_i]_{i\le n})$$

of dependent games $[Z] \in \mathsf{Ty}([X_i]_{i\le n}.[Y_j]_{j\le m})$. Indeed, then we can define λ-abstraction and evaluation as on the usual simply typed function game $☺(Y_1) \Rightarrow \cdots \Rightarrow ☺(Y_m) \Rightarrow ☺(Z)$. What remains to be defined are Π-types $\Pi_{[Y_j]_j}[Z_k]_k$ of general dependent context games $[Z_k]_k \in \mathsf{Ty}([X_i]_{i\le n}.[Y_j]_{j\le m})$. These can be reduced to the former, as $\Sigma_{f:\Pi_{x:A}B}\Pi_{x:A}C[f(x)/y]$ satisfies the rules for $\Pi_{x:A}\Sigma_{y:B}C$. Conclusion: our CwF supports Π-types.

Corollary 6.1. *Note that this means that* $\mathsf{Ctxt}(\mathbf{DGame}_!)$ *is in particular a ccc.*

Id-*Types.* We turn to identity types next, which are essentially defined as those of the domain semantics of DTT [21]. Interestingly, due to the more intensional nature of function types in game semantics, these identity types acquire a more intensional character as well, refuting FunExt.

For $[Y_j]_j \in \mathsf{Ty}([X_i]_i)$, define $\mathsf{Id}_{[Y_j]_j} \in \mathsf{Ty}([X_i]_i.[Y_j]_j.[Y_{j'}]_{j'})$ through the intersection of subgames of $☺(Y_j)$ for $1 \le j \le m$, where we identify a strategy σ on X with the subgame $\sigma \cup \{sa \in P_X \mid s \in \sigma\} \trianglelefteq X$:

$$\mathsf{Id}_{[Y_j]_j}([\sigma_i]_i, [\tau_j]_j, [\tau'_j]_j) := [\mathsf{Id}_{Y_j}]_j([\sigma_i]_i, [\tau_j]_j, [\tau'_j]_j) := [\tau_j \cap \tau'_j]_j.$$

Here, $☺(\mathsf{Id}_{Y_j}) := ☺(Y_j)$. To be precise, really, Id_{Y_j} also takes arguments of types $[\mathsf{Id}_{Y_k}]_{k<j}$ which it ignores. Note that, by definition, the plays of $\mathsf{Id}_{Y_j}([\sigma_i]_i, [\tau_j]_j, [\tau'_j]_j)$ are closed under all Opponent moves in $☺(Y_j) = ☺(\mathsf{Id}_{Y_j})$. Note that this means that $\mathsf{O\text{-}sat}(\mathsf{Id}_{Y_j}([\sigma_i]_i, [\tau_j]_j, [\tau'_j]_j)) = \mathsf{Id}_{Y_j}([\sigma_i]_i, [\tau_j]_j, [\tau'_j]_j)$.

Id-I is interpreted by

$$\mathsf{refl}_{[f_j]_j} := [f_j]_j \in \mathsf{Tm}([X_i]_i, \mathsf{Id}_{[Y_j]_j}\{\langle[\mathsf{der}_{X_i}]_i, [f_j]_j, [f_j]_j\rangle\}) = \{[g_j]_j \mid g_j \in \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{[X_i]_i} f_j\{[g_k]_{k<j}\}))\},$$

where we interpret the strategy $f_j\{[g_k]_{k<j}\}$ as a game depending on $[X_i]_i$, noting that for any $[\sigma_i]_i \in \mathsf{str}(☺(X_1)) \times \cdots \times \mathsf{str}(☺(X_n))$ we have that $f_j\{[g_k]_{k<j}\}\{[\sigma_i]_i\}$ defines a strategy on $☺(Y_j)$ and hence a subgame of $☺(Y_j)$.

For the (strong) Id-E rule, suppose we are given

- $[Z_k]_k \in \mathsf{Ty}([X_i]_i.[Y_j]_j.[Y_j]_j.\mathsf{Id}_{[Y_j]_j})$;
- $[f_k]_k \in \mathsf{Tm}([X_i]_i.[Y_j]_j, [Z_k]_k\{\langle \mathsf{der}_{[X_i]_i}, \mathsf{der}_{[Y_j]_j}, \mathsf{der}_{[Y_j]_j}, \mathsf{refl}_{\mathsf{der}_{[Y_j]_j}}\rangle\})$.

Then, we produce

$$[f'_k]_k \in \mathsf{Tm}([X_i]_i.[Y_j^{(1)}]_j.[Y_j^{(2)}]_j.\mathsf{Id}_{[Y_j]_j}, [Z_k]_k).$$

Here, f'_k is the strategy f_k where we identify the input type Y_j of f_k with the input type Id_{Y_j} of f'_k. (Hence, $[f'_k]_k$ does not ever visit $[Y_j^{(1)}]_j$ or $[Y_j^{(2)}]_j$.) Note that such f'_k are well-defined strategies, as

- because winning strategies are maximal, we have a bijection

$$\mathsf{Ctxt}(\mathbf{DGame}_!)([], [X_i]_i.[Y_j]_j) \;\cong\; \mathsf{Ctxt}(\mathbf{DGame}_!)([], [X_i]_i.[Y_j]_j.[Y_j]_j.\mathsf{Id}_{[Y_j]_j})$$

$$\langle [\sigma_i]_i, [\tau_j]_j\rangle \longmapsto \langle [\sigma_i]_i, [\tau_j]_j, [\tau_j]_j, [\tau_j]_j\rangle,$$

  allowing us to make all necessary P-moves in Z_k;

- Id-types contain all O-moves, allowing f'_k to make all necessary P-moves in $!\mathsf{Id}_{Y_j}$.

Observing that $\circledcirc(\mathsf{Id}_{Y_j}) = \circledcirc(Y_j)$, it follows that f'_k are winning strategies, as f_k are.

Remark 6.2 (Interpretation of subst)**.** *Note that* $\Gamma, x : A, x' : A, p : \mathsf{Id}_A(x, x') \vdash \mathsf{subst}(p, -) : B \Rightarrow B[x'/x]$ *gets interpreted as a simple copycat between the two copies of* $[\![B]\!]$*, which is a well-defined strategy again because the only morphisms* $\mathsf{Ctxt}([], [[\![A]\!], [\![A]\!], [\![\mathsf{Id}_A]\!]])$ *are of the form* $\langle x, x, x\rangle$ *with* $x \in \mathsf{wstr}([\![A]\!])$.

In addition to being non-extensional (i.e. refuting the principle of equality reflection), these identity types can be said to be intensional in a positive sense.

Theorem 6.3. *Streicher's Criteria of Intensionality [18] are satisfied, i.e.*

(I1) there exist $\vdash A\ \mathsf{type}$ *such that* $x, y : A, z : \mathsf{Id}_A(x, y) \nvdash x \equiv y : A$*;*
(I2) there exist $\vdash A\ \mathsf{type}$ *and* $x : A \vdash B\ \mathsf{type}$ *such that* $x, y : A, z : \mathsf{Id}_A(x, y) \nvdash B \equiv B[y/x]\ \mathsf{type}$*;*
(I3) for all $\vdash A\ \mathsf{type}$*,* $\vdash p : \mathsf{Id}_A(t, s)$ *implies* $\vdash t \equiv s : A$.

Proof. (I1) Let us write $\mathbf{p}_{[\mathbb{B}_*{}^{(i)}]}$ for $\mathbf{p}_{[\mathbb{B}_*{}^{(1)}, \mathbb{B}_*{}^{(2)}, \mathsf{Id}_{\mathbb{B}_*}], [\mathbb{B}_*{}^{(i)}]}$ and $[\![-]\!]$ for the interpretation functor from the syntax of $\mathsf{DTT}-$ into $\mathsf{Ctxt}(\mathbf{DGame}_!)$. (I1) relies on the interpretation of terms carrying intensionality. Take $[\![A]\!] := [\mathbb{B}_*]$. Then, we have to show that $\mathbf{p}_{[\mathbb{B}_*{}^{(1)}]} \neq \mathbf{p}_{[\mathbb{B}_*{}^{(2)}]} \in \mathsf{Tm}([\mathbb{B}_*].[\mathbb{B}_*].\mathsf{Id}_{[\mathbb{B}_*]}, [\mathbb{B}_*])$. We note that $\mathbf{p}_{[\mathbb{B}_*{}^{(1)}]}\{\langle[\bot], [\mathrm{tt}], [\bot]\rangle\} = [\bot]$ while $\mathbf{p}_{[\mathbb{B}_*{}^{(2)}]}\{\langle[\bot], [\mathrm{tt}], [\bot]\rangle\} = [\mathrm{tt}]$, which shows that $(I1)$ holds.

(I2) (I2) relies on semantic types having intensional features. Take $[\![A]\!] := [\mathbb{B}_*]$ and $[\![B]\!] := (\bot, \mathrm{ff} \mapsto [I], \mathrm{tt} \mapsto [\mathbb{B}_*])$. Then, we have to show that $[\![B]\!]\{\mathbf{p}_{[\mathbb{B}_*{}^{(1)}]}\} \neq [\![B]\!]\{\mathbf{p}_{[\mathbb{B}_*{}^{(2)}]}\} \in \mathsf{Ty}([\mathbb{B}_*].[\mathbb{B}_*].\mathsf{Id}_{[\mathbb{B}_*]})$. Now, $[\![B]\!]\{\mathbf{p}_{[\mathbb{B}_*{}^{(1)}]}\}\{\langle[\bot], [\mathrm{tt}], [\bot]\rangle\} = [\![B]\!]\{\bot\} = [I]$ while $[\![B]\!]\{\mathbf{p}_{[\mathbb{B}_*{}^{(2)}]}\}\{\langle[\bot], [\mathrm{tt}], [\bot]\rangle\} = [\![B]\!]\{[\mathrm{tt}]\} = [\mathbb{B}_*]$, so we conclude that $(I2)$ holds.

(I3) Given $[\sigma_i]_i, [\tau_i]_i \in \mathsf{Tm}([], [X_i]_i)$ and $[p_i]_i \in \mathsf{Tm}([], \mathsf{Id}_{[X_i]_i}([\sigma_i]_i, [\tau_i]_i)) := \{[q_i]_i \mid q_i \in \mathsf{wstr}(\mathsf{O\text{-}sat}(\sigma_i\{[q_j]_{j<i}\} \cap \tau_i\{[q_j]_{j<i}\}))\} = \{[q_i]_i \mid q_i \in \mathsf{wstr}(\sigma_i\{[q_j]_{j<i}\} \cap \tau_i\{[q_j]_{j<i}\})\}$, we have that $[p_i]_i \leq [\sigma_i \cap \tau_i]_i \leq [\sigma_i]_i, [\tau_i]_i$. Now, $[p_i]_i$, $[\sigma_i]_i$ and $[\tau_i]_i$ all consist of winning hence maximal strategies and therefore coincide.

□

The proofs of (I1) and (I2) also work for the domain model of DTT. (I3) relies on a crucial difference between the domain and games models: winning strategies are maximal, while to account for function types of domains with totality, we cannot assume that total domain elements are maximal. For similar reasons, FunExt is seen to fail in the games model.

Let us compare the situations in the interpretation of DTT. While the identity types in the domain model simply measure observational equivalence (with respect to contexts which take value in ground types), in the games model we can express a whole spectrum of notions of equality of terms $x : A \vdash s, t : B$ using the Id-types, ranging from

strict intensional equality ($\vdash p : \mathsf{Id}_{\Pi_{x:A}B}(\lambda_{x:A}s, \lambda_{x:A}t)$) via observational equivalence with respect to contexts which take value in B or, put differently, provide an argument of type A ($x : A \vdash p : \mathsf{Id}_B(s,t)$) to observational equivalence with respect to contexts which take value in ground types (by pulling all the Π-constructors which occur in B out of the Id-constructor as well).

