

Synthesizing nested relational queries from implicit specifications

Anonymized for Double-blind Reviewing

ABSTRACT

Derived datasets can be defined implicitly or explicitly. An *implicit definition* (of dataset O in terms of datasets \vec{I}) is a logical specification involving the source data \vec{I} and the interface data O . It is a valid definition of O in terms of \vec{I} , if any two models of the specification agreeing on \vec{I} agree on O . In contrast, an *explicit definition* is a query that produces O from \vec{I} . Variants of *Beth's theorem* [6] state that one can convert implicit definitions to explicit ones. Further, this conversion can be done effectively given a proof witnessing implicit definability in a suitable proof system. We prove the analogous effective implicit-to-explicit result for nested relations: implicit definitions, given in the natural logic for nested relations, can be effectively converted to explicit definitions in the nested relational calculus (NRC). As a consequence, we can effectively extract rewritings of NRC queries in terms of NRC views, given a proof witnessing that the query is determined by the views.

ACM Reference Format:

Anonymized for Double-blind Reviewing. 2022. Synthesizing nested relational queries from implicit specifications. In *Proceedings of ACM Conference (PODS '23)*. ACM, New York, NY, USA, 26 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

One way of describing a virtual datasource is via *implicit definition*: a specification Σ – e.g. in logic – involving symbols for the “virtual” object O and the stored “input” data \vec{I} . The specification may mention other data objects (e.g. auxiliary views). But to be an implicit definition, any two models of Σ that agree on \vec{I} must agree on O . In the case where Σ is in first-order logic, this hypothesis can be expressed as a first-order entailment, using two copies of the vocabulary, primed and unprimed, representing the two models:

$$\Sigma \wedge \Sigma' \wedge \bigwedge_{I_i \in \vec{I}} \forall \vec{x}_i [I_i(\vec{x}_i) \leftrightarrow I'_i(\vec{x}_i)] \models \forall \vec{x} [O(\vec{x}) \leftrightarrow O'(\vec{x})] \quad (\star)$$

Above Σ' is a copy of Σ with primed versions of each predicate.

A fundamental result in logic states that we can replace an implicit definition with an *explicit definition*: a first-order query Q such that whenever $\Sigma(\vec{I}, O, \dots)$ holds, $O = Q(\vec{I})$. The original result of this kind is *Beth's theorem* [6], which deals with classical first-order logic. Segoufin and Vianu's [32] looks at the case where Σ is in *active-domain first-order logic*, or equivalently a Boolean relational

algebra expression. Their conclusion is that one can produce an explicit definition of O over \vec{I} in relational algebra. [32] focused on the special case where $\Sigma(I_1 \dots I_j, \vec{B}, O)$ specifies each I_i as a view defined by an active-domain first-order formula φ_{V_i} over base data \vec{B} , and also defines O as an active-domain first-order query φ_Q over \vec{B} . In this case, Σ implicitly defining O in terms of \vec{I} is called “determinacy of the query by the views”. Segoufin and Vianu's result implies that *whenever a relational algebra query Q is determined by relational algebra views \vec{V} , then Q is rewritable over the views by a relational algebra query*.

Prior Beth-style results like [6, 32] are effective. From a proof of the entailment (\star) in a suitable proof system, one can extract an explicit definition effectively, even in polynomial time. While in early proofs of Beth's theorem, the proof systems were custom-designed for the task of proving implicit definitions, and the bounds were not stated. Later on standard proof systems such as tableaux [33] or resolution [19] were employed, and the polynomial claim was explicit. It is important that in our definition of implicit definability, we require the existence of a proof witness. By the completeness theorem for first-order logic, requiring such a proof witness is equivalent to demanding that implicit definability of O over \vec{I} holds for all instances, not just finite ones.

This paper deals with the situation for *nested relations*, a data model heavily explored in the database community. There is a natural analog of active domain first-order logic, suitable for implicit specification. These are the Δ_0 formulas, logical expressions where quantification is over elements within nested sets defined by terms. The notion of a Δ_0 specification $\Sigma(\vec{i}, o, \dots)$ implicitly defining nested relation o in terms of \vec{i} is the obvious one: for any two nested relations satisfying Σ , and agreeing on \vec{i} , they must agree on o . There is also a natural notion of proof witness for determinacy, using a proof system for Δ_0 formulas. The analog of relational algebra for explicit definitions is *nested relational calculus* NRC [40], which is the standard query language for nested relations. Our main result is:

From a proof p that Σ implicitly defines o in terms of \vec{i} , we can obtain, in PTIME, an NRC expression E that explicitly defines o from \vec{i} , relative to Σ .

A special case of this result concerns NRC views and queries. Our result implies that *if we have NRC views \vec{V} that determine an NRC query Q , then we can generate – from a suitable proof – an NRC rewriting of Q in terms of \vec{V}* .

The fact that such an NRC rewriting exists whenever there is a functional relationship was proven in [4]. But the argument was model-theoretic, and it was unclear how to obtain any algorithm for producing the NRC definition E from a proof.

Example 1.1. We consider the case where our specification $\Sigma(Q, V, B)$ describes a view V , a query Q , as well as some constraints on the base data B . Our base data B is of type $\text{Set}(\mathcal{U} \times \text{Set}(\mathcal{U}))$, where \mathcal{U}

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).
PODS '23, June 2023, Seattle, Washington, USA

© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

refers to the basic set of elements, the “Ur-elements”. That is, B is a set of pairs, where the first item is a data item and the second is a set of data items. View V is of type $\text{Set}(\mathcal{U} \times \mathcal{U})$, a set of pairs, given by the query that is the usual “flattening” of B : in NRC this can be expressed as $\{\langle \pi_1(b), c \rangle \mid c \in \pi_2(b) \mid b \in B\}$. The view definition can be converted to a specification in our logic.

A query Q might ask for a selection of the pairs in B , those whose first component is contained in the second: $\{b \in B \mid \pi_1(b) \in \pi_2(b)\}$. The definition of Q can also be incorporated into our specification.

View V is not sufficient to answer Q in general. This is the case if we assume as part of Σ an integrity constraint stating that the first component of B is a key. We can prove that $\Sigma(Q, V, B)$ implicitly defines Q in terms of V , and from this proof our algorithm can produce an NRC rewriting of Q in terms of V . ◀

Organization. We overview related work in Section 2 and provide preliminaries in Section 3. Section 4 presents our main result. It is proven in Section 6, making use of infrastructure from Section 5. We close with discussion in Section 7. Due to space constraints, many proofs are deferred to the appendix.

2 RELATED WORK

In addition to the theorems of Beth and Segoufin-Vianu mentioned in the introduction, there are numerous works on effective Beth-style results for other logics. Some concern fragments of classical first-order logic, such as the guarded fragment [5, 18]; others deal with non-classical logics such as description logics [36]. The Segoufin-Vianu result is closely related to variations of Beth’s theorem and Craig interpolation for relativized quantification, such as Otto’s interpolation theorem [30]. There are also effective interpolation and definability results for logics richer than or incomparable to first-order logic, such as fragments of fixpoint logics [2, 11]. There are even Beth-style results for full infinitary logic [23], but there one can not hope for effectivity. The connection between Beth-style results and view rewriting originates in [27, 32]. The idea of using effective Beth results to generate view rewritings from proofs appears in [13], and is explored in more detail first in [37] and later in [3].

Our main result relates to Beth theorems “up-to-isomorphism”. Our implicit definability hypothesis is that two models that satisfy a specification and agree on the inputs must agree on the output nested relations, where “agree on the output” means up to extensional equivalence of sets, which is a special (definable) kind of isomorphism. Beth-like theorems up to isomorphism originate in Gaifman’s [14] and are studied extensively by Hodges and his collaborators (e.g. [15–17]). The focus is model-theoretic, with emphasis on connections with categoricity and classification in classical model theory.

Our effective Beth-like theorem for nested relations extends two results in [4]. One is an ineffective result, which makes use of idea in [14]. but without any effectivity. Another is an effective result, but only for a restricted notion of “constructive proof”, which is not complete for classical logic.

3 PRELIMINARIES

Nested relations. We deal with schemas that describe objects of various *types* given by the following grammar.

$$T, U ::= \mathcal{U} \mid T \times U \mid \text{Unit} \mid \text{Set}(T)$$

For simplicity throughout the remainder we will assume only two basic types. There is the one-element type Unit , which will be used to construct Booleans. And there is \mathcal{U} , the “scalars” or *Ur-elements* whose inhabitants are not specified further. From the Ur-elements and a unit type we can build up the set of types via product and the power set operation. We use standard conventions for abbreviating types, with the n -ary product abbreviating an iteration of binary products. A *nested relational schema* consists of declarations of variable names associated to objects of given types.

Example 3.1. An example nested relational schema declares two objects $R : \text{Set}(\mathcal{U} \times \mathcal{U})$ and $S : \text{Set}(\mathcal{U} \times \text{Set}(\mathcal{U}))$. That is, R is a set of pairs of Ur-elements: a standard “flat” binary relation. S is a collection of pairs whose first elements are Ur-elements and whose second elements are sets of Ur-elements. ◀

The types have a natural interpretation. The unit type has a unique member and the members of $\text{Set}(T)$ are the sets of members of T . An *instance* of such a schema is defined in the obvious way.

For the schema in Example 3.1 above, assuming that $\mathcal{U} = \mathbb{N}$, one possible instance has $R = \{\langle 4, 6 \rangle, \langle 7, 3 \rangle\}$ and $S = \{\langle 4, \{6, 9\} \rangle\}$.

Δ_0 formulas. We need a logic appropriate for talking about nested relations. A natural and well-known subset of first-order logic formulas with a set membership relation are the Δ_0 formulas. They are built up from equality of Ur-elements via Boolean operators as well as relativized existential and universal quantification. All terms involving tupling and projections are allowed.