The principle FunExt of function extensionality intuitively state that, from the point of view of the Id-types, functions are extensional objects: black boxes which merely send inputs to outputs without any internal temporal structure. It is refuted in our model.

Theorem 6.4. FunExt *is refuted: for* $\vdash f, g : \Pi_{x:A}B$*, we do not generally have* $z : \Pi_{x:A}\mathsf{Id}_B(f(x), g(x)) \vdash \mathsf{FunExt}_{f,g} : \mathsf{Id}_{\Pi_{x:A}B}(f,g)$.

Proof. For our counter example, we let $[\![A]\!] = [\![B]\!] = [\mathbb{B}_*]$.

Let f be the non-strict constantly tt strategy $[\![f]\!] = [\{\epsilon, *\mathsf{tt}\}]$ and let g be the usual strict strategy that outputs tt (and examines its argument once) $[\![g]\!] = [\{\epsilon\} \cup \{*(i,*) \mid i\} \cup \{*(i,*)(i,\mathsf{tt})\mathsf{tt} \mid i\} \cup \{*(i,*)(i,\mathsf{ff})\mathsf{tt} \mid i\}]$. Noting that $[\![f]\!]\{x\} \cap [\![g]\!]\{x\} = [\![g]\!]\{x\} = [\{\epsilon, *\mathsf{tt}\}]$ if $x \neq \bot$ and else $[\{\epsilon, *\}]$, we have an inhabitant $\{\epsilon, *\mathsf{tt}\} \in \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{\mathbb{B}_*}[\![g]\!])) = \mathsf{Tm}([\mathbb{B}_*], \mathsf{Id}_{[\mathbb{B}_*]}([\![f]\!], [\![g]\!]))$. However, we do not have an inhabitant of

$$\mathsf{Tm}([], [\mathsf{Id}_{\Pi_{[\mathbb{B}_*]}[\mathbb{B}_*]}]([\![f]\!], [\![g]\!])) = \mathsf{Tm}([], [\![f]\!] \cap [\![g]\!]) = \mathsf{wstr}(\emptyset_*) = \emptyset.$$

□

On the other hand, it turns out that we do have the principle of uniqueness of identity proofs UIP, by playing derelicted copycats between (the first) $[\![\mathsf{Id}_A]\!]$ and $[\![\mathsf{Id}_{\mathsf{Id}_A}]\!]$. This principle intuitively says that types have trivial (discrete) spatial structure, from the point of view of the Id-types.

Theorem 6.5. *We have* $x, y : A, p, q : \mathsf{Id}_A(x,y) \vdash \mathsf{UIP}_A : \mathsf{Id}_{\mathsf{Id}_A(x,y)}(p,q)$.

Proof. Let us write $[X_i]_i$ for $[\![A]\!]$. We can play derelicted copycats between $!\mathsf{Id}_{X_i}$ and $\mathsf{Id}_{\mathsf{Id}_{X_i}}$. This follows as all elements of $\mathsf{Ctxt}(\mathbf{DGame}_!)([], [X_i]_i.[X_i]_i.[\mathsf{Id}_{X_i}]_i)$ are all of the form $\langle x, x, x\rangle$ for $x \in \mathsf{Ctxt}(\mathbf{DGame}_!)([], [X_i]_i)$. Therefore, when a move is first made in $!\mathsf{Id}_{X_i}$ (by Opponent), it automatically narrows down the fibre of $\mathsf{Id}_{\mathsf{Id}_{X_i}}$ such that the move can be copied. Moreover, Id-types contain all O-moves so Player can make all her moves in $!\mathsf{Id}_{X_i}$ □

7. Ground Types: Finite Dependent Type Families

In this section, we show how we can additionally give finite inductive type families an interpretation. To interpret their elimination rule using history-free strategies, we restrict our attention to the full subcategory $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi Id}}$ of $\mathsf{Ctxt}(\mathbf{DGame}_!)$ on the hierarchy of context games generated by the semantic constructions interpreting 1-, Σ-, Π- and Id-types and substitution, starting from finite dependent games (as defined below). These finite dependent games will play the rôle of semantic ground types to build a type hierarchy for which we prove completeness results in the next section.

We consider the interpretation of $\mathsf{DTT}-$ in $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi Id}}$ and will denote the interpretation functor by $[\![-]\!]$.

Theorem 7.1 (Finite Dependent Game)**.** *A finite inductive type family* $B := (a_i \mapsto_i \{b_{i,j} \mid j\})(x)$ *in context* $x : A$*, where* $B[a_i/x]$*, for* $1 \leq i \leq n$ *is generated by* $\{b_{ij} \mid 1 \leq j \leq m_i\}$*, has an interpretation in* $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi Id}}$ *as a* ***finite dependent game****:*

$$[\![B]\!] : [\![a_i]\!] \longmapsto [\{b_{i,j} \mid 1 \leq j \leq m_i\}_*] \qquad \text{else} \longmapsto [\emptyset_*]$$

and

$$\textcircled{\scriptsize ☺}([\![B]\!]) = \{b_{i,j} \mid 1 \leq i \leq n,\ 1 \leq j \leq m_i\}_*.$$

Proof. To be explicit, we interpret the $\mathsf{case}^{p,q}$-constructs rather than the (equivalent) case-constructs, as we shall be using the former later.

For interpreting the F-rule, note that all $[\![a_i]\!]$ can be interpreted (by applying $\textcircled{\scriptsize ☺}(-)$) as winning strategies on games in the simply typed hierarchy over finite inductive games and hence are finite objects. Therefore, continuity of the function interpreting the type forming operation for finite inductive type families holds trivially.

The interpretation of the I-rules is clear: $[\![b_{i,j}]\!]$ is the unique strategy on $[\![B[a_i/x]]\!]$ that replies to $*$ with the move $b_{i,j}$.

We inductively construct $[\![\mathsf{case}^{p,q}_{B[a/x],C}(b,\{c_{i,j}\}_{i,j})]\!] : [\![\cdot]\!] \longrightarrow [\![\Pi_{A'}C[b/y]]\!]$, with structural induction on C (apart from the case of $C = 1$, which is trivial). We consider the (more general) base case of arbitrary $[\![C]\!]$ that assign to each $\sigma \in \mathsf{str}(☺([\![A']\!])\&☺([\![B]\!])\{[\![a]\!]\})$ a finite inductive game with initial move $*$. After that, the case constructs for more general C are obtained from commutative conversions. Note that substitutions are already dealt with because we have been considering the more general base case where some of the constructors of $[\![C]\!]$ can coincide, while subtitution commutes with Π, Σ and Id.

Let us consider our base case. We define $[\![\mathsf{case}^{p,q}_{B[a/x],C}(b,\{c_{i,j}\}_{i,j})]\!]$ by noting that

$$[\![\mathsf{case}_{B^T,C^T}]\!]([\![b^T]\!],\{[\![c^T_{i,j}]\!](\mathsf{der}_{A'},[\![a^T]\!],[\![b^T]\!))\}_{i,j})$$

in fact defines a (winning) strategy on $[\![\Pi_{A'}C[b/y]]\!]$, where $(-)^T$ is the syntactic translation from section 2.

We verify that this yields a strategy $[\![\mathsf{case}^{p,q}_{B[a/x],C}(b,\{c_{i,j}\}_{i,j})]\!]$ on $[\![\Pi_{A'}C[b/y]]\!]$ (which clearly automatically is winning, as usual, as we never restrict O-moves in games of dependent functions). Let us write $[\![A']\!] = [X_i]_i$, so $[\![\mathsf{case}^{p,q}_{B[a/x],C}(b,\{c_{i,j}\}_{i,j})]\!]$ will be a strategy on $\mathsf{O\text{-}sat}(\Pi_{X_1}\cdots\Pi_{X_n}[\![C[b/y]]\!])$.

Let $sa'b' \in [\![\mathsf{case}^{p,q}_{B[a/x],C}(b,\{c_{i,j}\}_{i,j})]\!]$. Then, we verify that for all $\overline{sa'b'\restriction_{!☺(X_1)}} \subseteq \sigma'_1 \in \mathsf{wstr}(X_1()),\dots\ \overline{sa'b'\restriction_{!☺(X_n)}} \subseteq \sigma'_n \in \mathsf{wstr}(X_n(\sigma'_1,\dots,\sigma'_{n-1}))$, we have that

$$sa'b'\restriction_{!☺(X_1)} \in P_{!X_1()} \wedge \cdots \wedge sa'b'\restriction_{!☺(X_n)} \in P_{!X_n(\sigma'_1,\dots,\sigma'_{n-1})} \wedge sa'b'\restriction_{☺([\![C]\!])} \in P_{[\![C[b/y]]\!](\sigma'_1,\dots,\sigma'_n)} \qquad (*),$$

provided that already

$$sa'\restriction_{!☺(X_1)} \in P_{!X_1()} \wedge \cdots \wedge sa'\restriction_{!☺(X_n)} \in P_{!X_n(\sigma'_1,\dots,\sigma'_{n-1})} \wedge sa'\restriction_{☺([\![C]\!])} \in P_{[\![C[b/y]]\!](\sigma'_1,\dots,\sigma'_n)}.$$

Clearly, because of the type $[\![\Pi_{A'}B[a/x]]\!]$ of $[\![b]\!]$, all Player moves of $[\![\mathsf{case}^{p,q}_{B[a/x],C}(b,\{c_{i,j}\}_{i,j})]\!]$ respect the type $[\![\Pi_{A'}C[b/y]]\!]$ at least until some $[\![c_{i,j}]\!]$ is called. Now, the crux is that $[\![c_{i,j}]\!]$ is only ever called after $[\![b]\!]$ has already replied with the move $b_{i,j}$. This means that for any $[\sigma'_i]_i$ we are considering, we have that $[\sigma'_i]_i;[\![b]\!] = [\![b_{i,j}]\!]$. Moreover, because of the type of b, we have that $[\![b_{i,j}]\!] = [\sigma'_i]_i;[\![b]\!] = [\![b]\!]\{[\sigma'_i]_i\}$ is a winning strategy on $[\![B]\!]\{[\![a]\!]\{[\sigma'_i]_i\}\}$, while $[\![a]\!]\{[\sigma'_i]_i\}$ is a winning strategy on $[\![A]\!]$. Therefore, because of the definition of $[\![B]\!]$, we conclude that $[\sigma'_i]_i;[\![a]\!] = [\![a_i]\!]$, as the fibres of B are disjoint. The upshot is that the semantic type $[\![\Pi_{x':A'}\Pi_{p_{i,j}:\mathsf{Id}_A(a_i,a)}\Pi_{q_{i,j}:\mathsf{Id}_{B[a/x]}(\mathsf{subst}(p_{i,j},b_{i,j}),b)}C[b/y]]\!]$ of $[\![c_{i,j}]\!]$ now gives us that the continuation of the play along $[\![c_{i,j}]\!](\mathsf{der}_{A'},[\![a]\!],[\![b]\!])$ still respects our condition $(*)$.

For B-Discr, note that $\mathsf{Id}_{B(a_i)}(b_{i,j},b_{i',j'})() = \{*\} = \emptyset_*$ if $(i,j) \neq (i',j')$, which has no winning strategies. Hence, we can interpret $\mathsf{exfalso}$ by the non-strict strategy $[\![c]\!]$. □

We have obtained the following.

Corollary 7.2. *We have a sound interpretation of DTT− in* $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1}\Sigma\Pi\mathsf{Id}}$.

We turn to the issue of soundness of the interpretation of DTT in the next section.

Remark 7.3. *Note that we could also model finite inductive type families $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$ where A contains non-trivial Π-type constructors. (This is more general than what we considered in section 2.) However, in this case, we note that there are no* ***history-free*** *winning strategies on $\mathsf{O\text{-}sat}(\Pi_{[\![A]\!]}[\![(a_i \mapsto_i \{b_{i,j} \mid j\})(x)]\!])$ as $(a_i \mapsto_i \{b_{i,j} \mid j\})(x)$ does not respect observational equivalence. Indeed, for each $\vdash a_i : A$ there is some $\vdash a' : A$ which is observationally equivalent but not judgementally equal while the fibres of $[\![(a_i \mapsto_i \{b_{i,j} \mid j\})(x)]\!]$ are disjoint in the sense that distinct fibres share no winning strategies.*

As an example, let $A = \mathbb{B}_* \Rightarrow \mathbb{B}_*$ and $B(\lambda_{x:\mathbb{B}}\mathsf{tt}) = \{1\}_*$, $B(\lambda_{x:\mathbb{B}}\mathsf{case}(x,\mathsf{tt},\mathsf{tt})) = \{2\}_*$ and all other fibres non-empty as well. In that case, a winning strategy τ on $\mathsf{O\text{-}sat}(\Pi_A B)$ has to be strict, as a non-strict winning strategy would make Player exclude some fibre by either playing 1 or 2. Now, history-freeness and the bracketing condition mean that τ can base its response in B only on the observational equivalence class of the argument of type A it is given. However, the fibres $B(\lambda_{x:\mathbb{B}}\mathsf{case}(x,\mathsf{tt},\mathsf{tt}))$ and $B(\lambda_{x:\mathbb{B}}\mathsf{tt})$ have intersection $\emptyset_*$, meaning that player cannot respond in B if Opponent provides the argument $\lambda_{x:\mathbb{B}}\mathsf{tt}$.

We have therefore not considered this level of generality. Moreover, the definability argument given in the next section would not extend to this setting.

8. Soundness, Faithfulness and Completeness Results

In this section, we show that the interpretation of DTT in $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi Id}}$ is sound and faithful and, if we limit Id-types to only occur strictly positively and at most once, that it is, additionally, fully complete. The proof of soundness and faithfulness follows from the fact that our game semantics for DTT factors faithfully over the usual game semantics for simple type theory. The proof of definability proceeds in six steps:

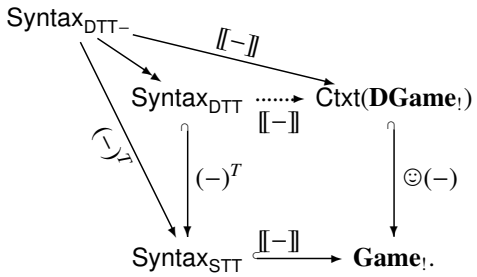
1. interpreting a dependently typed strategy f on a larger (simply typed) game;
2. for a strict f, performing the decomposition of [8] in the simply typed world, as usual, to obtain simply typed strategies g^j and h_y that are called in the execution of f;
3. noting that these g^j and h_y can actually be assigned a more precise dependent type, the trick being that we accumulate appropriate negatively occurring Id-types as the decomposition proceeds inductively;
4. observing that the iterated decomposition of strict strategies strictly decreases a positive integer norm and therefore eventually terminates after finitely many steps, producing only non-strict strategies;
5. for a non-strict f, noting that f is directly definable using the constructors $b_{i,j}$ for finite type families and an appropriate Id-type witness r to map $b_{i,j}$ to the appropriate fibre, using a subst-term for Id-types;
6. constructing this r, using a generalised case construct, reflexivity witnesses and the exfalso-eliminator for uninhabited identity types.

8.1. Soundness, Faithfulness and Some Steps Towards Completeness

We first prove faithfulness of the interpretation of DTT in our model.

Theorem 8.1 (Soundness and Faithfulness)**.** *The interpretation* $[\![-]\!]$ *of DTT in* $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi Id}}$ *is sound and faithful.*

Proof. We note that we have the following commutative diagram of (non-dashed) functors, where, in the light of corollary 7.2, soundness amounts to arguing that our interpretation of DTT− factors over DTT (denoting the factorisation with the dashed functor)



Here, the top and bottom sides of the outer square, respectively are the interpretation functor of DTT− in our model, which exists according to corollary 7.2, and the usual interpretation of simple type theory with finite ground types (or, a total finitary PCF, if you will) in the intuitionistic category of games and winning history-free strategies of [8]. Recall that the latter is (full and) faithful according to theorem 3.13. The left side of the inner square is the faithful (non-full) functor defined in section 2.4. Note that faithfulness of the interpretation of DTT automatically follows from the faithfulness of these two functors, if we can prove soundness. Finally, the right side of either square is the semantic equivalent of this syntactic translation, which we define next.

We have an inductively defined translation $\mathsf{Ctxt}(\mathbf{DGame}_!) \xrightarrow{☺(-)} \mathbf{Game}_!$:

$$☺([A_i]_{1\leq i\leq m}) := \mathop{\&}_{1\leq i\leq m} ☺(A_i)$$
$$☺([]) := I.$$

Note that this also satisfies

$$☺(\Pi_{A_1}\cdots\Pi_{A_n}B) = ☺(A_1)\Rightarrow\cdots\Rightarrow ☺(A_n)\Rightarrow ☺(B)$$
$$☺(\mathsf{Id}_C) = ☺(C).$$

This automatically extends to a faithful (non-full) functor by interpreting the winning dependently typed strategies on A as simply typed strategies on $☺(A)$, which are obviously also winning as we never restrict Opponent moves in our games of dependent functions. Clearly, our strategies remain history-free, seeing that we can use the same history-free skeleton. Faithfulness of this functor together with commutativity of the outer square gives us that the dashed arrow is a (unique) well-defined functor, i.e. we have a sound interpretation of DTT in $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi Id}}$. □

Next, we first prove two lemmas, which encompass steps 2. and 3. and, respectively, step 4. in the definability proof.

Lemma 8.2 (Decomposition)**.** *Let us suppose we have a context game $[A_i]_{i\leq n}$ in $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi Id}}$ with $A_i = \Pi_{B^{i,1}} \dots \Pi_{B^{i,q_i}} Y^i_* = \Pi_{[B^{i,j}]_j} Y^i_*\{c^i\}$ where Y^i_* is a finite inductive dependent game depending on the context game $[C^i_l]_l$ and $[A_k]_{k<i}.[B^{i,j}]_j \xrightarrow{c^i} [C^i_l]_l$. Let us say Y^i_* has constructors y in fibre $Y^i_*\{[c^i_y]_i\}$.*

Then, it follows that, when given a strategy f that does not visit $\mathsf{Id}_{[D_k]_k}$,

$$f \in \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{[A_i]_i}\Pi_{\mathsf{Id}_{[D_k]_k}([d^0_k]_k,[d_k]_k)}X_*)),$$

with $\mathsf{str}(☺(A_1)\&\cdots\&☺(A_n)) \xrightarrow{X} \mathcal{P}(☺(X))$ a continuous function, where $☺(X)$ is some finite set, and context morphisms $[d_k]_k, [d^0_k]_k : [A_i]_i \longrightarrow [D_k]_k$, we can decompose it (uniquely) as follows:

- *if f is non-strict, then $f = [A_i]_i \longrightarrow [] \xrightarrow{x} [☺(X_*)]$ for some $x \in \bigcup \mathsf{im}(X)$ such that $x \in X_*([\tau_i]_i)$ for all $[] \xrightarrow{[\tau_i]_i} [A_i]_i$ such that $[d_k]_k\{[\tau_i]_i\} = [d^0_k]_k\{[\tau_i]_i\}$;*
- *if f is strict, then $f = \mathbf{C}_i(g^1, \dots, g^{q_i}, (h_y \mid y \in \bigcup \mathsf{im}(Y^i)))$ where $\mathbf{C}_i$ embodies a case-construct that we shall define in the proof,*

where

$$g^j \in \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{[A_i]_i}\Pi_{\mathsf{Id}_{[D^k]_k}([d^0_k]_k,[d_k]_k)}B^{i,j}\{\langle[\mathsf{der}_{A_l}]_{l<i}, [g^{j'}]_{j'<j}\rangle\}))$$

and

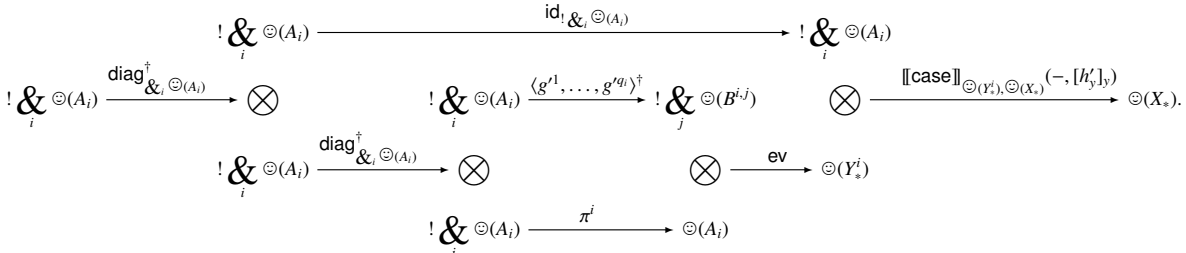
$$h_y \in \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{[A_i]_i}\Pi_{\mathsf{Id}_{[D^k]_k.[C^i_l]_l.[Y^i_*]}(\langle[d^0_k]_k,[c^i_y],[y]\rangle,\langle[d_k]_k,[\widetilde{c^i}],[\phi]\rangle)}X_*)),$$

where $\widetilde{c^i} := \langle[\mathsf{der}_{A_{i'}}]_{i'<i}, [g_j]_j\rangle; c^i$ and $\phi := \lambda_{[\tau_k]_k}\tau_i\{[g^j]_j\{[\tau_k]_k\}\}$ (and we write $\mathsf{im}(Y^i)$ for the image of Y^i and $\lambda_{[\tau_k]_k}$ for the obvious semantic λ-abstraction). Here, neither g^j nor h_y visits the Id-type.

Proof. Note that we can consider $☺(f)$ as a strategy on $☺(A_1) \Rightarrow \cdots \Rightarrow ☺(A_n) \Rightarrow ☺(X_*)$ as f does not visit the Id-type. The decomposition lemma [8, 23] for the game semantics of (finitary) PCF now gives us three cases:

- $☺(f) = \bot$
- $☺(f) = \&_i ☺(A_i) \longrightarrow I \xrightarrow{x} ☺(X_*)$ for some $x \in \bigcup \mathsf{im}(X)$;
- $☺(f) = \mathbf{C}'_i(g'^1, \dots, g'^{q_i}, (h'_y \mid y \in \bigcup \mathsf{im}(Y^i)))$, for a (unique) $1 \leq i \leq n$ and (unique) $(g'^j) \in \mathsf{str}(☺(A_1) \Rightarrow \cdots \Rightarrow ☺(A_n) \Rightarrow ☺(B^{i,j}))$ and $h'_y \in \mathsf{str}(☺(A_1) \Rightarrow \cdots \Rightarrow ☺(A_n) \Rightarrow ☺(X_*))$, where (writing π^i for the derelicted projection to the i-th component, ev for the obvious evaluation morphism, and denoting the semantic case construct with $[\![\mathsf{case}]\!]$)

$$\mathbf{C}'_i(g'^1, \dots, g'^{q_i}, (h'_y \mid y \in \bigcup \mathsf{im}(Y^i))) :=$$



Note that the first case cannot occur as f is winning.

For the second case, due to the restriction on P-moves in Π-games and the interpretation of Id-types, a partial function on moves precisely defines a non-strict strategy if it responds to $*$ with a move in $\bigcup \mathsf{im}(X)$ such that $x \in X_*([\tau_i]_i)$ for all $[] \xrightarrow{[\tau_i]_i} [A_i]_i$ such that $[d_k]_k\{[\tau_i]_i\} = [d_k^0]_k\{[\tau_i]_i\}$.

For the third case, note the following.