Formally, we deal with multi-sorted first-order logic, with sorts corresponding to each of our types. We use the following syntax for Δ_0 formulas and terms. Terms are built from variables using tupling and projections. All formulas and terms are assumed to be well-typed in the obvious way, with the expected sort of t and u being \mathcal{U} in expressions $t =_{\mathcal{U}} u$ and $t \neq_{\mathcal{U}} u$, and in $\exists t \in_T u \varphi$ the sort of t is T and the sort of u is $\text{Set}(T)$.

$$\begin{aligned} t, u &::= x \mid () \mid \langle t, u \rangle \mid \pi_1(t) \mid \pi_2(t) \\ \varphi, \psi &::= t =_{\mathcal{U}} t' \mid t \neq_{\mathcal{U}} t' \mid \top \mid \perp \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \\ &\quad \forall x \in_T t \varphi(x) \mid \exists x \in_T t \varphi(x) \end{aligned}$$

Note that there is no primitive negation, and no equalities for sorts other than \mathcal{U} . Negation $\neg\varphi$ will be defined as a macro by induction on φ by dualizing every connective. Other connectives can be derived in the usual way on top of negation: $\varphi \rightarrow \psi$ by $\neg\varphi \vee \psi$.

More crucial is the fact that Δ_0 formulas does not allow a membership atom. An *extended Δ_0 formula* allows also membership literal $x \in_T y$, $x \notin_T y$ at every type T .

The notion of an extended Δ_0 formula φ *entailing* another formula ψ is the standard one in first-order logic, meaning that every model of φ is a model of ψ . We emphasize here that by every model, we include models where membership is not extensional. An important point is that: *when φ and ψ are Δ_0 , rather than extended Δ_0 , “every model” can be replaced by “every nested relation”.* When we consider formulas that are Δ_0 , we write $\varphi \models_{\text{nested}} \psi$ for entailment.

The point above is due to two facts. First, we have neither \in or equality at higher types as an atomic predicate. This guarantees that any model can be modified, without changing the truth value of Δ_0 formulas, into a model satisfying extensionality: if we have x and y with $(\forall z \in_T x \ z \in_T y) \wedge (\forall z \in_T y \ z \in_T x)$ then x and y must be the same. Secondly, a well-typed extensional model is isomorphic to a nested relation, by the well-known Mostowski collapse construction that iteratively identifies elements that have the same members. The lack of primitive membership and equality relations in Δ_0 formulas allows us to avoid having to consider extensionality axioms, which would require special handling in our proof system.

Equality, inclusion and membership predicates “up to extensionality” may be defined as macros by induction on the involved types, while staying within Δ_0 formulas.

$$\begin{aligned} t \hat{\in}_T u &::= \exists z' \in u \ t \equiv_T z' \\ t \subseteq_T u &::= \forall z \in t \ z \hat{\in}_T u \\ t \equiv_{\text{Set}(T)} u &::= t \subseteq_T u \wedge u \subseteq_T t \\ t \equiv_{\text{Unit}} u &::= \top \quad t \equiv_{\mathbb{U}} u := t =_{\mathbb{U}} u \\ t \equiv_{T_1 \times T_2} u &::= \pi_1(t) \equiv_{T_1} \pi_1(u) \wedge \pi_2(t) \equiv_{T_2} \pi_2(u) \end{aligned}$$

We will use small letters for variables in Δ_0 formulas, except in examples when we sometimes use capitals to emphasize that an object is of set type. We drop the type subscripts T in bounded quantifiers, primitive memberships, and macros \equiv_T when clear. Of course membership-up-to-equivalence $\hat{\in}$ and membership \in agree on extensional models, and hence are not interchangeable on general models, and hence are not interchangeable in Δ_0 formulas. For example:

$$x \in y, x \in y' \models \exists z \in y \ z \in y'$$

But we do not have

$$x \hat{\in} y, x \hat{\in} y' \models \exists z \in y \ z \in y'$$

A set of primitive membership expressions $t \in u$ (i.e. extended Δ_0 formulas) will be called an \in -context.

Let us now introduce notation for instantiating a block of bounded quantifiers at a time. A *variable membership atom* is a membership atom $x \in y$ where x, y are variables. An *ordered variable \in -context* is a list of variable membership atoms.

Given a variable membership atom $x \in y$ and a Δ_0 formula of the form $\varphi_0 = \exists w \in y \ \varphi_1$, the *specialization of φ_0 using $x \in y$* is simply $\varphi_1[x/w]$. We generalize this to specializing φ_0 using an ordered variable \in -context $x_1 \in y_1 \dots x_i \in y_i$ by induction on i : when $\varphi_0 = \exists x_1 \in y_1 \ \varphi_1$, we first let φ_1 be the specialization of φ_0 using $x_1 \in y_1$ and then let φ_1' be the specialization of φ_1 using $x_2 \in y_2 \dots x_i \in y_i$, the latter given inductively. If φ_1 or φ_1' is not of the required form, then the specialization is not defined. A specialization of φ with respect to a variable \in -context is a specialization with respect to an ordering of some subset of the context. A *maximal specialization (max. spec.)* of φ_0 with respect to a \in -context is a specialization which is not existential leading. That is, no other variable membership can be applied to perform further specialization.

Nested Relational Calculus. We review the main language for declaratively transforming nested relations, Nested Relational Calculus (NRC). Variables occurring in expressions are typed, and each expression is associated with an *output type*, both of these being in the type system described above. We let Bool denote the type

$$\begin{aligned} E, E' ::= & \ x \mid () \mid \langle E, E' \rangle \mid \pi_1(E) \mid \pi_2(E) \mid \quad (\text{variable, (un)tupling}) \\ & \{E\} \mid \text{GET}_T(E) \mid \bigcup \{E \mid x \in E'\} \quad ((\text{un})\text{nesting, binding union}) \\ & \mid \emptyset \mid E \cup E' \mid E \setminus E' \quad (\text{finite unions, difference}) \end{aligned}$$

Figure 1: NRC syntax (typing rules omitted)

$\text{Set}(\text{Unit})$. Then Bool has exactly two elements, and will be used to simulate Booleans. The grammar for NRC expressions is presented in Figure 1.

The definition of the free and bound variables of an expression is standard, the union operator $\bigcup \{E \mid x \in R\}$ binding the variable x . The semantics of these expressions should be fairly evident, see [40]. If E has type T , and has input (i.e. free) variables $x_1 \dots x_n$ of types $T_1 \dots T_n$, respectively, then the semantics associates with E a function that given a binding associating each free variable with a value of the appropriate type, returns an object of type T . For example, the expression $()$ always returns the empty tuple, while \emptyset_T returns the empty set of type T .

The language NRC as originally defined cannot express certain natural transformations whose output type is \mathbb{U} . To get a canonical language for such transformations, above we included in our NRC syntax a family of operations $\text{GET}_T : \text{Set}(T) \rightarrow T$ that extracts the unique element from a singleton. GET was considered in [40]. The semantics are: if E returns a singleton set $\{x\}$, then $\text{GET}_T(E)$ returns x ; otherwise it returns some default object of the appropriate type. In [34], it is shown that GET is not expressible in NRC at sort \mathbb{U} . However, GET_T for general T is definable from $\text{GET}_{\mathbb{U}}$ and the other NRC constructs.

As explained in prior work (e.g. [40]), on top of the NRC syntax above we can support richer operations as “macros”. For every type T there is an NRC expression $=_T$ of type Bool representing equality of elements of type T . In particular, there is an expression $=_{\mathbb{U}}$ representing equality between \mathbb{U} -elements. For every type T there is an NRC expression \in_T of type Bool representing membership between an element of type T in an element of type $\text{Set}(T)$. We can define conditional expressions, joins, projections on k -tuples, and k -tuple formers. NRC is efficiently *closed under composition*: given $E(x, \dots)$ and $F(\vec{i})$ with output type matching the type of input variable x , we can form an expression $E(F)$ whose free variables are those of E other than x , unioned with those of F .

Finally, we note that NRC is closed under Δ_0 *comprehension*: if E is in NRC, φ is a Δ_0 formula, then we can efficiently form an expression $\{z \in E \mid \varphi\}$ which returns the subset of E such that φ holds. We make use of these macros freely in our examples of NRC, such as Example 1.1.

Connections between NRC queries using Δ_0 formulas. Given an NRC expression E with input relations \vec{i} , we can create a Δ_0 formula $\Sigma_E(\vec{i}, o)$ that is an *input-output specification of E* : a formula such that Σ_E implies $o = E(\vec{i})$ and whenever $o = E(\vec{i})$ holds there is an expansion with Σ_E holding. For the “composition-free” fragment in which comprehensions \bigcup can only be over input variables – this conversion can be done in PTIME. But it cannot be done efficiently for general NRC, under complexity-theoretic hypotheses [20].

We also write entailments that use NRC expressions, e.g. $\varphi(x, \vec{c}, \dots) \models_{\text{nested}} x \in E(\vec{c})$ for $\varphi \Delta_0$ and $E \in \text{NRC}$. An entailment with \models_{nested} involving NRC expressions means that in every *nested*

relation satisfying φ , x is in the output of E on \vec{c} . Note that the semantics of NRC expressions is only defined on nested relations.

4 IMPLICIT VS EXPLICIT AND THE STATEMENT OF THE MAIN RESULT

We now formalize our implicit-to-explicit result. A Δ_0 formula $\varphi(\vec{i}, \vec{a}, o)$ *implicitly defines variable o in terms of variables \vec{i} up to extensionality* if we have

$$\varphi(\vec{i}, \vec{a}, o) \wedge \varphi(\vec{i}, \vec{a}', o') \vdash_{\text{nested}} o \equiv_T o' \quad (\star)$$

Recall that \equiv_T is equivalence-modulo-extensionality. It can be replaced by equality if we add extensionality axioms on the left of the entailment symbol. φ will be called an *implicit definition up to extensionality* of o in terms of \vec{i} .

An NRC expression E using free variables in \vec{i} *explicitly defines o up to extensionality relative to Δ_0 formula $\varphi(\vec{i}, \vec{a}, o)$* if for every model of φ , E applied to \vec{i} produces o' with $o' \equiv_T o$. Assuming extensionality, the conclusion is equivalent to $o' = o$.

In [4], it was shown that implicit Δ_0 definitions can be converted to NRC definitions:

THEOREM 1. Δ_0 formula $\varphi(\vec{i}, \vec{a}, o)$ *implicitly defines o with respect to \vec{i} up to extensionality if and only if there is an NRC expression $E(\vec{i})$ that explicitly defines o up to extensionality relative to φ .*

We explain how Theorem 1 implies Segoufin and Vianu's [32] result for relational algebra. Suppose $\Sigma(\vec{I}, O, \dots)$ is a single-sorted first-order logic formula over predicates that include $\vec{I} \cup \{O\}$, using only *active domain quantification* - quantification over the union of projections of predicates - and suppose that any two models of Σ that agree on \vec{I} agree on O . Such a Σ can be considered a special kind of Δ_0 formula, and the hypothesis implies that O is implicitly defined by \vec{I} relative to Σ . Our conclusion is that there is an NRC expression that produces O from \vec{I} . We now use well-known results about the “conservativity” of NRC over relational algebra for set-to-set transformation [31, 39, 40]: NRC expressions transforming relations to relations can be converted to relational algebra expressions.