- $g'^j = \mathbb{O}(g^j)$ for (unique) $g^j \in \mathsf{str}(\mathsf{O\text{-}sat}(\Pi_{[A_i]_i}\Pi_{\mathsf{Id}_{[D^k]_k}([d_k^0]_k,[d_k]_k)}B^{i,j}\{\langle[\mathsf{der}_{A_l}]_{l<i},[g^{j'}]_{j'<j}\rangle\}))$. This will follow once we show that $((g^j)^\dagger)^\dagger \in \mathsf{str}(\mathsf{O\text{-}sat}(\Pi_{[A_i]_i}\Pi_{\mathsf{Id}_{[D^k]_k}([d_k^0]_k,[d_k]_k)}!!B^{i,j}\{\langle[\mathsf{der}_{A_l}]_{l<i},[g^{j'}]_{j'<j}\rangle\}))$. The argument will proceed by complete induction on j. Assume the claim holds for g^k with $k < j$. We will show it also holds for g^j.

  We need to show that for $s^j = s'ab \in ((g^j)^\dagger)^\dagger$, for any $\overline{s'ab\restriction_{!\mathbb{O}(A_1)}} \subseteq \tau_1 \in \mathsf{wstr}(A_1()), \ldots, \overline{s'ab\restriction_{!\mathbb{O}(A_n)}} \subseteq \tau_n \in \mathsf{wstr}(A_n(\tau_1,\ldots,\tau_{n-1}))$ s.t. $[\tau_i]_i;[d_k^0]_k = [\tau_i]_i;[d_k]$, $s'a \in P_{A_1()\Rightarrow\cdots\Rightarrow A_n(\tau_1,\ldots,\tau_{n-1})\Rightarrow !!B^{i,j}(\tau_1,\ldots,\tau_{i-1},\langle\tau_1,\ldots,\tau_n\rangle;g^1,\ldots,\langle\tau_1,\ldots,\tau_n\rangle;g^{j-1})}$ implies that also $s'ab \in P_{A_1()\Rightarrow\cdots\Rightarrow A_n(\tau_1,\ldots,\tau_{n-1})\Rightarrow !!B^{i,j}(\tau_1,\ldots,\tau_{i-1},\langle\tau_1,\ldots,\tau_n\rangle;g^1,\ldots,\langle\tau_1,\ldots,\tau_n\rangle;g^{j-1})}$.

  Let us assume that the hypothesis of this implication is true. Now, note that $s' \in ((g^j)^\dagger)^\dagger$ extends to $tab = *_{X_*}(0,*)_{!Y^i_*}s^1\cdots s^{j-1}s^j \in f$ for any $s^k \in ((g^k)^\dagger)^\dagger$, for $1 \le k \le j-1$. We can choose $s^k \in \langle\tau_1,\ldots,\tau_n\rangle\|((g^k)^\dagger)^\dagger$ such that $\bigcup\overline{\overline{s^k}\restriction_{!!\mathbb{O}(B^{i,k})}} = \langle\tau_1,\ldots,\tau_n\rangle;g^k|_{B^{i,k}(\tau_1,\ldots,\tau_{i-1},\langle\tau_1,\ldots,\tau_n\rangle;g^1,\ldots,\langle\tau_1,\ldots,\tau_n\rangle;g^{k-1})}$. (We write $\overline{s}$ to indicate we apply $\overline{(-)}$ first to s and then again to each member of the resulting set of plays.) Note that we can do this as $\langle\tau_1,\ldots,\tau_n\rangle;g^k$ is finite as a partial function on moves. In fact, $s^k \in P_{A_1()\Rightarrow\cdots\Rightarrow A_n(\tau_1,\ldots,\tau_{n-1})\Rightarrow !!B^{i,k}(\tau_1,\ldots,\tau_n,\langle\tau_1,\ldots,\tau_n\rangle;g^1,\ldots,\langle\tau_1,\ldots,\tau_n\rangle;g^{k-1})}$, as a consequence of our induction hypothesis.

  Then, as $tab \in f$ is a play in $\mathsf{O\text{-}sat}(\Pi_{A_1}\cdots\Pi_{A_{i-1}}\Pi_{(\Pi_{B^{i,1}}\cdots\Pi_{B^{i,q_i}}Y^i_*)}\Pi_{A_{i+1}}\cdots\Pi_{A_n}\Pi_{\mathsf{Id}_{[D^k]_k}([d_k^0]_k,[d_k]_k)}X_*)$, we have that for all $\overline{tab\restriction_{!\mathbb{O}(A_1)}} \subseteq \tau'_1 \in \mathsf{wstr}(A_1()), \ldots, \overline{tab\restriction_{!\mathbb{O}(A_n)}} \subseteq \tau'_n \in \mathsf{wstr}(A_n(\tau'_1,\ldots,\tau'_{n-1}))$ s.t. $[\tau'_i]_i;[d_k^0]_k = [\tau'_i]_i;[d_k]$, $ta \in P_{A_1()\Rightarrow\cdots\Rightarrow A_n(\tau'_1,\ldots,\tau'_{n-1})\Rightarrow X_*(\tau'_1,\ldots,\tau'_n)}$ implies that also $tab \in P_{A_1()\Rightarrow\cdots\Rightarrow A_n(\tau'_1,\ldots,\tau'_{n-1})\Rightarrow X_*(\tau'_1,\ldots,\tau'_n)}$. Note that by construction of tab, $[\tau_i]_i$ is one such $[\tau'_i]_i$ and is in fact the only one we are interested in, so we simply write $[\tau_i]_i$ for both. Note that the hypothesis of the implication under consideration actually holds by our assumptions about $s'a$ and s^k. Therefore, its conclusion $tab \in P_{A_1()\Rightarrow\cdots\Rightarrow A_n(\tau_1,\ldots,\tau_{n-1})\Rightarrow X_*(\tau_1,\ldots,\tau_n)}$ follows.

  Now, it follows immediately from the restriction on plays tab in $\mathsf{O\text{-}sat}(\Pi_{A_1}\cdots\Pi_{A_{i-1}}\Pi_{(\Pi_{B^{i,1}}\cdots\Pi_{B^{i,q_i}}Y^i_*)}\Pi_{A_{i+1}}\cdots\Pi_{A_n}\Pi_{\mathsf{Id}_{[D^k]_k}([d_k^0]_k,[d_k]_k)}X_*)$ that if g^j makes the move b in $!\mathbb{O}(A_l)$, then it also satisfies the rules of $\mathsf{O\text{-}sat}(\Pi_{[A_i]_i}\Pi_{\mathsf{Id}_{[D^k]_k}([d_k^0]_k,[d_k]_k)}B^{i,j}\{\langle[\mathsf{der}_{A_l}]_{l<i},[g^{j'}]_{j'<j}\rangle\})$. The interesting case is when b is a move in $!!B^{i,j}$. To deal with this case, we note that the restriction on plays tab in $\mathsf{O\text{-}sat}(\Pi_{A_1}\cdots\Pi_{A_{i-1}}\Pi_{(\Pi_{B^{i,1}}\cdots\Pi_{B^{i,q_i}}Y^i_*)}\Pi_{A_{i+1}}\cdots\Pi_{A_n}\Pi_{\mathsf{Id}_{[D^k]_k}([d_k^0]_k,[d_k]_k)}X_*)$ combined with the definition of $(\Pi_{B^{i,1}}\cdots\Pi_{B^{i,q_i}}Y^j{}_*)(\tau_1,\ldots,\tau_i)$ that there exist $\bigcup\overline{\overline{tab\restriction_{!!\mathbb{O}(B^{i,1})}}} \subseteq \sigma^1 \in \mathsf{wstr}(B^{i,1}(\tau_1,\ldots,\tau_n)),\ldots,\bigcup\overline{\overline{tab\restriction^{\uparrow}_{!!\mathbb{O}(B^{i,q_i})}}} \subseteq \sigma^{q_i} \in \mathsf{wstr}(B^{i,q_i}(\tau_1,\ldots,\tau_n,\sigma^1,\ldots,\sigma^{q_i-1}))$, such that $tab\restriction_{!\mathbb{O}(A_i)} \in P_{!(B^{i,1}(\tau_1,\ldots,\tau_n)\Rightarrow\cdots\Rightarrow B^{i,q_i}(\tau_1,\ldots,\tau_n,\sigma^1,\ldots,\sigma^{q_i-1})\Rightarrow Y^i{}_*(\tau_1,\ldots,\tau_n,\sigma^1,\ldots,\sigma^{q_i}))}$ so, in particular, $tab\restriction_{!\mathbb{O}(A_i)}\restriction_{!!\mathbb{O}(B^{i,j})} \in P_{!!B^{i,j}(\tau_1,\ldots,\tau_n,\sigma^1,\ldots,\sigma^{j-1})}$.

  To complete the argument, we note that by construction of t, we have that $\bigcup\overline{\overline{tab\restriction_{!!\mathbb{O}(B^{i,k})}}} = \langle\tau_1,\ldots,\tau_n\rangle;g^k|_{B^{i,k}(\tau_1,\ldots,\tau_{i-1},\langle\tau_1,\ldots,\tau_n\rangle;g^1,\ldots,\langle\tau_1,\ldots,\tau_n\rangle;g^{k-1})}$, for $1 \le k \le j-1$. We conclude that $s'ab\restriction_{!!B^{i,j}} \in P_{!!B^{i,j}(\tau_1,\ldots,\tau_n,\langle\tau_1,\ldots,\tau_n\rangle;g^1,\ldots,\langle\tau_1,\ldots,\tau_n\rangle;g^{j-1})}$.

- $h'_y = \mathbb{O}(h_y)$ for (unique) $h_y \in \mathsf{str}(\mathsf{O\text{-}sat}(\Pi_{A_1}\cdots\Pi_{A_n}\Pi_{\mathsf{Id}_{[D^k]_k.[C^i_l]_l.[Y^i_*]}(\langle[d_k^0]_k,[c^i_y],[y]\rangle,\langle[d_k]_k,[\widetilde{c^i}],[\phi]\rangle)}X_*))$. Indeed, note that $*s \in h_y$ iff $*(0,*)t(0,y)s \in f$ for some $*ty \in \phi \in \mathsf{wstr}(\mathsf{O\text{-}sat}(\Pi_{A_1}\cdots\Pi_{A_n}\Pi_{\mathsf{Id}_{[D^k]_k}([d_k^0]_k,[d_k]_k)}Y^i_*\{\langle[\mathsf{der}_{A_l}]_{l<i},[g^j]_j\rangle\}))$. It then follows that $*s \in \mathsf{O\text{-}sat}(\Pi_{A_1}\cdots\Pi_{A_n}\Pi_{\mathsf{Id}_{[D^k]_k.[C^i_l]_l.[Y^i_*]}(\langle[d_k^0]_k,[c^i_y],[y]\rangle,\langle[d_k]_k,[\widetilde{c^i}],[\phi]\rangle)}X_*)$ by the following observation. Observing that $\phi\{[\overline{t\restriction_{!A_i}}]_i\} = y$, note that, for winning $[\tau_i]_i \ge [\overline{*s\restriction_{!A_i}}]_i$, we also have that $[\tau_i]_i \ge [\overline{*(0,*)t(0,y)s\restriction_{!A_i}}]_i$ for some $*ty \in \phi$ iff $\phi\{[\tau_i]_i\} \ge y$ i.e. $\phi\{[\tau_i]_i\} = y$ as y is a maximal strategy on $\mathbb{O}(Y_*)$. (Indeed, we can take $*ty \in \langle\tau_1,\ldots,\tau_n\rangle\|\phi$.) It automatically then follows that also $\widetilde{c^i}\{[\tau_i]_i\} = c^i_y$.

- We can now note that $f = \mathbf{C}_i(g^1, \ldots, g^{q_i}, (h_y \mid y \in \bigcup \mathsf{im}(Y^i)))$, where $\mathbf{C}_i$ is defined exactly as $\mathbf{C}'_i$ but using instead the dependently typed substitution and the dependently typed construct $[\![\mathsf{case}^{p,q}]\!]_{Y^i_*\{\langle [\mathsf{der}_{A_l}]_{l<i}, [g^j]_{j\leq q_i}\rangle\}, X_*}$. (That is, $\mathbf{C}_i$ and $\mathbf{C'}_i$ are the same, except for typing.) Note that $\mathsf{ev}(\pi^i, \langle g^1, \ldots, g^{q_i}\rangle)$ and h_y feed into this case construct.