Proof systems for Δ_0 formulas. Our main result is an effective version of Theorem 1. For this we need to formalize our proof system for Δ_0 formulas, which will allow us to talk about proof witnesses for implicit definability.

If we want to talk only about *effective generation* of NRC witnesses from proofs, we can use a basic proof system for Δ_0 formulas, whose inference rules are shown in Figure 2.

The node labels are a variation of the traditional rules for first-order logic, with a couple of quirks related to the specifics of Δ_0 formulas. Each node label has shape $\Theta; \Gamma \vdash \Delta$ where

- Θ is an \in -context. Recall that these are sets of membership atoms — the only formulas in our proof system that are extended Δ_0 but not Δ_0 . They will emerge during proofs involving Δ_0 formulas when we start breaking down bounded quantifier formulas.
- Γ and Δ are finite sets of Δ_0 formulas.

For example, REFL in the figure is a “congruence rule”, capturing that terms that are equal are interchangeable. Informally, it says that to prove conclusion Δ from a hypothesis that includes a formula φ

$$\begin{array}{c}
 \text{Ax} \frac{}{\Theta; \Gamma, \varphi \vdash \varphi, \Delta} \qquad \text{!-L} \frac{}{\Theta; \Gamma, \perp \vdash \Delta} \\
 \\
 \text{!-R} \frac{\Theta; \Gamma \vdash \neg \varphi, \Delta}{\Theta; \Gamma, \varphi \vdash \Delta} \qquad \text{!-R} \frac{\Theta; \Gamma, \varphi \vdash \Delta}{\Theta; \Gamma \vdash \neg \varphi, \Delta} \\
 \\
 \text{!-R} \frac{\Theta; \Gamma \vdash \varphi_1, \Delta \quad \Theta; \Gamma \vdash \varphi_2, \Delta}{\Theta; \Gamma \vdash \Delta, \varphi_1 \wedge \varphi_2} \qquad \text{!-R} \frac{\Theta; \Gamma \vdash \varphi_1, \varphi_2, \Delta}{\Theta; \Gamma \vdash \varphi_1 \vee \varphi_2, \Delta} \\
 \\
 \text{!-R} \frac{\Theta, y \in b; \Gamma \vdash \varphi[y/x], \Delta}{\Theta; \Gamma \vdash \forall x \in b \varphi, \Delta} \quad y \text{ FRESH} \\
 \\
 \text{!-R} \frac{\Theta, t \in b; \Gamma \vdash \varphi[t/x], \exists x \in b \varphi, \Delta}{\Theta, t \in b; \Gamma \vdash \exists x \in b \varphi, \Delta} \\
 \\
 \text{REFL} \frac{\Theta; \Gamma, t =_{\mathbb{U}} t \vdash \Delta}{\Theta; \Gamma \vdash \Delta} \qquad \text{REPL} \frac{\Theta; \Gamma, t =_{\mathbb{U}} u, \varphi[u/x], \varphi[t/x] \vdash \Delta}{\Theta; \Gamma, t =_{\mathbb{U}} u, \varphi[t/x] \vdash \Delta} \\
 \\
 \times_{\eta} \frac{\Theta[\langle x_1, x_2 \rangle / x]; \Gamma[\langle x_1, x_2 \rangle / x] \vdash \Delta[\langle x_1, x_2 \rangle / x]}{\Theta; \Gamma \vdash \Delta} \quad x_1, x_2 \text{ FRESH} \\
 \\
 \times_{\beta} \frac{\Theta[x_i / x]; \Gamma[x_i / x] \vdash \Delta[x_i / x]}{\Theta[\pi_i(\langle x_1, x_2 \rangle) / x]; \Gamma[\pi(\langle x_1, x_2 \rangle) / x] \vdash \Delta[\pi_i(\langle x_1, x_2 \rangle) / x]} \quad i \in \{1, 2\}
 \end{array}$$

Figure 2: Proof rules for a Δ_0 calculus, without restrictions for efficient generation of witnesses.

including variable t and an equality $t =_{\mathbb{U}} u$, it suffices to add to the hypotheses a copy of φ with u replacing some occurrences of t .

A proof tree whose root is labelled by $\Theta; \Gamma \vdash \Delta$ witnesses that, for any given meaning for the free variables, if all the membership relations in Θ and all formulas in Γ are satisfied, then there is a formula in Δ which is true. We say that we have a proof of a single formula φ when we have a proof of $\emptyset; \emptyset \vdash \varphi$.

The proof system is easily seen to be sound: if $\Theta; \Gamma \vdash \Delta$, then $\Theta; \Gamma \models \Delta$, where we remind the reader that \models considers all models, not just extensional ones. It can be shown to be complete by a standard technique (a “Henkin construction”, see the appendix).

To generate NRC definitions *efficiently* from proof witnesses will require a more restrictive proof system, in which we enforce some ordering on how proof rules can be applied, depending on the shape of the hypotheses. We refer to proofs in this system as *focused proofs*, the terminology being inspired by the proof search literature [26]. To this end, we categorize formulas as being either *existential-leading* (EL) or *alternative-leading* (AL) according to their top-level connective. Only atomic formulas are both EL and AL, and the only other EL formulas are existentials; all the others are AL.¹

$$\begin{array}{ll}
 \varphi^{\text{EL}} & := \quad t =_{\mathbb{U}} u \mid t \neq_{\mathbb{U}} u \mid \exists x \in t. \psi \\
 \varphi^{\text{AL}} & := \quad t =_{\mathbb{U}} u \mid t \neq_{\mathbb{U}} u \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \top \mid \perp \mid \forall t \in b. \psi
 \end{array}$$

¹In the literature on focusing, these are referred to as “positive” and “negative” formulas, but we avoid this terminology due to clashes with other uses of those terms.

$$\begin{array}{c}
= \frac{}{\Theta \vdash x =_{\mathcal{U}} x, \Delta} \quad \top \frac{}{\Theta \vdash \top, \Delta} \\
\neq \frac{\Theta \vdash t \neq_{\mathcal{U}} u, \alpha[u/x], \alpha[t/x], \Delta^{\text{EL}} \quad \alpha \text{ atomic}}{\Theta \vdash t \neq_{\mathcal{U}} u, \alpha[t/x], \Delta^{\text{EL}}} \\
\wedge \frac{\Theta \vdash \varphi_1, \Delta \quad \Theta \vdash \varphi_2, \Delta}{\Theta \vdash \varphi_1 \wedge \varphi_2, \Delta} \quad \vee \frac{\Theta \vdash \varphi_1, \varphi_2, \Delta}{\Theta \vdash \varphi_1 \vee \varphi_2, \Delta} \\
\vee \frac{\Theta, y \in b \vdash \varphi[y/x], \Delta \quad y \text{ fresh}}{\Theta \vdash \forall x \in b. \varphi, \Delta} \\
\exists \frac{\Theta \vdash \varphi', \varphi, \Delta^{\text{EL}} \quad \varphi' \text{ a max. spec. of } \varphi \text{ w.r.t. } \Theta}{\Theta \vdash \varphi, \Delta^{\text{EL}}} \\
\times_{\eta} \frac{\Theta[\langle x_1, x_2 \rangle / x] \vdash \Delta^{\text{EL}}[\langle x_1, x_2 \rangle / x] \quad x_1, x_2 \text{ fresh}}{\Theta \vdash \Delta^{\text{EL}}} \\
\times_{\beta} \frac{\Theta[x_i/x] \vdash \Delta^{\text{EL}}[x_i/x] \quad i \in \{1, 2\}}{\Theta[\pi_i(\langle x_1, x_2 \rangle) / x] \vdash \Delta^{\text{EL}}[\pi_i(\langle x_1, x_2 \rangle) / x]}
\end{array}$$

Figure 3: Our focused calculus for efficient generation of witnesses. We assume that formulas φ^{AL} (respectively contexts Δ^{EL}) are AL (respectively contain only EL formulas) and that x, y, z are variables.

Our focused proof system is shown in Figure 3. A superficial difference from Figure 2 is that the focused system is “almost 1-sided”: Δ_0 formulas only occur on the right, with only \in -contexts on the left. In particular, a top-level goal $\Theta; \Gamma \vdash \Delta$ in the higher-level system would be expressed as $\Theta \vdash \neg \Gamma, \Delta$ in this system. We will often abuse notation by referring to focused proofs of a 2-sided sequent $\Theta; \Gamma \vdash \Delta$, considering them as “macros” for the corresponding 1-sided sequent. For example, the hypothesis of the \neq rule could be written in 2-sided notation as $\Theta, t =_{\mathcal{U}} u \vdash \alpha[u/x], \alpha[t/x], \Delta^{\text{EL}}$ while the conclusion could be written as $\Theta, t =_{\mathcal{U}} u \vdash \alpha[t/x], \Delta^{\text{EL}}$. As with REPL in the prior system, this rule is about duplicating a hypothesis with some occurrences of t replaced by u .

A major aspect of the restriction, related to the terminology *focused*, is that the \exists -R rule enforces that blocks of existentials are instantiated all at once and that all other formulas in the context are also EL.

Soundness is evident, since it is a special case of the proof system above. Completeness is not as obvious, since we are restricting the proof rules. But we can translate proofs in the more general system of Figure 2 into a focused proof, but with an exponential blow-up: see the appendix for details.

Furthermore, since for Δ_0 formulas equivalence over all structures is the same as equivalence over nested relations, a Δ_0 formula φ is provable exactly when $\models_{\text{nested}} \varphi$.

Example 4.1. Let us look at how to formalize a variation of Example 1.1. The specification $\Sigma(B, V)$ includes two conjuncts $C_1(B, V)$

and $C_2(B, V)$. $C_1(B, V)$ states that every pair $\langle k, e \rangle$ of V corresponds to a $\langle k, S \rangle$ in B with $k \in S$:

$$\forall v \in V \exists b \in B. \pi_1(v) =_{\mathcal{U}} \pi_1(b) \wedge \pi_2(v) \hat{=} \pi_2(b)$$

$C_2(B, V)$ is:

$$\forall b \in B \forall e \in \pi_2(b) \exists v \in V. \pi_1(v) =_{\mathcal{U}} \pi_1(b) \wedge \pi_2(v) =_{\mathcal{U}} e$$

Let us assume a stronger constraint, $\Sigma_{\text{lossless}}(B)$, saying that the first component is a key and second is non-empty:

$$\begin{aligned} \forall b \in B \forall b' \in B. \pi_1(b) =_{\mathcal{U}} \pi_1(b') \rightarrow b \equiv b' \\ \wedge \forall b \in B \exists e \in \pi_2(b). \top \end{aligned}$$

With Σ_{lossless} we can show something stronger than in Example 1.1: $\Sigma \wedge \Sigma_{\text{lossless}}$ implicitly defines B in terms of V . That is, the view determines the identity query, which is witnessed by a proof of

$$\Sigma(B, V) \wedge \Sigma_{\text{lossless}}(B) \wedge \Sigma(B', V) \wedge \Sigma_{\text{lossless}}(B') \rightarrow B \equiv B'$$

Let’s prove this informally. Assuming the premise, it is sufficient to prove $B \subseteq B'$ by symmetry. So fix $\langle k, S \rangle \in B$. By the second conjunct of $\Sigma_{\text{lossless}}(B)$, we know there is $e \in S$. Thus by $C_2(B, V)$, V contains the pair $\langle k, e \rangle$. Then, by $C_1(B', V)$, there is a S' such that $\langle k, S' \rangle \in B'$. To conclude it suffices to show that $S \equiv S'$. There are two similar directions, let us detail the inclusion $S \subseteq S'$; so fix $s \in S$. By $C_2(B, V)$, we have $\langle k, s \rangle \in V$. By $C_1(B', V)$ there exists S'' such that $\langle k, S'' \rangle \in B'$ with $s \in S''$. But since we also have $\langle k, S' \rangle \in B'$, the constraint $\Sigma_{\text{lossless}}(B)$ implies that $S' \equiv S''$, so $s \in S'$ as desired. \triangleleft

Main result. A derivation of (\star) in our proof system will be referred to as a *witness* to the implicit definability of o in terms of \vec{i} up to extensionality. With these definitions, we now state formally our main result, the effective version of Theorem 1:

THEOREM 2 (EFFECTIVE IMPLICIT TO EXPLICIT FOR NESTED DATA). *Given a witness for an implicit definition of o in terms of \vec{i} up to extensionality relative to Δ_0 $\varphi(\vec{i}, \vec{a}, o)$, one can compute NRC expression E such that for any \vec{i}, \vec{a} and o , if $\varphi(\vec{i}, \vec{a}, o)$ then $E(\vec{i}) = o$. Furthermore, if the witness is focused, this can be done in polynomial time.*

Application to views and queries. We have a consequence for rewriting queries over views. Consider a query given by NRC expression E_Q over inputs \vec{B} and NRC expressions $E_{V_1} \dots E_{V_n}$ over \vec{B} . E_Q is *determined* by $E_{V_1} \dots E_{V_n}$, if every two nested relations (finite or infinite) interpreting \vec{B} that agree on the output of each E_{V_i} agree on the output of E_Q . An NRC rewriting of E_Q in terms of $E_{V_1} \dots E_{V_n}$ is an expression $R(V_1 \dots V_n)$ such that for any nested relation \vec{B} , if we evaluate each E_{V_i} on \vec{B} to obtain V_i and evaluate R on the resulting $V_1 \dots V_n$, we obtain $Q(\vec{B})$.

Given E_Q and $E_{V_1} \dots E_{V_n}$, let $\Sigma_{\vec{V}, Q}(\vec{V}, \vec{B}, Q, \dots)$ conjoin the input-output specifications, as defined in Section 3, for $E_{V_1} \dots E_{V_n}$ and E_Q . This formula has variables $\vec{B}, V_1 \dots V_n, Q$ along with auxiliary variables for subqueries. A proof witnessing determinacy of E_Q by $E_{V_1} \dots E_{V_n}$, is a proof that $\Sigma_{\vec{V}, Q}$ implicitly defines Q in terms of \vec{V} .

COROLLARY 3. *From a witness that a set of NRC views \vec{V} determines an NRC query Q , we can produce an NRC rewriting of Q in terms of \vec{V} . If the witness is focused, this can be done in PTIME.*

The notion of determinacy of a query over views relative to a Δ_0 theory (e.g. the key constraint in Example 1.1) is a straightforward generalization of the definitions above, and Corollary 3 extends to this setting.

In the case where we are dealing with flat relations, the effective version is well-known: see Toman and Weddell's [37], and the discussion in [3, 13].

We emphasize that the result involves equivalence up to extensionality, which underlines the distinction from the classical Beth theorem. If we wrote out implicit definability up to extensionality as an entailment involving two copies of the signature, we would run into problems in applying the standard proof of Beth's theorem.

5 TOOLS FOR THE MAIN THEOREM

Interpolation. The first tool for our main theorem will be an interpolation result. Informally, such results say that if we have an entailment involving two formulas, a “left” formula φ_L and a “right” formula φ_R , we can get an “explanation” for the entailment that factors through an expression only involving non-logical symbols (in our case, variables) that are common to φ_L and φ_R .

THEOREM 4. *Let Θ be an \in -context and Γ, Δ finite sets of Δ_0 formulas. Then from any proof of $\Theta; \Gamma \vdash \Delta$, we can compute in linear time a Δ_0 formula θ with $FV(\theta) \subseteq FV(\Theta, \Gamma) \cap FV(\Delta)$, such that $\Theta; \Gamma \vdash \theta$ and $\theta \vdash \Delta$*

The θ produced by the theorem is a *Craig interpolant*. Craig's interpolation theorem [9] states that when $\Gamma \vdash \Delta$ with Γ, Δ in first-order logic, such a θ exists in first-order logic. Our variant states one can find θ in Δ_0 efficiently from a proof of the entailment in either of our Δ_0 proof systems. We have stated the result for the 2-sided system. It holds also for the 1-sided focused system, where the partition of the formulas into left and right of the proof symbol is arbitrary. The argument is induction on proof length, roughly following prior interpolation algorithms [33].

Some admissible rules. As we mentioned earlier, our focused proof system is extremely low-level, and so it is convenient to have higher-level proof rules as macros. We formalize this below.

Definition 5. A rule with premise $\Theta' \vdash \Delta'$ and conclusion $\Theta \vdash \Delta$

$$\frac{\Theta' \vdash \Delta'}{\Theta \vdash \Delta}$$

is (polytime) *admissible* in a given calculus if a proof of the conclusion $\Theta \vdash \Delta$ in that calculus can be computed from a proof of the premise $\Theta' \vdash \Delta'$ (in polynomial time).

Up to rewriting the sequent to be one-sided, all the rules in Figure 2 are polytime admissible in the focused calculus. Our main theorem will rely on the polytime admissibility within the focused calculus of additional rules that involve chains of existential quantifiers. To state them, we need to introduce a generalization of bounded quantification: “quantifying over subobjects of a variable”. For every type T , define a set of words over the three-letter alphabet $\{1, 2, m\}$ of *subtype occurrences of T* inductively as follows:

- The empty word ε is a subtype occurrence of any type
- If p is a subtype occurrence of T , mp is a subtype occurrence of $\text{Set}(T)$.

- If $i \in \{0, 1\}$ and p is a subtype occurrence of T_i , ip is a subtype occurrence of $T_1 \times T_2$.

Given subtype occurrence p and quantifier symbol $Q \in \{\forall, \exists\}$, define the notation $Q x \in_p t.\varphi$ by induction on p :

- $Q x \in_m t.\varphi$ is $Q x \in t$
- $Q x \in_{mp} t.\varphi$ is $Q y \in t.Q x \in_p y.\varphi$ with y a fresh variable
- $Q x \in_{ip} t.\varphi$ is $Q x \in_p \pi_i(t).\varphi$ when $i \in \{1, 2\}$.

Now we are ready to state the results we need on admissibility in the body of the paper, referring in each case to the focused calculus. Some further routine rules are used in the appendices. The first states that if we have proven that there exists a subobject of o' equivalent to object r , then we can prove that for each element z of r there is a corresponding equivalent subobject z' within o' .

LEMMA 6. *Assume p is a subtype occurrence for the type of the term o' . The following is polytime admissible*

$$\frac{\Theta \vdash \Delta, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}{\Theta, z \in r \vdash \Delta, \exists z' \in_{mp} o'. z \equiv_{T'} z'}$$

Furthermore, the size of the output proof is at most the size of the input proof.

The second rule states that we can move between an equivalence of r, r' and a universally-quantified biconditional between memberships in r and r' . Because we are dealing with Δ_0 formulas, the universal quantification has to be bounded by some additional variable a .

LEMMA 7. *The following is polytime admissible (where p is a subtype occurrence of the type of o')*

$$\frac{\Theta \vdash \Delta, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}{\Theta \vdash \Delta, \exists r' \in_p o'. \forall z \in a. z \in r \leftrightarrow z \in r'}$$

The NRC Parameter Collection Theorem. Our last tool is a kind of interpolation result connecting Δ_0 formulas and NRC:

THEOREM 8 (NRC PARAMETER COLLECTION). *Let L, R be sets of variables with $C = L \cap R$ and*

- φ_L and $\lambda(z)$ Δ_0 formulas over L
- φ_R and $\rho(z, y)$ Δ_0 formulas over R
- r a variable of R and c a variable of C .