- Finally, to see that g^j and h_y define winning strategies, we note that their infinite plays are Player-wins as they arise as labelled subtrees of f which is winning. We need to verify that they are total. This also follows immediately from the totality of f together with the fact that Opponent moves are, by definition, not restricted in (O-saturated) games of dependent functions. Indeed, if $s \in g^j$ and sa is a valid extension of the play, then $** sa$ is a valid extension of $** s \in f$ to which f hence g^j has a response b. Similarly, if $*s \in h_y$ and $*sa$ is a valid extension of the play, then $*(0,*)t(0,y)sa$ is a valid extension of $*(0,*)t(0,y)s \in f$ to which f has a response b, being a total strategy. Therefore, $*sab \in h_y$.

□

Lemma 8.3 (Norm for dependent strategies)**.** *Let* $[A_i]_i \xrightarrow{[d_k]} [D_k]_k$ *and* X_* *as in the previous lemma. Let us write* $E := \Pi_{[A_i]_i}\Pi_{\mathsf{Id}_{[D^k]_k}([d^0_k],[d_k]_k)}X_*$. *Then, we have a norm* $\|-\|_E : \mathsf{wstr}(\mathsf{O\text{-}sat}(E)) \longrightarrow \mathbb{N}$ *- we sometimes leave out the subscript E - for any such E such that* $f = \mathbf{C}_i(g^1, \ldots, g^{q_i}, (h_y \mid y \in \bigcup \mathsf{im}(Y^i)))$ *implies that*

$$\|g^i\|, \|h_y\| < \|f\|.$$

Proof. We define a norm $\|-\|_{☺(E)} : \mathsf{wstr}(☺(E)) \longrightarrow \mathbb{N}$ for games $☺(E)$ of the $I\& \Rightarrow$-hierarchy over finite flat games and extend this to a norm on $\mathsf{wstr}(\mathsf{O\text{-}sat}(E))$ by precomposition with the injection $\mathsf{wstr}(\mathsf{O\text{-}sat}(E)) \xrightarrow{☺(-)} \mathsf{wstr}(☺(E))$. The idea behind this norm is that winning strategies on games of the $I\& \Rightarrow$-hierarchy over finite flat games are finite objects in the sense that they only contain finitely many finite plays if we do not allow Opponent to open multiple threads of the same game - remember that infinite plays in winning strategies are always due to Opponent opening an infinite number of threads of the same game.

Inductively, if T is a type of STT (i.e. formed from finite ground types G by the grammar $T ::= G \mid \top \mid \& \mid \Rightarrow$), we define a type LT of intuitionistic linear logic without disjunctions over finite types (i.e. formed from finite ground types G by the grammar $LT ::= G \mid !LT \mid LT \multimap LT \mid LT \otimes LT \mid LT\&LT \mid I \mid \top$, where we note that in our interpretation $[\![\top]\!] = [\![I]\!]$ and where we identify the intuitionistic type $A \Rightarrow B$ with the linear type $!A \multimap B$) by removing each positive occurrence of ! in T or, equivalently, replacing each even-depth occurrence of $\Rightarrow$ with $\multimap$. Essentially, $[\![LT]\!]$ is obtained from the game $[\![T]\!]$ by not allowing Opponent to open more than one thread of any game. Note that we have a canonical winning strategy representing a generalised dereliction $[\![T]\!] \xrightarrow{\mathsf{gder}_{[\![LT]\!]}} [\![LT]\!]$ which is defined in the obvious way from dereliction maps on subtypes using the functoriality of $\top$, $\&$, $\Rightarrow$ and $\multimap$.

Now, if we can show that $W_{[\![LT]\!]} = \emptyset$, it follows that the norm $\|\sigma\|_{[\![T]\!]} := \Sigma_{s\in\sigma;\mathsf{gder}_{[\![LT]\!]}/\approx_{[\![LT]\!]}} \mathsf{length}(s)$ is well-defined for $\sigma \in \mathsf{wstr}([\![T]\!])$. (Here, by $\sigma; \mathsf{gder}_{[\![LT]\!]}/\approx_{[\![LT]\!]}$, we mean some skeleton for $\sigma; \mathsf{gder}_{[\![LT]\!]}$) Indeed, there are only finitely many Opponents for $[\![LT]\!]$ as Opponent can only make a choice between finitely many alternatives for each connective in formula LT, of which there are finitely many. Moreover, interactions with Player never become unboundedly long because $W_{[\![LT]\!]} = \emptyset$.

We show that $W_{[\![LT]\!]} = \emptyset$. Define classes of formulas $\mathtt{AllWin}, \mathtt{NoWin}$ by mutual induction as follows. In this definition, we use G to stand for any game all of whose maximal positions are of length 2, F^A (respectively, F^N) and their subscripted versions to range over $\mathtt{AllWin}$ (respectively, $\mathtt{NoWin}$) games.

$$\begin{aligned} \mathtt{AllWin} :: &\quad G \mid F^A_1 \otimes F^A_2 \mid \ F^A_1 \& F^A_2 \mid !F^A \mid F^N \multimap F^A \\ \mathtt{NoWin} :: &\quad G \mid F^N_1 \otimes F^N_2 \mid \ F^N_1 \& F^N_2 \mid F^A \multimap F^N. \end{aligned}$$

It follows from a simple inductive argument that

- for all $\mathtt{AllWin}$ games F^A, $W_{F^A} = P^\infty_{F^A}$;
- for all $\mathtt{NoWin}$ formulas F^N, $W_{F^N} = \emptyset$.

Now, to conclude that $W_{[\![LT]\!]} = \emptyset$, we observe that $LT \in \mathtt{NoWin}$, as all occurrences of ! are negative.

Finally, if $f = \mathbf{C}_i(g^1, \dots, g^{q_i}, (h_y \mid y \in \bigcup \mathsf{im}(Y^i)))$, then, plays of g^j and h_y properly extend to plays of f as discussed in the previous proof. Therefore, it follows that $\|g^i\|, \|h_y\| < \|f\|$. □

Now, we combine steps 1.-4. to reduce the definability of strict strategies to that of non-strict ones.

Lemma 8.4 (Defining Strict Strategies from Non-Strict Ones)**.** *All morphisms in* $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi}}$ *are definable in DTT if we assume that the non-strict ones are, where we write* $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi}}$ *for the full subcategory of* $\mathsf{Ctxt}(\mathbf{DGame}_!)$ *on the objects formed by the interpretation of types of* DTT *formed without* Id*-constructors.*

Proof. Let T be a type of DTT with Π, Σ, 1 and finite inductive type families and let $f \in \mathsf{Ctxt}(\mathbf{DGame}_!)([], [\![T]\!])$. If $T = \Sigma_{x_1:T^1} \dots \Sigma_{x_{n-1}:T^{n-1}} T^n$ (including the case of $T = 1$ if $n = 0$), then, we know that both in the syntax and semantics f decomposes as $\langle f_1, \dots, f_n \rangle$. The interesting remaining case to deal with therefore is definability for $T = \Pi_{x:T'} S[q/x']$ where $x' : Q \vdash S$ type and $x : T' \vdash q : Q$, i.e. for $T = \Pi_{x:T'} S[q/x']$ where S is a finite inductive type family. (In that case $[\![S[q/x']]\!] = X_*$ has finite inductive games as fibres.)

Once we are in this situation, the argument to show that $f \in \mathsf{Ctxt}(\mathbf{DGame}_!)([], [\![T]\!]) = \mathsf{wstr}(\mathsf{O\text{-}sat}([\![T]\!]))$ is definable in DTT will proceed by complete induction on $\|f\|$, which will terminate according to lemma 8.3. For the sake of our inductive argument, let us consider the more general case of $f \in \mathsf{wstr}(\mathsf{O\text{-}sat}([\![\Pi_{x:T'} \Pi_{\mathsf{Id}_D(d,d^0)} S[q/x']]\!]))$ which does not visit the Id-type. Note that we may assume WLOG that $T' = \Sigma_{T^1} \dots \Sigma_{T^{n-1}} T^n$ with $T^i = \Pi_{T'^1} \cdots \Pi_{T'^{q_i}} U[v/x'']$, where $x'' : V \vdash U$ type and $x_1 : T^1, \dots, x_{i-1} : T^{i-1}, x'_1 : T^{1\prime}, \dots, x'_{q_i} : T^{q_i\prime} \vdash v : V$ and where $[\![U[v/x'']]\!] = Y^i_*$. This is where we invoke lemma 8.2.

If f is strict, then f can be expressed as

$$\mathbf{C}_i(g^1, \dots, g^{q_i}, (h_y \mid y \in \bigcup \mathsf{im}(Y^i))) = [\![\lambda_{x:T'} \mathsf{case}^{p,q}_{U[v/x''][\mathsf{fst}(x)/x_1,\dots,(i-1)-\mathsf{th}(x)/x_{i-1},G^1x/x'_1,\dots,G^{q_i}x/x'_{q_i}],S[q/x']}(x_i(G^1x)\cdots(G^{q_i}x), \{H_y x\}_y)]\!],$$

where by the induction hypothesis $g^i = [\![G^i]\!]$ and $h_y = [\![H_y]\!]$.

If f is non-strict, it is definable by assumption.

We conclude that f is definable in DTT. □

8.2. A Challenge: Definability of Non-Strict Strategies

Like we have seen, the results in the previous section allow us to reduce the definability of arbitrary strategies on the games we consider to that of non-strict strategies. While the definability of non-strict strategies is trivial in the world of simple types, the same is not true when we are considering dependent types.

A first example to consider when trying to understand the definability of non-strict strategies may be the following. Let us consider the decomposition of what is essentially the evaluation strategy on $[\![\Pi_{x:\mathbb{B}} \Pi_{f:\mathbb{B}\Rightarrow\mathbb{B}} \mathsf{just}(f(x))]\!]$ (playing a copycat between the first and second copy of $[\![\mathbb{B}]\!]$ and between the third copy of $[\![\mathbb{B}]\!]$ and $\copyright([\![\mathsf{just}]\!])$). The decomposition of this strategy requires us to define the non-strict strategies (appearing as some h_y in the decomposition lemma) which return y on $[\![\Pi_{x:\mathbb{B}} \Pi_{f:\mathbb{B}\Rightarrow\mathbb{B}} \Pi_{p:\mathsf{Id}_{\mathbb{B}}(y,f(x))} \mathsf{just}(f(x))]\!]$, where $y \in \{\mathsf{tt}, \mathsf{ff}\}$. In this case, we can simply define this strategy as $\lambda_x \lambda_f \lambda_p \mathsf{subst}(p, y)$. We see that even this simple example is not entirely trivial and requires us to make non-trivial use of the identity proof p.

A more complicated example is obtained if we consider the decomposition of the same strategy on the type $[\![\Pi_{x:\mathbb{B}} \Pi_{f:\mathbb{B}\Rightarrow\mathbb{B}} \mathsf{just}(\mathsf{strict}(f)(x))]\!]$ instead, where we write $\mathsf{strict}(f) := \lambda_{x:\mathbb{B}} \mathsf{case}_{\mathbb{B},\mathbb{B}}(x, f(\mathsf{tt}), f(\mathsf{ff}))$ for a strict version of f. Note that this type has the same interpretation in the model so we can in fact consider the same strategy on it. This time, it is less clear how we should define the non-strict strategy returning y on $[\![\Pi_{x:\mathbb{B}} \Pi_{f:\mathbb{B}\Rightarrow\mathbb{B}} \Pi_{p:\mathsf{Id}_{\mathbb{B}}(y,f(x))} \mathsf{just}(\mathsf{strict}(f)(x))]\!]$ which we obtain in the decomposition. We might try $\lambda_x \lambda_f \lambda_p \mathsf{subst}(q, \mathsf{subst}(p, y))$ where $x : \mathbb{B}, f : \mathbb{B} \Rightarrow \mathbb{B} \vdash q : \mathsf{Id}_{\mathbb{B}}(f(x), \mathsf{strict}(f)(x))$ can be obtained through a case distinction on x.