Suppose that we have a proof of

$$\varphi_L \wedge \varphi_R \rightarrow \exists y \in_p r \forall z \in c (\lambda(z) \leftrightarrow \rho(z, y))$$

Then one may compute in polynomial time an NRC expression E with free variables in C such that

$$\varphi_L \wedge \varphi_R \rightarrow \{z \in c \mid \lambda(z)\} \in E$$

If λ was a “common formula” — one using only variables in C — then the nested relation $\{z \in c \mid \lambda(z)\}$ would be definable over C in NRC via Δ_0 -comprehension. Unfortunately λ is a “left formula”, possibly with variables outside of C . Our hypothesis is that it is equivalent to a “parameterized right formula”: a formula with variables in R and parameters that lie below them. Intuitively, this can happen only if λ can be rewritten to a formula $\rho'(z, x)$ with variables of C and a distinguished $c_0 \in C$ such that

$$\varphi_L \wedge \varphi_R \rightarrow \exists x \in_p c_0 \forall z \in c (\lambda(z) \leftrightarrow \rho'(z, x))$$

And if this is true, we can use an NRC expression over C to define a set that will contain the correct “parameter” value x defining λ . From this we can define a set containing the nested relation $\{z \in c \mid \lambda(z)\}$. A formalization of this rough intuition — “when left formulas are equivalent to parameterized right formulas, they are equivalent to parameterized common formulas” — can be found in the appendix.

Sketch of the proof of Theorem 8. To get the desired conclusion, we need to prove a more general statement by induction over proof trees. Besides making the obvious generalization to handle two sets of formulas instead of the particular formulas φ_L and φ_R , as well as some corresponding left and right \in -contexts, that may appear during the proof, we need to additionally generate a new formula θ that only uses common variables, which can replace φ_R in the conclusion. This is captured in the following lemma:

LEMMA 9. *Let L, R be sets of variables with $C = L \cap R$ and*

- $\Delta_L, \lambda(z)$ *a set of Δ_0 formulas over L*
- $\Delta_R, \rho(z, y)$ *a set of Δ_0 formulas over R*
- Θ_L (respectively Θ_R) *a \in -context over L (respectively over R)*
- r *a variable of R and c a variable of C .*

Suppose that we have a proof tree with conclusion

$$\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \exists y \in_p r \forall z \in c (\lambda(z) \leftrightarrow \rho(z, y))$$

Then one may compute in polynomial time an NRC expression E and a Δ_0 formula θ using only variables from C such that

$$\Theta_L \models_{\text{nested}} \Delta_L, \theta \vee \{z \in c \mid \lambda(z)\} \in E \quad \text{and} \quad \Theta_R \models_{\text{nested}} \Delta_R, \neg \theta$$

In this induction over the size of the proof of

$$\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \exists y \in_p r \forall z \in c (\lambda(z) \leftrightarrow \rho(z, y))$$

we make a case distinction according to which rule is applied last. Many of the cases use standard techniques, we focus in the body of the paper on the most novel case.

Let us write \mathcal{G} for the formula $\exists y \in_p r \forall z \in c. (\lambda(z) \leftrightarrow \rho(z, y))$ and Λ for the set $\{z \in c \mid \lambda(z)\}$. The most difficult inductive case is where the last rule applied is \exists , and where \mathcal{G} is the main formula, i.e., when the last step, where we pick some witness w for y using the \exists rule, has shape

$$\frac{\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \forall z \in c. (\lambda(z) \leftrightarrow \rho(z, w)), \mathcal{G}}{\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \mathcal{G}}$$

Now notice that, due to our restriction on the \exists rule, all formulas in Δ_L and Δ_R are EL. Therefore, the only possible shape of the proof, when reasoning backward from the goal, is via successive applications of the \forall, \wedge rules. This means we have two strict subproofs with respective conclusions

$$\begin{aligned} & \Theta_L, z \in c, \Theta_R \vdash \lambda(z), \Delta_L, \neg \rho(z, w), \Delta_R, \mathcal{G} \\ \text{and} \quad & \Theta_L, z \in c, \Theta_R \vdash \neg \lambda(z), \Delta_L, \rho(z, w), \Delta_R, \mathcal{G} \end{aligned}$$

Applying the inductive hypothesis, we obtain NRC expressions E_1^{IH} , E_2^{IH} and formulas $\theta_1^{\text{IH}}, \theta_2^{\text{IH}}$ which contain free variables in $C \cup \{z\}$

such that all of the following hold

$$\begin{aligned} & \Theta_L, z \in c \models_{\text{nested}} \lambda(z), \Delta_L, \theta_1^{\text{IH}} \vee \Lambda \in E_1^{\text{IH}} \\ \text{and} \quad & \Theta_L, z \in c \models_{\text{nested}} \neg \lambda(z), \Delta_L, \theta_2^{\text{IH}} \vee \Lambda \in E_2^{\text{IH}} \\ \text{and} \quad & \Theta_R \models_{\text{nested}} \neg \rho(z, w), \Delta_R, \neg \theta_1^{\text{IH}} \\ \text{and} \quad & \Theta_R \models_{\text{nested}} \rho(z, w), \Delta_R, \neg \theta_2^{\text{IH}} \end{aligned}$$

With this in hand, we set

$$\begin{aligned} \theta &:= \exists z \in c. \theta_1^{\text{IH}} \wedge \theta_2^{\text{IH}} \\ \text{and} \quad E &:= \left\{ \{z \in c \mid \theta_2^{\text{IH}}\} \right\} \cup \bigcup \left\{ E_1^{\text{IH}} \cup E_2^{\text{IH}} \mid z \in c \right\} \end{aligned}$$

Note in particular that the free variables of E and θ are contained in C , since we bind z . The verification that E and θ suffice is routine; see the appendix.

Notice that the structure of our construction is: (1) by induction, we get NRC expressions E_i^{IH} satisfying the required property over nested relations; (2) using our witness proof, we get a formula θ satisfying some invariant over all models, and thus in particular on all nested relations; (3) we perform a construction combining θ and E_i^{IH} , and reason with the naïve semantics of NRC to argue for correctness. We use this reasoning template in the proof of our main theorem as well.

6 PROOF OF THE MAIN RESULT

We now turn to the proof of our main result:

THEOREM 2 (EFFECTIVE IMPLICIT TO EXPLICIT FOR NESTED DATA). *Given a witness for an implicit definition of o in terms of \vec{i} up to extensionality relative to Δ_0 $\varphi(\vec{i}, \vec{a}, o)$, one can compute NRC expression E such that for any \vec{i}, \vec{a} and o , if $\varphi(\vec{i}, \vec{a}, o)$ then $E(\vec{i}) = o$. Furthermore, if the witness is focused, this can be done in polynomial time.*

We have as input a proof of

$$\varphi(\vec{i}, \vec{a}, o) \wedge \varphi(\vec{i}, \vec{a}', o') \rightarrow o \equiv_T o'$$

and we want an NRC expression $E(\vec{i})$ such that

$$\varphi(\vec{i}, \vec{a}, o) \models_{\text{nested}} E(\vec{i}) \equiv_T o$$

This will be a consequence of the following theorem.

THEOREM 10. *Given Δ_0 $\varphi(\vec{i}, \vec{a}, o)$ and $\psi(\vec{i}, \vec{b}, o')$ together with a focused proof with conclusion*

$$\Theta(\vec{i}, \vec{a}, r); \varphi(\vec{i}, \vec{a}, r), \psi(\vec{i}, \vec{b}, o') \vdash \exists r' \in_p o'. r \equiv_T r'$$

we can compute in polynomial time an NRC expression $E(\vec{i})$ such that

$$\Theta(\vec{i}, \vec{a}, r); \varphi(\vec{i}, \vec{a}, r), \psi(\vec{i}, \vec{b}, o') \models_{\text{nested}} r \in E(\vec{i})$$

That is, we can find an NRC query that “collects answers”. Assuming Theorem 10, let’s prove the main result.

PROOF OF THEOREM 2. We assume o has a set type, deferring the simple product and Ur-element cases (the latter using GET) to the appendix. Fix an implicit definition of o up to extensionality relative to $\varphi(\vec{i}, \vec{a}, o)$ and a focused proof of

$$\varphi(\vec{i}, \vec{a}, o) \wedge \varphi(\vec{i}, \vec{a}', o') \vdash o \equiv_{\text{Set}(T)} o'$$

We can apply a simple variation of Lemma 6 for the “empty path” p to obtain a focused derivation of

$$r \in o; \varphi(\vec{i}, \vec{a}, o), \varphi(\vec{i}, \vec{a}', o') \vdash \exists r' \in o' r \equiv_T r' \quad (i)$$

Then applying Theorem 10 gives a NRC expression $E(\vec{i})$ such that

$$\varphi(\vec{i}, \vec{a}, o) \wedge r \in o \wedge \varphi(\vec{i}, \vec{a}', o') \vdash_{\text{nested}} r \in E(\vec{i})$$

Thus, the object determined by \vec{i} is always contained in $E(\vec{i})$. Coming back to (i), we can obtain a derivation of

$$r \in o; \varphi(\vec{i}, \vec{a}, o) \vdash \varphi(\vec{i}, \vec{a}', o') \rightarrow \exists r' \in o' r \equiv_T r'$$

and applying interpolation (Theorem 4) to that gives a Δ_0 formula $\kappa(\vec{i}, r)$ such that the following are valid

$$r \in o \wedge \varphi(\vec{i}, \vec{a}, o) \rightarrow \kappa(\vec{i}, r) \quad (ii)$$

$$\kappa(\vec{i}, r) \wedge \varphi(\vec{i}, \vec{a}', o') \rightarrow \exists r' \in o' r \equiv_T r' \quad (iii)$$

We claim that $E_\kappa(\vec{i}) = \{x \in E(\vec{i}) \mid \kappa(\vec{i}, x)\}$ is the desired NRC expression. To show this, assume $\varphi(\vec{i}, \vec{a}, o)$ holds. We know already that $o \subseteq E(\vec{i})$ and, by (ii), every $r \in o$ satisfies $\kappa(\vec{i}, r)$, so $o \subseteq E_\kappa(\vec{i})$. Conversely, if $x \in E_\kappa(\vec{i})$, we have $\kappa(\vec{i}, x)$, so by (iii), we have that $x \in o$, so $E_\kappa(\vec{i}) \subseteq o$. So $E_\kappa(\vec{i}) = o$, which concludes the proof. \square

We now turn to the proof of Theorem 10.

Proof of Theorem 10. We prove the theorem by induction over the type T . We only prove the inductive step for set types: the inductive case for products are straightforward.

For $T = \mathcal{U}$, the base case of the induction, it is clear that we can take for E an expression computing the set of all \mathcal{U} -elements in the transitive closure of \vec{i} . This can clearly be done in NRC.

So now, we assume $T = \text{Set}(T')$ and that Theorem 10 holds up to T' . We have a focused derivation of

$$\Theta; \varphi(\vec{i}, r), \psi(\vec{i}, o') \vdash \exists r' \in_p o' r \equiv_{\text{Set}(T')} r' \quad (iv)$$

omitting the additional variables for brevity.

From our input derivation, we can easily see that each element of r must be equivalent to some element below o' . This is reflected by Lemma 6, which allows us to efficiently compute a proof of

$$\Theta, z \in r; \varphi(\vec{i}, r), \psi(\vec{i}, o') \vdash \exists z' \in_{mp} o' z \equiv_{T'} z'$$

We can then apply the inductive hypothesis of our main theorem at sort T' , which is strictly smaller than $\text{Set}(T')$. This yields a NRC expression $E^{\text{IH}}(\vec{i})$ of type $\text{Set}(T')$ such that

$$\Theta, z \in r; \varphi(\vec{i}, r), \psi(\vec{i}, o') \vdash_{\text{nested}} z \in E^{\text{IH}}(\vec{i})$$

That is, our original hypotheses entail $r \subseteq E^{\text{IH}}(\vec{i})$.

Thus, we have used the inductive hypothesis to get a “superset expression”. But now we want an expression that has r as an element. We will do this by unioning a collection of definable subsets of $E^{\text{IH}}(\vec{i})$. To get these, we come back to our input derivation (iv). By Lemma 7, we can efficiently compute a derivation of

$$\Theta; \varphi(\vec{i}, r), \psi(\vec{i}, o') \vdash \exists r' \in_p o' \forall z \in a (z \in r \leftrightarrow z \in r')$$

where we take a to be a fresh variable of sort $\text{Set}(T')$. Now, applying our NRC Parameter Collection result (Theorem 8) we obtain a NRC expression $E^{\text{coll}}(\vec{i}, a)$ satisfying

$$\Theta; \varphi(\vec{i}, r), \psi(\vec{i}, o') \vdash_{\text{nested}} a \cap r \in E^{\text{coll}}(\vec{i}, a)$$

Now, recalling that we have $r \subseteq E^{\text{IH}}(\vec{i})$ and instantiating a to be $E^{\text{IH}}(\vec{i})$, we can conclude that

$$\Theta; \varphi(\vec{i}, r), \psi(\vec{i}, o') \vdash_{\text{nested}} r \in E^{\text{coll}}(\vec{i}, E^{\text{IH}}(\vec{i}))$$

Thus we can take $E^{\text{coll}}(\vec{i}, E^{\text{IH}}(\vec{i}))$ as an explicit definition. \square

7 DISCUSSION AND FUTURE WORK

Our effective nested Beth result implies that whenever a set of NRC views determines an NRC query, the query is rewritable over the views in NRC. Further, from a proof witnessing determinacy in our proof system, we can efficiently generate the rewriting. Our result applies to a setting where we have determinacy with respect to constraints and views, as in Example 1.1, or to general Δ_0 implicit definitions that may not stem from views.

In terms of impact on databases, a crucial limitation of our work is that we do not know how to find the proofs. In the case of relational data, we know of many “islands of decidability” where proofs of determinacy can be found effectively – e.g. for views and queries in guarded logics [1]. But it remains open to find similar decidability results for views/queries in fragments of NRC. The need to find proofs automatically is pressing since our system is so low-level that it is difficult to do proofs by hand. Indeed, a formal proof of implicit definability for Example 1.1, or even the simpler Example 4.1, would come to several pages.

The implicit-to-explicit methodology requires a proof of implicit definability, which implies implicit definability over all instances, not just finite ones. This requirement is necessary: one cannot hope to convert implicit definitions over finite instances to explicit NRC queries, even ineffectively. See the appendix for details.

Our key proof tool was the NRC Parameter Collection theorem, Theorem 8. There is an intuition behind this theorem that concerns a general setting, where we have a first-order theory Σ that factors into a conjunction of two formulas $\Sigma_L \wedge \Sigma_R$, and from this we have a notion of a “left formula” (with predicates from Σ_L), a “right formula” (predicates from Σ_R), and a “common formula” (all predicates occur in both Σ_L and Σ_R). Under the hypothesis that a left formula λ is definable from a right formula with parameters, we can conclude that the left formula must actually be definable from a common formula with parameters: see the appendix for a formal version and the corresponding proof.

Our work contributes to the broader topic of proof-theoretic vs model-theoretic techniques for interpolation and definability theorems. For Beth’s theorem, there are reasonably short model-theoretic [8, 24] and proof-theoretic arguments [10, 12]. In database terms, you can argue semantically that relational algebra is complete for rewritings of queries determined by views, and producing a rewriting from a proof of determinacy is not that difficult. But for a number of results on definability proved in the 60’s and 70’s [7, 14, 22, 25], there are short model-theoretic arguments, but no proof-theoretic ones. For our NRC analog of Beth’s theorem, the situation is more similar to the latter case: the model-theoretic proof of completeness [4] is relatively short and elementary, but generating explicit definitions from proofs is much more challenging. We hope that our results and tools represent a step towards providing effective versions, and towards understanding the relationship between model-theoretic and proof-theoretic arguments.

REFERENCES

- [1] Vince Bárány, Michael Benedikt, and Balder ten Cate. 2018. Some Model Theory of Guarded Negation. *J. Symb. Log.* 83, 4 (2018), 1307–1344.
- [2] Michael Benedikt, Pierre Bourhis, and Michael Vanden Boom. 2019. Definability and Interpolation within Decidable Fixpoint Logics. *Log. Methods Comput. Sci.* 15, 3 (2019), 29:1–29:53.
- [3] Michael Benedikt, Balder ten Cate, Julien Leblay, and Efthymia Tsamoura. 2016. *Generating Plans from Proofs: The Interpolation-Based Approach to Query Reformulation*. Morgan Claypool, San Rafael, CA.
- [4] Michael Benedikt and Pierre Pradic. 2021. Generating Collection Transformations from Proofs. In *POPL*.
- [5] Michael Benedikt, Balder ten Cate, and Michael Vanden Boom. 2016. Effective Interpolation and Preservation in Guarded Logics. *ACM TOCL* 17, 2 (2016), 8:1–8:46.
- [6] E. W. Beth. 1953. On Padoa's Method in the Theory of Definitions. *Indag. Mathematicae* 15 (1953), 330 – 339.
- [7] C. C. Chang. 1964. Some New Results in Definability. *Bull. of the AMS* 70, 6 (1964), 808 – 813.
- [8] C. C. Chang and H. Jerome Keisler. 1992. *Model Theory*. North-Holland.
- [9] William Craig. 1957. Linear Reasoning. A New Form of the Herbrand-Gentzen Theorem. *J. Symb. Log.* 22, 03 (1957), 250–268.
- [10] William Craig. 1957. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory. *J. Symb. Log.* 22, 3 (1957), 269–285.
- [11] Giovanna D'Agostino and Marco Hollenberg. 2000. Logical Questions Concerning The μ -Calculus: Interpolation, Lyndon and Los-Tarski. *J. Symb. Log.* 65, 1 (2000), 310–332.
- [12] Melvin Fitting. 1996. *First-order Logic and Automated Theorem Proving* (second ed.). Springer.
- [13] Enrico Franconi, Volha Kerhet, and Nhung Ngo. 2013. Exact Query Reformulation over Databases with First-order and Description Logics Ontologies. *J. Artif. Int. Res.* 48 (2013), 885–922.
- [14] Haim Gaifman. 1974. Operations on Relational Structures, Functors and Classes I. In *Proc. of the Tarski Symposium (Proc. of Symposia in Pure Mathematics)*, Vol. 25. 20–40.
- [15] Wilfrid Hodges. 1975. A Normal Form for Algebraic Constructions II. *Logique et Analyse* 18, 71/72 (1975), 429–487.
- [16] Wilfrid Hodges. 1993. *Model Theory*. Cambridge University Press.
- [17] Wilfrid Hodges, I.M. Hodkinson, and Dugald Macpherson. 1990. Omega-Categoricity, Relative Categoricity and Coordinatisation. *Annals of Pure and Applied Logic* 46, 2 (1990), 169 – 199.
- [18] Eva Hoogland, Maarten Marx, and Martin Otto. 1999. Beth Definability for the Guarded Fragment. In *LPAR*.
- [19] Guoxiang Huang. 1995. Constructing Craig Interpolation Formulas. In *Computing and Combinatorics*.
- [20] Christoph Koch. 2006. On the Complexity of Non-recursive XQuery and Functional Query Languages on Complex Values. *ACM TODS* 31, 4 (2006), 1215–1256.
- [21] Phokion G. Kolaitis. 1990. Implicit Definability on Finite Structures and Unambiguous Computations. In *LICS*.
- [22] David Kueker. 1971. Generalized Interpolation and Definability. *Annals of Mathematical Logic* 1, 4 (1971), 423–468.
- [23] E. G. K. Lopez-Escobar. 1965. An Interpolation Theorem for Denumerably Long Sentences. *Fundamenta Mathematica* 57 (1965), 253–272.
- [24] Roger C. Lyndon. 1959. An Interpolation Theorem in the Predicate Calculus. *Pacific J. Math.* 9 (1959), 129–142.
- [25] Michael Makkai. 1964. On a Generalization of a Theorem of E. W. Beth. *Acta Math. Ac. Sci. Hung.* 15 (1964), 227–235.
- [26] Sonia Marin, Dale Miller, Elaine Pimentel, and Marco Volpe. 2022. From Axioms to Synthetic Inference Rules via Focusing. *Annals of Pure and Applied Logic* 173, 5 (2022), 103091.
- [27] Alan Nash, Luc Segoufin, and Victor Vianu. 2010. Views and Queries: Determinacy and Rewriting. *ACM TODS* 35, 3 (2010), 1–41.
- [28] Sara Negri and Jan von Plato. 1998. Cut Elimination in the Presence of Axioms. *Bull. Symb. Log.* 4, 4 (1998), 418–435.
- [29] Sara Negri and Jan von Plato. 2001. *Structural Proof Theory*. Cambridge University Press.
- [30] Martin Otto. 2000. An Interpolation Theorem. *Bull. Symb. Log.* 6, 4 (2000), 447–462.
- [31] Jan Paredaens and Dirk Van Gucht. 1992. Converting Nested Algebra Expressions into Flat Algebra Expressions. *ACM TODS* 17, 1 (1992), 65–93.
- [32] Luc Segoufin and Victor Vianu. 2005. Views and Queries: Determinacy and Rewriting. In *PODS*.
- [33] Raymond M. Smullyan. 1968. *Craig's Interpolation Lemma and Beth's Definability Theorem*. In: *First-Order Logic*. Springer, 127–133.
- [34] Dan Suciu. 1995. *Parallel Programming Languages for Collections*. Ph.D. Dissertation. Univ. Pennsylvania.
- [35] Gaisi Takeuti. 1987. *Proof Theory* (second ed.). North-Holland.
- [36] Balder ten Cate, Enrico Franconi, and Inanç Seylan. 2013. Beth Definability in Expressive Description Logics. *J. Artif. Int. Res.* 48, 1 (2013), 347–414.
- [37] David Toman and Grant Weddell. 2011. *Fundamentals of Physical Design and Query Compilation*. Morgan Claypool.
- [38] Arne S. Troelstra and Helmut Schwichtenberg. 2000. *Basic Proof Theory*. Cambridge University Press.
- [39] Jan Van den Bussche. 2001. Simulation of the Nested Relational Algebra by the Flat Relational Algebra, with an Application to the Complexity of Evaluating Powerset Algebra Expressions. *Theor. Comput. Sci.* 254, 1–2 (2001), 363–377.
- [40] Limsoon Wong. 1994. *Querying Nested Collections*. Ph.D. Dissertation. Univ. Pennsylvania.