Things get genuinely more complicated, however, if x is a function. Note that in the previous example, we had to perform a case distinction on x, which suggests we might need a mechanism to perform case distinctions on functions to generalise to this example. Indeed, consider the evaluation strategy on $[\![\Pi_{x:\mathbb{B}\Rightarrow\mathbb{B}} \Pi_{f:(\mathbb{B}\Rightarrow\mathbb{B})\Rightarrow\mathbb{B}} \mathsf{just}(\mathsf{strict}(f)(x))]\!]$ (we can employ a similar definition of $\mathsf{strict}(f)$: one that tests the argument of f first on tt and then on ff and then applies f on a strict version of the argument of f to return). In the decomposition of this strategy, we obtain a non-strict strategy on $[\![\Pi_{x:\mathbb{B}\Rightarrow\mathbb{B}} \Pi_{f:(\mathbb{B}\Rightarrow\mathbb{B})\Rightarrow\mathbb{B}} \Pi_{p:\mathsf{Id}_{\mathbb{B}}(y,f(x))} \mathsf{just}(\mathsf{strict}(f)(x))]\!]$ which returns y. Again, we may want to define this strategy as

$\lambda_x\lambda_f\lambda_p\mathsf{subst}(q, \mathsf{subst}(p, y))$. Now, however, we would like to define q through a case distinction on the function x. In order to define such strategies, we introduce in the next section certain generalised case constructs which allow us to make a case distinction on the observational equivalence class of a (finite) function and observational identity types, which capture the observational equivalence of (finite) functions.

8.3. Syntactic Intermezzo: Generalised Case Constructs

In this section, we introduce some technical devices that we shall need to prove the definability of non-strict strategies in order to complete the completeness proof in the next section.

Let A' and B be closed types built without Id-type constructors and let $x' : A', y : B \vdash C\ \mathsf{type}$ be a type family. We can define a set $\mathsf{obs}(B) = \{b_1, \dots, b_{\#\mathsf{obs}(B)}\}$ of **canonical representatives of the observational equivalence classes** of closed terms of type B (which is independent of A' and C) and a type family $x' : A' \vdash \mathsf{IdObs}_B(b', b)\ \mathsf{type}$, for any $x' : A' \vdash b', b : B$, which we call the **observational identity type** of B (which measures observational equivalence of b' and b). Then, for any $x' : A' \vdash b : B$ and $\{x' : A', p_i : \mathsf{IdObs}_B(b_i, b) \vdash c_i : C[b/y]\}_{1\le i\le \#\mathsf{obs}(B)}$ we can define a **generalised case construct** to perform case-analysis on the observational equivalence classes of B

$$x' : A' \vdash \mathsf{gcase}^p_{B,C}(b, \{c_i\}_i) : C[b/y].$$

IdObs, obs and gcase^p will be defined through a mutual induction on the structure of the types that appear as their parameters.

We begin by defining IdObs. Let $x : A \vdash B\ \mathsf{type}$ and $x : A \vdash f, g : B$. We inductively (by structural induction on B) define $x : A \vdash \mathsf{IdObs}_B(f, g)\ \mathsf{type}$ as follows (and note that we have obvious subst operations defined on these observational identity types and the more general $\mathsf{let}\ p\ \mathsf{be}\ \mathsf{refl}_z\ \mathsf{in}\ -$ eliminators, as for the usual identity types, which are well-defined as long as the types we are substituting/eliminating in respect observational equivalence):

- for $B = 1$, define $\mathsf{IdObs}_1(\langle\rangle, \langle\rangle) := 1$;
- for $B = \Sigma_{y:E}D$, define $\mathsf{IdObs}_{\Sigma_{y:E}D}(\langle f, f'\rangle, \langle g, g'\rangle) := \Sigma_{p:\mathsf{IdObs}_E(f,g)}\mathsf{IdObs}_{D[g/y]}(\mathsf{subst}(p, f'), g')$;
- for $B = \Pi_{y:E}D$, define $\mathsf{IdObs}_{\Pi_{y:E}D}(f, g) := \Sigma_{p_1:\mathsf{IdObs}_{D[e_1/y]}(f(e_1),g(e_1))} \cdots$ $\Sigma_{p_{\#\mathsf{obs}(\Pi_{x:A}E)-1}:\mathsf{IdObs}_{D[e_{\#\mathsf{obs}(\Pi_{x:A}E)-1}/y]}(f(e_{\#\mathsf{obs}(\Pi_{x:A}C)-1}),g(e_{\#\mathsf{obs}(\Pi_{x:A}E)-1}))}\mathsf{IdObs}_{D[e_{\#\mathsf{obs}(\Pi_{x:A}E)}/y]}(f(e_{\#\mathsf{obs}(\Pi_{x:A}E)}), g(e_{\#\mathsf{obs}(\Pi_{x:A}E)}))$, where $\mathsf{obs}(\Pi_{x:A}E) = \{e_1, \dots, e_{\#\mathsf{obs}(\Pi_{x:A}E)}\}$ (which we define later);
- for $B = (e_i \mapsto_i \{b_{i,j} \mid j\})(e)$ and $x : A \vdash e : E$, we define $\mathsf{IdObs}_B(f, g) : \mathsf{Id}_B(f, g)$, i.e. for ground types, IdObs and Id agree.

The idea is that for a closed type B, $\mathsf{IdObs}_B(f, g)$ measures the observational equivalence (or extensional equality) of f and g, rather than their intensional equality $\vdash f \equiv g : B$, which $\mathsf{Id}_B(f, g)$ measures.

Note that the set $\mathsf{obs}(B)$ of representatives of observational equivalence classes of a finite inductive type B is equal to its set of constructors, while $\mathsf{obs}(\Sigma_{x:D}E) \cong \Sigma_{d\in\mathsf{obs}(D)}\mathsf{obs}(E[d/x])$ and, by the context lemma, $\mathsf{obs}(\Pi_{x:D}E) \cong \Pi_{d\in\mathsf{obs}(D)}\mathsf{obs}(E[d/x])$. In particular, we see that the number of observational equivalence classes at all types of our hierarchy is finite.

Let $\vdash A\ \mathsf{type}$, define $\mathsf{obs}(A)$ inductively on the structure of A:

- for $A = 1$, define $\mathsf{obs}(A) := \{\langle\rangle\}$;
- for $A = \Sigma_{y:E}D$, define $\mathsf{obs}(A) := \{\langle e_i, d_{i,j}\rangle \mid e_i \in \mathsf{obs}(E)\ \wedge\ d_{i,j} \in \mathsf{obs}(D[e_i/y])\}$;
- for $A = \Pi_{y:E}D$, define $\mathsf{obs}(A) := \{\lambda_{y:E}\mathsf{gcase}^p_{E,D}(y, \{\mathsf{subst}(p_i, d(e_i))\}_{e_i\in\mathsf{obs}(E)}) \mid d \in \Pi_{e_i\in\mathsf{obs}(E)}\mathsf{obs}(D[e_i/y])\}$ (note that this is well-defined as D respects observational equivalence, as it does not involve any Id-types in its construction);
- for $A = (e_i \mapsto_i \{b_{i,j} \mid j\})(e)$ with $\vdash e : E$, we can define $\mathsf{obs}(A) := \{b_{k,j} \mid \text{ s.t. } \vdash e \equiv e_k\}$ (note that such k are unique if they exists, because fibres are disjoint) .

Finally, we define $x' : A' \vdash \mathsf{gcase}^p_{B,C}(b, \{c_i\}_i) : C[b/y]$, given $\vdash A', B\ \mathsf{type}$, $x' : A', y : B \vdash C\ \mathsf{type}$, $x' : A' \vdash b : B$ and $\{x' : A', p_i : \mathsf{IdObs}_B(b_i, b) \vdash c_i : C[b/y]\}_{1\le i\le\#\mathsf{obs}(B)}$. We do this by structural induction on B:

- $\mathsf{gcase}^{p}_{1,C}(b,\{c\}) := c$;
- $\mathsf{gcase}^{p}_{\Sigma_{w:D}E,C}(b,\{c_{i,j}\}_{i,j}) := \mathsf{gcase}^{p^1}_{D,C[b/y]}(\mathsf{fst}(b),\{\mathsf{gcase}^{p^2}_{E[d_i/w],C[b/y]}(\mathsf{subst}((p^1_i)^{-1},\mathsf{snd}(b)),\{c_{i,j}\}_j)\}_i)$, where $\mathsf{obs}(D) = \{d_1,\dots,d_{\#\mathsf{obs}(D)}\}$ and where we write $(-)^{-1}$ for the usual groupoid inversion map $\Gamma, x:A, y:A, p:\mathsf{Id}_A(x,y) \vdash p^{-1}:\mathsf{Id}_A(y,x)$ which is obtained by applying the elimination rule for Id-types to the reflexivity witness;
- $\mathsf{gcase}^{p}_{\Pi_{w:D}E,C}(b,\{c_{\{i\overset{f}{\mapsto}j_i\}_i}\}_f) :=$
$$\begin{aligned}&\mathsf{gcase}^{p^1}_{E[d_1/w],C[b/y]}(b(d_1),\{\cdots\{\\ &\qquad\vdots\\ &\qquad\mathsf{gcase}^{p^{\#\mathsf{obs}(D)-1}}_{E[d_{\#\mathsf{obs}(D)-1}/w],C[b/y]}(b(d_{\#\mathsf{obs}(D)-1}),\{\\ &\qquad\mathsf{gcase}^{p^{\#\mathsf{obs}(D)}}_{E[d_{\#\mathsf{obs}(D)}/w],C[b/y]}(b(d_{\#\mathsf{obs}(D)}),\{\\ &\qquad c_f\\ &\qquad\}_{j_{\#\mathsf{obs}(D)}})\\ &\qquad\}_{j_{\#\mathsf{obs}(D)-1}})\\ &\qquad\vdots\\ &\}_{j_1});\end{aligned}$$
- for $B = (d_i \mapsto_i \{b_{i,j} \mid j\})(d)$ with $\vdash d : D$, we can define $\mathsf{gcase}^{p'}_{B,C}(b,\{c_j\}_j) := \mathsf{case}^{p,q}_{B,C}(b,\{e_{i,j}\}_{i,j})$, where
$$e_{i,j} := \begin{cases}\mathsf{subst}(p_i,\lambda_{p'_j}c_j)(q_{i,j}) & \text{if } \vdash d \equiv d_i : D\\ \mathsf{subst}(\mathsf{exfalso},\lambda_{p'_1}c_1)(q_{i,j}) & \text{else}\end{cases}$$
(where exfalso is well-defined as D is formed from 1, Σ-types and finite inductive type families).

8.4. Completeness Completed

Next, we complete the definability proof by showing how to define non-strict strategies from the syntax of DTT, using the devices introduced in section 8.3.

Theorem 8.5 (Full Completeness at Id-free type hierarchy)**.** *All morphisms in* $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi}}$ *are definable in* DTT*, where we write* $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi}}$ *for the full subcategory of* $\mathsf{Ctxt}(\mathbf{DGame}_!)$ *on the objects formed by the interpretation of types of* DTT *formed without* Id*-constructors.*

Proof. To show definability, by lemma 8.4, all that remains to be done is demonstrate definability for non-strict f.