A COMPARISON TO THE SITUATION WITH FINITE INSTANCES

Our result concerns a specification $\Sigma(\vec{I}, O \dots)$ such that \vec{I} implicitly defines O . This can be defined “syntactically” – via the existence of a proof (e.g. in our own proof system). Thus, the class of queries that we deal with could be called the “provably implicitly definable queries”. The same class of queries can also be defined semantically, and this is how implicitly defined queries are often presented. But in order to be equivalent to the proof-theoretic version, we need the implicit definability of the object O over \vec{I} to hold considering all nested relations $\vec{I}, O \dots$, not just finite ones. Of course, the fact that when you phrase the property semantically requires referencing unrestricted instance does not mean that our results depend on the existence of infinite nested relations.

Discussion of finite vs. unrestricted instances appears in many other papers (e.g. [4]). And the results in this submission do not raise any new issues with regard to the topic. But we discuss what happens if we take the obvious analog of the semantic definition, but using only finite instances. Let us say that a Δ_0 specification $\Sigma(\vec{I}, O \vec{A})$ is *implicitly defines O in terms of \vec{I} over finite instances* if for any *finite* nested relations $\vec{I}, O, \vec{A}, O', \vec{A}'$, if $\Sigma(\vec{I}, O, \vec{A}) \wedge \Sigma(\vec{I}, O', \vec{A}')$ holds, then $O = O'$. If this holds, then Σ defines a query, and we call such a query *finitely implicitly definable*.

This class of queries is reasonably well understood, and we summarize what is known about it:

- *Can finitely implicitly definable queries always be defined in NRC?* The answer is a resounding “no”: one can implicitly define the powerset query over finite nested relations. Bootstrapping this, one can define iterated powersets, and show that the expressiveness of implicit definitions is the same as queries in NRC enhanced with powerset – a query language with non-elementary complexity. Even in the setting of relational queries, considering only finite instances leads to a query class that is not known to be in PTIME [21].
- *Can we generate explicit definitions from specifications Σ , given a proof that Σ implicitly defines O in terms of \vec{I} over finite instances?* It depends on what you mean by “a proof”, but in some sense there is no way to make sense of the question: there is no complete proof system for such definitions. This follows from the fact that the set of finitely implicitly definable queries is not computably enumerable.
- *Is sticking to specifications Σ that are implicit definitions over all inputs – as we do in this work – too strong?* Here the answer can not be definitive. But we know of no evidence that this is too restrictive in practice. Implicit specifications suffice to specify any NRC query. And the answer to the first question above says that if we modified the definition in the obvious way to get a larger class, we would allow specification of queries that do not admit efficient evaluation. The answer to the second question above says that we do not have a witness to membership in this larger class.

B CAPTURING NRC EXPRESSIONS WITH Δ_0 FORMULAS

In the body of the paper we mentioned that for every NRC expression $E(\vec{i})$, we can create a Δ_0 expression $\varphi_E(\vec{i}, o)$ such that $E(\vec{i}) = o$ exactly when $\varphi_E(\vec{i}, o)$ holds. These were called “input-output specifications”. This conversion is needed to reason about determinacy of queries by views in our formalism. If we start with the views and queries in NRC, we can use this transformation to get a corresponding Δ_0 specification.

Note that $o = E$ is itself an NRC expression of Boolean type. The result then follows from the fact that every NRC expression $E(\vec{w})$ of Boolean type can be converted to a Δ_0 formula $\varphi(\vec{w})$. This conversion can be done in polynomial time for the “composition-free” syntax for NRC [20] mentioned briefly in the body: in composition-free NRC, we restrict $\bigcup \{E|x \in E'\}$ so that E' must be a variable. One can normalize every expression to be of this form. The normalization is exponential, and under complexity-theoretic hypotheses one cannot do better [20]. The conversion from NRC Boolean expressions to Δ_0 is given in full detail in [4], although it is very similar to results on simulating NRC with flat relations given in prior work (e.g. [39]).

C COMPLETENESS OF PROOF SYSTEMS

In the body of the paper we mentioned that the completeness of the proof systems is argued using a standard method. We outline this for the higher-level system in Figure 2.

One has a sequent $\Theta; \Gamma \vdash \Delta$ that is not provable. We want to construct a countermodel: one that satisfying all the formulas in Θ and Γ but none of the formulas in Δ . We construct a tree with $\Theta; \Gamma \vdash \Delta$ at the root by iteratively applying applicable inference rules in reverse: in “proof search mode”, generating subgoals from goals. We apply the rules whenever possible in a given order, enforcing some fairness constraints: a rule that is active must be eventually applied along an infinite search, and if a choice of terms must be made (as with the \exists -R rule), all possible choices of terms are eventually made in an application of the rule. For example, if we have a disjunction $\rho_1 \vee \rho_2$ on the right, we may immediately “apply \vee -R”: we generate a subgoal where on the right hand side we add ρ_1, ρ_2 . Finite branches leading to a sequent that does not match the conclusion of any rule or axiom are artificially extended to infinite branches by repeating the topmost sequent.

By assumption, this process does not produce a proof, and thus we have an infinite branch b of the tree. We create a model M_b whose elements are the variables that appear on the branch, where an element inherits the type of its variable. The memberships correspond to the membership atoms that appear on the left of any sequent in b , and also the atoms that appear negated on the right hand side of any sequent.

We claim that M_b is the desired countermodel. It suffices to show that for every sequent $\Theta; \Gamma \vdash \Delta$ in b , M_b is a counterexample to the sequent: it satisfies the conjunction of formulas on the left and none of the formulas on the right. We prove this by induction on the logical complexity of the formula. For atoms it is immediate by construction. Each inductive step will involve the assumptions about inference rules not terminating proof search. For example, suppose for some sequent b_i in b of the above form, Δ contains $\rho_1 \vee \rho_2$, we want to show that M_b

satisfies $\neg(\rho_1 \vee \rho_2)$. But we know that in some successor of b , we would have applied \vee -R, and thus have a descendant with ρ_1, ρ_2 within the left. By induction M_b satisfies $\neg\rho_1$ and $\neg\rho_2$. Thus M_b satisfies $\neg(\rho_1 \vee \rho_2)$ as desired. The other connectives and quantifiers are handled similarly.

D Δ_0 INTERPOLATION: PROOF SKETCH OF THEOREM 4

We recall the statement:

Let Θ be an \in -context and Γ, Δ finite sets of Δ_0 formulas. Then from any proof of $\Theta; \Gamma \vdash \Delta$ we can compute in linear time a Δ_0 formula θ with $FV(\theta) \subseteq FV(\Theta, \Gamma) \cap FV(\Delta)$, such that $\Theta; \Gamma \vdash \theta$ and $\Theta; \theta \vdash \Delta$

Recall also that we claim this for both the higher-level 2-sided system and the 1-sided system, where the 2-sided syntax is a “macro”: $\Theta; \Gamma \vdash \Delta$ is a shorthand for $\Theta \vdash \neg\Gamma, \Delta$, where $\neg\Gamma$ is itself a macro for dualizing connectives. Thus in the 1-sided version, we are arbitrarily classifying some of the Δ_0 formulas as Left and the others as Right, and our interpolant must be common according to that partition.

We stress that there are no new ideas needed in proving Theorem 4 — unlike for our main tool, the Parameter Collection Theorem, or our final result. The construction for Theorem 4 proceeds exactly as in prior interpolation theorems for similar calculi [33, 35, 38]. Similar constructions are utilized in works for query reformulation in databases, so for a presentation geared towards a database audience one can check [37] or the later [3].

We explain the argument for the higher-level 2-sided system. We prove a more general statement, where we partition the context and the formulas on both sides of \vdash into Left and Right. So we have

$$\Theta_L \Theta_R; \Gamma_L, \Gamma_R \vdash \Delta_L, \Delta_R$$

And our inductive invariant is that we will compute in linear time a θ such that:

$$\begin{aligned} \Theta_L; \Gamma_L \vdash \theta, \Delta_L \\ \Theta_R; \Gamma_R, \theta \vdash \Delta_R \end{aligned}$$

And we require that $FV(\theta) \subseteq FV(\Theta_L, \Gamma_L, \Delta_L) \cap FV(\Theta_R, \Gamma_R, \Delta_R)$. This generalization is used to handle the negation rules, as we explain below.

We proceed by induction on the depth of the proof tree.

One of the base cases is where we have a trivial proof tree, which uses rule (Ax) to derive:

$$\Theta; \Gamma, \varphi \vdash \varphi, \Delta$$

We do a case distinction on where the occurrences of φ sit in our partition. Assume the occurrence on the left is in Γ_L and the occurrence on the right is in Δ_R . Then we can take our interpolant θ to be φ . Suppose the occurrence on the left is Γ_L and the occurrence on the right is in Δ_L . Then we can take θ to be \perp . The other base cases are similar.