If f is non-strict, we know from lemma 8.2 that f answers with some move a s.t. for all $\vdash t : T'$ s.t. $\vdash d^0[t/x] \equiv d[t/x]$, $[] \xrightarrow{a} [\![S[q[t/x]/x']]\!]$. Now, as S is a finite inductive type family, we know that $a = [\![s_0]\!]$ for some $\vdash s_0 : S[q_0/x']$ where we write $q_0 := q[t/x]$ (noting that $q[t/x]$ is independent of t as the fibres of S are disjoint and the interpretations of constructors of finite inductive types is faithful). Now, a term of the correct type to define f can be defined as
$$\dfrac{\dfrac{\dfrac{}{\vdash s_0 : S[q_0/x]}}{x:T', p':\mathsf{Id}_D(d^0,d) \vdash s_0 : S[q_0/x']}}{x:T', p':\mathsf{Id}_D(d^0,d) \vdash \mathsf{subst}(r,s_0) : S[q/x'],}$$
presuming we have some term $x:T', p':\mathsf{Id}_D(d^0,d) \vdash r : \mathsf{Id}_Q(q_0,q)$.

To construct r, we use our knowledge that $\vdash q_0 \equiv q[t_\alpha/x] : Q[t_\alpha/x]$ for all representatives $\vdash t_\alpha : T'$ of observational equivalence classes of T' such that $\vdash d^0[t_\alpha/x] \equiv d[t_\alpha/x] : D[t_\alpha/x]$. Define
$$\lambda_{p':\mathsf{Id}_D(d^0,d)}r := \mathsf{gcase}^{p}_{T',\Pi_{\mathsf{Id}_D(d^0,d)}\mathsf{Id}_Q(q_0,q)}\left(x,\left\{\begin{array}{ll}\mathsf{subst}(p_\alpha,\lambda_{p'}\mathsf{refl}_{q_0}) & \text{if } \vdash d[t_\alpha/x] \equiv d^0[t_\alpha/x] : D[t_\alpha/x]\\ \mathsf{subst}(p_\alpha,\mathsf{exfalso}) & \text{else}\end{array}\right\}_\alpha\right),$$

where we use a generalised case construct, as defined in section 8.3, (noting that $\Pi_{\mathsf{Id}_D(d^0,d)}\mathsf{Id}_Q(q_0,q)$ indeed respects observational equivalence, it follows that this definition type checks), to distinguish between the observational equivalence classes $t_1,\dots,t_{\#\mathsf{obs}(T')}$ of terms of T'. Noting that d^0 and $d[t/x]$ are closed terms of $D[t/x] = \Sigma_{w_1:E_1[f_1[t_\alpha/x]/x_1''']}\cdots\Sigma_{w_{N-1}:E_{N-1}[f_{N-1}[t_\alpha/\alpha]/x_{N-1}'']}E_N[f_N[t_\alpha/x]/x_N''']$ where $x''' : F_k \vdash E_k$ type is a finite inductive type family and $x : T' \vdash f_k : F_k$, we can use the exfalso eliminator of E_k-Discr to map out of $\mathsf{Id}_{E_k[f_k[t_\alpha/x]/x_k''']}(d_k^0, d_k[t_\alpha/x])$ if $\nvdash d^0 \equiv d[t_\alpha/x] : D[t_\alpha/x]$, as $d_k[t_\alpha/x]$ is judgementally equal to a constructor of E_k and d_k^0 already is a constructor of E_k by construction. □

We have obtained the following, combining theorems 8.1 and 8.5.

Corollary 8.6 (Full and Faithful Completeness at Id-free type hierarchy)**.** *All morphisms in* $\mathsf{Ctxt}(\mathbf{DGame}_!)^{\mathrm{fin1\Sigma\Pi}}$ *are faithfully definable in DTT.*

Next, we show that full (and faithful) completeness still holds if we allow one strictly positive occurrence[8] of an Id-type. This shows, in particular, that the notion of propositional identity coincides in syntax and semantics for open terms of the Id-free type hierarchy.

Theorem 8.7 (Full and Faithful Completeness for strictly positive Id-types)**.** *All morphisms in* $\mathsf{Ctxt}(\mathbf{DGame}_!)([],[\![\Pi_{x:A}\mathsf{Id}_B(f,g)]\!])$ *for* $x : A \vdash f, g : B$ *are faithfully definable in DTT, if* $\vdash A$ type *and* $x : A \vdash B$ type *are types built without* Id*-constructors.*

Proof. Faithfulness has already been argued in theorem 8.1.

Given an inhabitant p of $\mathsf{Ctxt}(\mathbf{DGame}_!)([\![1]\!],[\![\Pi_{x:A}\mathsf{Id}_B(f,g)]\!])$, for any $\cdot \vdash a : A$, evaluating p at $[\![a]\!]$ gives $p\{[\![a]\!]\} \in \mathsf{Ctxt}(\mathbf{DGame}_!)([\![1]\!],[\![\mathsf{Id}_B(f[a/x],g[a/x])]\!])$, which by (I3) of theorem 6.3 implies that $[\![f[a/x]]\!] = [\![f]\!]\{[\![a]\!]\} = [\![g]\!]\{[\![a]\!]\} = [\![g[a/x]]\!]$. Seeing that our model is faithful at the 1ΣΠ-hierarchy over finite inductive type families, we conclude that $\cdot \vdash f[a/x] \equiv g[a/x] : B$ for all $\cdot \vdash a : A$.

Using this information, we now construct a canonical inhabitant $x : A \vdash q_{A,B}^{f,g} : \mathsf{Id}_B(f,g)$. (Note that generally not $[\![q_{A,B}^{f,g}]\!] = p$.) We do this by performing a case distinction on the observational equivalence classes a_i of terms of A that x could belong to and then outputs $\vdash \mathsf{refl}_{f[a_i/x]} : \mathsf{Id}_{B[a_i/x]}(f[a_i/x], f[a_i/x]) \equiv \mathsf{Id}_{B[a_i/x]}(f[a_i/x], g[a_i/x])$. Indeed, we define $q_{A,B}^{f,g} := \mathsf{gcase}^p_{A,\mathsf{Id}_B(f,g)}(x, \{\mathsf{subst}(p_i, \mathsf{refl}_{f[a_i/x]})\}_i)$, using the generalised case construct of section 8.3 (noting that $\mathsf{Id}_B(f,g)$ respects observational equivalence, it follows that this definition type checks as it allows us to substitute along an element p_i of an observational identity type). This shows that completeness holds for types of the form $\Pi_{x:A}\mathsf{Id}_B(f,g)$.

Now, finally, we note that we can also interpret p as a morphism $q \in \mathsf{Ctxt}(\mathbf{DGame}_!)([],[\![\Pi_{x:A}B]\!])$. Theorem 8.5 now gives us $x : A \vdash h : B$ such that $[\![h]\!] = q$. Note that $x : A \vdash \mathsf{refl}_h : \mathsf{Id}_B(h,h)$ and, therefore, $x : A \vdash \mathsf{subst}(q^{\langle h,h\rangle,\langle f,g\rangle}_{A,\Sigma_{y:B}B}, \mathsf{refl}_h) : \mathsf{Id}_B(f,g)$. Moreover, we easily see that $[\![\mathsf{subst}(q^{\langle h,h\rangle,\langle f,g\rangle}_{A,\Sigma_{y:B}B}, \mathsf{refl}_h)]\!] = p$. This shows that p is definable in DTT and that we have, in fact, full completeness. □

Remark 8.8. *We would like to point out to the reader the phenomenon that (full) completeness at types involving positively occurring* Id*-type constructors crucially relies on faithfulness of the model. This is illustrated here for the case of one strictly positively occurring* Id*-type. It seems plausible that a full (and faithful) completeness result could be obtained similarly for more general types in which* Id*-types are allowed to occur positively.*

Remark 8.9. *The question rises what the status is of full completeness results for general types of DTT, in which* Id*-types are also allowed to occur negatively. One would expect to add* UIP *to the syntax, as this principle is, as far as we are aware, not derivable without universes for the types we consider. It seems believable that full completeness would hold if we only allow* Id*-types of ground types to occur negatively. For the fully general case, it seems likely that we would need to add a series of eliminators*

$$\frac{\nvdash a \equiv a' : A \qquad \vdash \mathsf{Id}_B(b,b')\ \mathsf{type}}{p : \mathsf{Id}_A(a,a') \vdash \mathsf{exfalso} : \mathsf{Id}_B(b,b').}$$

[8] Recall that we say that a subformula B occurs strictly positively in a type A if it does not appear as the antecedent of any function types. In particular, in the case of DTT, we say that B occurs strictly positively in A if it does not occur as the left hand side argument of a Π-type constructor.

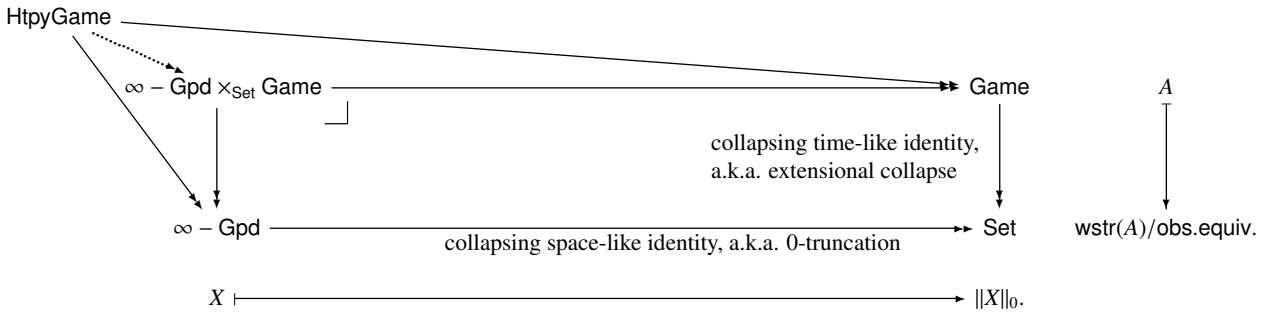
Otherwise, there is e.g. a non-strict strategy which responds tt *on* $[\![\Pi_{p:\mathsf{Id}_{\Pi_{x:\{a\}}\{a\}}(\lambda_{x:\{a\}}a,\lambda_{x:\{a\}}\mathsf{case}_{\{a\},\{a\}}(x,\{a\}))}\mathsf{Id}_{\{\mathsf{tt},\mathsf{ff}\}}(\mathsf{tt},\mathsf{ff})]\!]$ *which is not definable. One may question the desirability of such rules from a syntactic point of view, however.*

9. Future Work

Ultimately, the main goal is a thorough intensional, computational analysis of HoTT [4]. Obvious concrete directions for future work are the following:

- modifying the model to break UIP, while retaining the intensionality of function types (failure of function extensionality), in order to accommodate higher inductive types in the world of games - alternatively, pursuing the very closely related goal of developing a game semantics for quotient types[9];
- examining the phenomena of function extensionality and univalence;
- study of universes and a more intensional notion of type family;
- study of infinite (higher) inductive type families and their definability results, using the methods of [33];
- examining completeness properties of the model for the complete type hierarchy, including freely occurring Id-types;
- constructing models of DTT with effects and extending the current model to other evaluation strategies;
- synthesising strategies from a dependently typed specification;
- study of a possible embedding of the model in the co-Eilenberg-Moore category $\mathbf{Game}^!$, which might simplify its presentation.

In particular, to achieve the first item, we envisage a model based on a variation on nominal games. We propose to pursue a notion of what one might call a category of **homotopy games**, which should factor over the pullback of the **spatial and temporal extensional collapse** of the two models,



That is, we are looking for a setting to model DTT which combines the possibility of non-trivial propositional identity on ground types of the (∞-)groupoid model of DTT [3] with the failure of function extensionality of the game semantics. We hope this would not only result in a satisfactory game semantics for quotient types and higher inductive types, but would also give deeper insights into the subtle shades of intensionality that arise in dependent type theory, by cleanly separating out the time-like and space-like aspects of propositional identity.

[9] There is a conceptual challenge, here, of making sense of what such a game semantics for higher inductive or quotient types would mean. Indeed, quotient types and higher inductive types in their usual sense are both known to make the axiom of function extensionality derivable [19, 4]. One way to give meaning to this is to look for a game semantics whose extensional quotient supports higher inductive types or quotient types.

For the matter of models of DTT with effects, we should note in which ways the current model is robust with respect to a variation of the kinds of strategies considered. In general, rather than just modifying the notion of strategy to obtain the morphisms of our category, we should note that the definition of Π-games also refers to a particular choice of flavour of strategy (in our case, history-free well-bracketed deterministic winning strategies). This definition needs to be modified accordingly, if one would hope for dependent functions to compose.

On the one hand, the interpretation of the eliminator for Id-types relies crucially on the class of strategies we consider consisting only of maximal strategies. This means that their interpretation would not generalise straightforwardly to partial or non-deterministic strategies, meaning that our model does not straightforwardly extend to interpret fixpoint combinators or non-deterministic primitives [8, 14]. We feel there is a lesson to be learnt here about the nature of identity in settings of partial information and a further study is warranted.

On the other hand, the current proofs of soundness of the model do not rely in any way on the strategies we consider being well-bracketed or history-free. This suggests, in the light of [10, 11] that we could hope to extend the model to give a sound interpretation of variants of dependent type theory with respectively control operators and ground store or a combination of both.

In the world of games and weakly bracketed strategies, we note that although we can define control operators like call/cc for all simply typed games, the obvious candidate extensions of call/cc do not type check for all dependently typed context games. This means we appear to avoid the obstructions of [34] to classical dependent type theory in the sense of DTT with a call/cc-operator. Indeed, in particular, for the type $A = \Sigma_{x:\mathbb{B}} \mathsf{Id}_{\mathbb{B}}(\mathsf{tt}, x)$ used by Herbelin to derive his inconsistency result the obvious candidate for call/cc does not type check. Indeed, such an appropriate term $\mathsf{call/cc}_A$ of type $((A \Rightarrow \mathsf{Id}_{\mathbb{B}}(\mathsf{tt},\mathsf{ff})) \Rightarrow A) \Rightarrow A$ would decompose as $\langle \mathsf{call/cc}^1_A, \mathsf{call/cc}^2_A \rangle$, where $\vdash \mathsf{call/cc}^1_A : ((\Sigma_{x:\mathbb{B}} \mathsf{Id}_{\mathbb{B}}(\mathsf{tt}, x) \Rightarrow \mathsf{Id}_{\mathbb{B}}(\mathsf{tt},\mathsf{ff})) \Rightarrow \Sigma_{x:\mathbb{B}} \mathsf{Id}_{\mathbb{B}}(\mathsf{tt}, x)) \Rightarrow \mathbb{B}$ and $\vdash \mathsf{call/cc}^2_A : \Pi_{t:(\Sigma_{x:\mathbb{B}} \mathsf{Id}_{\mathbb{B}}(\mathsf{tt},x) \Rightarrow \mathsf{Id}_{\mathbb{B}}(\mathsf{tt},\mathsf{ff})) \Rightarrow \Sigma_{x:\mathbb{B}} \mathsf{Id}_{\mathbb{B}}(\mathsf{tt},x)} \mathsf{Id}_{\mathbb{B}}(\mathsf{tt}, \mathsf{call/cc}^1_A(t))$. The equivalent of the usual interpretation of call/cc of [10], which plays a copycat back and forth between $A^{(3)}$ and $A^{(2)}$, which copies the intial move in $A^{(3)}$ to $A^{(1)}$ if $\mathsf{Id}_{\mathbb{B}}(\mathsf{tt},\mathsf{ff})$ is opened by Opponent and which copies back Opponent's response in $A^{(1)}$ to $A^{(3)}$, is seen not to yield a sound interpretation of $\mathsf{call/cc}_A$ in our model. Indeed, $[\![\mathsf{call/cc}^2_A]\!]$ violates the rules of the game $\mathsf{O\text{-}sat}([\![\Pi_{t:(\Sigma_{x:\mathbb{B}} \mathsf{Id}_{\mathbb{B}}(\mathsf{tt},x) \Rightarrow \mathsf{Id}_{\mathbb{B}}(\mathsf{tt},\mathsf{ff})) \Rightarrow \Sigma_{x:\mathbb{B}} \mathsf{Id}_{\mathbb{B}}(\mathsf{tt},x)} \mathsf{Id}_{\mathbb{B}}(\mathsf{tt}, \mathsf{call/cc}^1_A(t))]\!])$. In particular, its play $*(0,*)(0,(0,*))(0,(0,(0,*)))(0,(0,(0,\mathsf{tt})))\mathsf{tt}$ does so with its last move. Indeed, this Player move excludes the value $\tau = [\![\lambda_{k:A \Rightarrow \mathsf{Id}_{\mathbb{B}}(\mathsf{tt},\mathsf{ff})} \langle \mathsf{ff}, k(\langle \mathsf{tt}, \mathsf{refl}_{\mathsf{tt}} \rangle) \rangle]\!]$ for $[\![t]\!]$, while only Opponent is allowed to restrict the fibre.

We note that a model of dependent types with ground store appears to follow immediately if we drop the history-freeness condition on strategies. The caveat (if we want to model Id-types) is that we have the work with a variant of ground store in which declared variables are automatically initialised to a default value, à la Java, in order to guarantee termination. It is not clear if a satisfactory generalisation to general references [12] should be expected, as their usual formulation allows one to define in particular a fixpoint combinator.

Another point to note is that we still obtain a sound model of DTT when we work with innocent strategies rather than history-free ones. This setting has the advantage that it is likely to provide more robust interpretations of various inductive types [33, 35].

Finally, we note that the option of a games model for call-by-value DTT along the lines of [17] seems plausible. Indeed, call-by-value evaluation could remove many of the technicalities of the current paper, which are caused by the requirement that dependent functions behave appropriately even when provided with underdetermined input. This is a complication that does not occur in the world of strict functions.

Acknowledgements

Samson Abramsky was supported by the EPSRC, AFOSR and the John Templeton Foundation. Radha Jagadeesan acknowledges support from the NSF. Matthijs Vákár was supported by the EPSRC and the Clarendon Fund. We are very grateful to the reviewers, whose reports were exceptionally meticulous and helpful.

References

[1] Altenkirch, T., McBride, C., McKinna, J.: Why dependent types matter. Manuscript, available at http://www.cs.nott.ac.uk/ txa/publ/ydtm.pdf (2005) 235

[2] McBride, C.: Faking it simulating dependent types in haskell. Journal of functional programming **12**(4-5) (2002) 375–392

[3] Awodey, S., Warren, M.A.: Homotopy theoretic models of identity types. Mathematical Proceedings of the Cambridge Philosophical Society **146**(01) (2009) 45–55

[4] HoTTbaki, U.: Homotopy Type Theory: Univalent Foundations of Mathematics. `http://homotopytypetheory.org/book`, Institute for Advanced Study (2013)

[5] Abramsky, S., Jagadeesan, R.: Games and full completeness for multiplicative linear logic. The Journal of Symbolic Logic **59**(02) (1994) 543–574

[6] Hyland, J.M.E., Ong, C.H.: On full abstraction for PCF: I, II, and III. Information and computation **163**(2) (2000) 285–408

[7] Nickau, H.: Hereditarily sequential functionals. In: Logical Foundations of Computer Science. Springer (1994) 253–264

[8] Abramsky, S., Jagadeesan, R., Malacaria, P.: Full abstraction for PCF. Information and Computation **163**(2) (2000) 409–470

[9] Abramsky, S., Jagadeesan, R.: A game semantics for generic polymorphism. Annals of Pure and Applied Logic **133**(1) (2005) 3–37

[10] Laird, J.: Full abstraction for functional languages with control. In: Logic in Computer Science, 1997. LICS'97. Proceedings., 12th Annual IEEE Symposium on, IEEE (1997) 58–67

[11] Abramsky, S., McCusker, G.: Linearity, sharing and state: a fully abstract game semantics for idealized algol with active expressions. Electronic Notes in Theoretical Computer Science **3** (1996) 2–14

[12] Abramsky, S., Honda, K., McCusker, G.: A fully abstract game semantics for general references. In: Logic in Computer Science, 1998. Proceedings. Thirteenth Annual IEEE Symposium on, IEEE (1998) 334–344

[13] Murawski, A.S., Tzevelekos, N.: Game semantics for good general references. In: Logic in Computer Science (LICS), 2011 26th Annual IEEE Symposium on, IEEE (2011) 75–84

[14] Harmer, R., McCusker, G.: A fully abstract game semantics for finite nondeterminism. In: Logic in Computer Science, 1999. Proceedings. 14th Symposium on, IEEE (1999) 422–430

[15] Danos, V., Harmer, R.S.: Probabilistic game semantics. ACM Transactions on Computational Logic (TOCL) **3**(3) (2002) 359–382

[16] Abramsky, S., Ghica, D.R., Murawski, A.S., Ong, C.H., Stark, I.D.: Nominal games and full abstraction for the nu-calculus. In: Logic in Computer Science, 2004. Proceedings of the 19th Annual IEEE Symposium on, IEEE (2004) 150–159

[17] Abramsky, S., McCusker, G.: Call-by-value games. In Nielsen, M., Thomas, W., eds.: Computer Science Logic. Volume 1414 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1998) 1–17

[18] Streicher, T.: Investigations into intensional type theory (1993) http://www.mathematik.tu-darmstadt.de/ streicher/HabilStreicher.pdf.

[19] Hofmann, M.: Extensional Constructs in Intensional Type Theory. Springer (1997)

[20] Bezem, M., Coquand, T., Huber, S.: A model of type theory in cubical sets. In: 19th International Conference on Types for Proofs and Programs (TYPES 2013). Volume 26. (2014) 107–128

[21] Palmgren, E., Stoltenberg-Hansen, V.: Domain interpretations of Martin-Löf's partial type theory. Annals of Pure and Applied Logic **48**(2) (1990) 135–196

[22] Hofmann, M., Streicher, T.: The groupoid interpretation of type theory. Twenty Five Years of Constructive Type Theory (1998)

[23] Abramsky, S.: Axioms for definability and full completeness. In: Proof, Language and Interaction: Essays in Honour of Robin. MIT Press (2000) 55–75

[24] Cartmell, J.: Generalised algebraic theories and contextual categories. Annals of Pure and Applied Logic **32** (1986) 209–243

[25] Pitts, A.M.: Categorical logic. In Abramsky, S., Gabbay, D., Maibaum, T., eds.: Handbook of Logic in Computer Science, Volume 5. OUP (2000) 39–128

[26] Dybjer, P.: Inductive families. Formal aspects of computing **6**(4) (1994) 440–465

[27] Abramsky, S., Jagadeesan, R.: Game semantics for access control. Electronic Notes in Theoretical Computer Science **249** (2009) 135–156

[28] McCusker, G.: Games and full abstraction for FPC. In: Logic in Computer Science, 1996. LICS'96. Proceedings., Eleventh Annual IEEE Symposium on, IEEE (1996) 174–183

[29] Vákár, M.: A categorical semantics for linear logical frameworks. In the proceedings of FoSSaCS 2015 (2015) Online version at http://arxiv.org/abs/1501.05016 .

[30] Carroll, L., Tenniel, J., Gardner, M.: The Annotated Alice. Penguin books (1965)

[31] Coquand, C.: A realizability interpretation of Martin-Löfs type theory. Twenty-Five Years of Constructive Type Theory (1998)

[32] Dybjer, P.: Internal type theory. In: Types for Proofs and Programs. Springer (1995) 120–134

[33] Abramsky, S., McCusker, G.: Games for Recursive Types. In Hankin, C., ed.: Theory and Formal Methods, Imperial College Press (1995) 1–20

[34] Herbelin, H.: On the degeneracy of Σ-types in presence of computational classical logic. In: Typed Lambda Calculi and Applications. Springer (2005) 209–220

[35] Clairambault, P.: Least and greatest fixpoints in game semantics. In: International Conference on Foundations of Software Science and Computational Structures, Springer (2009) 16–31