The inductive cases for forming the interpolant will work “in reverse” for each proof rule. That is, if we used an inference rule to derive sequent S from sequents S_1 and S_2 , we will partition the sequents S_1 and S_2 based on the partition of S . We will then apply induction to our partitioned sequent for S_1 to get an interpolant θ_1 , and also apply induction to our partitioned version of S_2 to get an interpolant θ_2 . We then put them together to get the interpolant for the partitioned sequent S . This “putting together” will usually reflect the semantics of the connective mentioned in the proof rule.

Consider the case where the last rule applied is the \neg -L rule: this is the case that motivates the more general invariant involving partitions. We have a partition of the final sequent $\Theta; \Gamma, \varphi \vdash \Delta$. We form a partition of the sequent $\Theta; \Gamma \vdash \neg\varphi, \Delta$ by placing $\neg\varphi$ on the same side (Left, Right) as φ was in the original partition. We then get an interpolant θ by induction. We just use θ for the final interpolant.

We consider the inductive case for \wedge -R. We have two top sequents, one for each conjunct. We partition them in the obvious way: each φ_i in the top is in the same partition that $\varphi_1 \wedge \varphi_2$ was in the bottom. Inductively we take the interpolants θ_1 and θ_2 for each sequent. We again do a case analysis based on whether $\varphi_1 \wedge \varphi_2$ was in Δ_L or in Δ_R .

Suppose $\varphi_1 \wedge \varphi_2$ was in Δ_R , so $\Delta_R = \varphi_1 \wedge \varphi_2, \Delta'_R$. Then we arranged that each φ_i was in Δ_R in the corresponding top sequent. So we know that $\Theta_L; \Gamma_L \vdash \theta_i, \Delta_L$ and $\Theta_R; \Gamma_R, \theta_i \vdash \varphi_i, \Delta'_R$ for $i = 1, 2$. Now can set the interpolant θ to be $\theta_1 \wedge \theta_2$.

In the other case, $\varphi_1 \wedge \varphi_2$ was in Δ_L , say $\Delta_L = \varphi_1 \wedge \varphi_2, \Delta'_L$. Then we would arrange each φ_i to be “Left” in the corresponding top sequent, so we know that $\Theta_L; \Gamma_L \vdash \theta_i, \varphi_i, \Delta'_L$ and $\Theta_R; \Gamma_R, \theta_i \vdash \Delta_R$ for $i = 1, 2$. We set $\theta = \theta_1 \vee \theta_2$ in this case.

With the \exists rule, a term in the inductively-assumed θ' for the top sequent may become illegal for the θ for the bottom sequent, since it has a free variable that is not common. In this case, the term in θ is replaced by a quantified variable, where the quantifier is existential or universal, depending on the partitioning, and bounded according to the requirements for deltazero formulas.

E DETAILS FOR THE PROOF OF THEOREM 8, NESTED PARAMETER COLLECTION

Recall the major result on collecting parameters stated in the body of the paper:

THEOREM 8 (NRC PARAMETER COLLECTION). *Let L, R be sets of variables with $C = L \cap R$ and*

- φ_L and $\lambda(z)$ Δ_0 formulas over L

- φ_R and $\rho(z, y)$ Δ_0 formulas over R
- r a variable of R and c a variable of C .

Suppose that we have a proof of

$$\varphi_L \wedge \varphi_R \rightarrow \exists y \in_p r \forall z \in c (\lambda(z) \leftrightarrow \rho(z, y))$$

Then one may compute in polynomial time an NRC expression E with free variables in C such that

$$\varphi_L \wedge \varphi_R \rightarrow \{z \in c \mid \lambda(z)\} \in E$$

As we mentioned in the body of the paper, this is a corollary of the following lemma.

LEMMA 9. Let L, R be sets of variables with $C = L \cap R$ and

- $\Delta_L, \lambda(z)$ a set of Δ_0 formulas over L
- $\Delta_R, \rho(z, y)$ a set of Δ_0 formulas over R
- Θ_L (respectively Θ_R) a \in -context over L (respectively over R)
- r a variable of R and c a variable of C .

Suppose that we have a proof tree with conclusion

$$\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \exists y \in_p r \forall z \in c (\lambda(z) \leftrightarrow \rho(z, y))$$

Then one may compute in polynomial time an NRC expression E and a Δ_0 formula θ using only variables from C such that

$$\Theta_L \models_{\text{nested}} \Delta_L, \theta \vee \{z \in c \mid \lambda(z)\} \in E \quad \text{and} \quad \Theta_R \models_{\text{nested}} \Delta_R, \neg\theta$$

We now give the full details of the proof of the lemma.

We prove this by induction over the size of the proof of $\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \exists y \in_p r \forall z \in c (\lambda(z) \leftrightarrow \rho(z, y))$, making a case distinction according to which rule is applied last. The way θ will be built will, perhaps unsurprisingly, be very reminiscent of the way interpolations are normally constructed in standard proof systems [12, 33].

For readability, we adopt the following conventions:

- We write \mathcal{G} for the formula $\exists y \in_p r \forall z \in c (\lambda(z) \leftrightarrow \rho(z, y))$ and Λ for the expression $\{z \in c \mid \lambda(z)\}$.
- For formulas and NRC expressions obtained by applying the induction hypothesis, we use the names θ^{IH} and E^{IH} (or $\theta_1^{\text{IH}}, \theta_2^{\text{IH}}$ and $E_1^{\text{IH}}, E_2^{\text{IH}}$) when the induction hypothesis is applied several times). In each subcase, our goal will be to build suitable θ and E .
- We will color pairs of terms, formulas and sets of formulas according to whether they are part of either $\Theta_L; \Delta_L$ or $\Theta_R; \Delta_R$ either at the start of the case analysis or when we want to apply the induction hypothesis. In particular, the last sequent of the proof under consideration will be depicted as

$$\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \mathcal{G}$$

- Unless it is non-trivial, we leave checking that the free variables in our proposed definition for E and θ are taken among variables of C to the reader.

With these convention in mind, let us proceed.

- If the last rule applied is the \top rule, in both cases we are going to take $E := \emptyset$, but pick θ to be \perp or \top according to whether \top occurs in Δ_L or Δ_R ; we leave checking the details to the reader.
- If the last rule applied is the \wedge rule, we have two cases according to the position of the principal formula $\varphi_1 \wedge \varphi_2$. In both cases, E will be obtained by unioning NRC expressions obtained from the induction hypothesis, and θ will be either a disjunction or a conjunction.
 - If we have $\Delta_L = \varphi_1 \wedge \varphi_2, \Delta'_L$, so that the proof has shape

$$\frac{\Theta_L, \Theta_R \vdash \varphi_1, \Delta'_L, \Delta_R, \mathcal{G} \quad \Theta_L, \Theta_R \vdash \varphi_2, \Delta'_L, \Delta_R, \mathcal{G}}{\Theta_L, \Theta_R \vdash \varphi_1, \Delta'_L, \Delta_R, \mathcal{G}}$$

by the induction hypothesis, we have NRC expressions $E_1^{\text{IH}}, E_2^{\text{IH}}$ and formulas $\theta_1^{\text{IH}}, \theta_2^{\text{IH}}$ such that

$$\begin{array}{ll} \Theta_L \models_{\text{nested}} \varphi_1, \Delta'_L, \theta_1^{\text{IH}} \vee \Lambda \in E_1^{\text{IH}} & \text{and} \quad \Theta_R \models_{\text{nested}} \Delta_R, \neg\theta_1^{\text{IH}} \\ \Theta_L \models_{\text{nested}} \varphi_2, \Delta'_L, \theta_2^{\text{IH}} \vee \Lambda \in E_2^{\text{IH}} & \text{and} \quad \Theta_R \models_{\text{nested}} \Delta_R, \neg\theta_2^{\text{IH}} \end{array}$$

In that case, we take $E = E_1^{\text{IH}} \cup E_2^{\text{IH}}$ and $\theta := \theta_1^{\text{IH}} \vee \theta_2^{\text{IH}}$. Weakening the properties on the left column, we have

$$\Theta_L \models_{\text{nested}} \varphi_i, \Delta'_L, \theta \vee \Lambda \in E$$

for both $i \in \{1, 2\}$, so we have

$$\Theta_L \models_{\text{nested}} \varphi_1 \wedge \varphi_2, \Delta'_L, \theta \vee \Lambda \in E$$

as desired. Since $\neg\theta = \neg\theta_1^{\text{IH}} \wedge \neg\theta_2^{\text{IH}}$, we get

$$\Theta_R \models_{\text{nested}} \Delta_R, \neg\theta$$

by combining both properties from the right column.

- Suppose the last rule applied is \vee with principal formula $\varphi_1 \vee \varphi_2$. Depending on whether $\Delta_L = \varphi_1 \vee \varphi_2, \Delta'_L$ or $\Delta_R = \varphi_1 \vee \varphi_2, \Delta'_R$, the proof will end with one of the following steps

$$\frac{\Theta_L, \Theta_R \vdash \varphi_1, \varphi_2, \Delta'_L, \Delta_R, \mathcal{G}}{\Theta_L, \Theta_R \vdash \varphi_1 \vee \varphi_2, \Delta'_L, \Delta_R, \mathcal{G}} \quad \text{or} \quad \frac{\Theta_L, \Theta_R \vdash \Delta'_L, \varphi_1, \varphi_2, \Delta_R, \mathcal{G}}{\Theta_L, \Theta_R \vdash \Delta'_L, \varphi_1 \vee \varphi_2, \Delta_R, \mathcal{G}}$$

In both cases, we apply the inductive hypothesis according to the obvious splitting of contexts and sets of formulas, to get a NRC definition E^{IH} along with a formula θ^{IH} that satisfy the desired semantic property. We set $E := E^{\text{IH}}$ and $\theta := \theta^{\text{IH}}$.

- Suppose the last rule applied is \forall with principal formula $\forall x \in b.\varphi$. As in the previous case, depending on whether $\Delta_L = \forall x \in b.\varphi, \Delta'_L$ or $\Delta_R = \forall x \in b.\varphi, \Delta'_R$, the proof will end with one of the following steps (assuming y is fresh below)

$$\frac{\Theta_L, y \in b, \Theta_R \vdash \varphi[y/x], \Delta'_L, \Delta_R, \mathcal{G}}{\Theta_L, \Theta_R \vdash \forall x \in b.\varphi, \Delta'_L, \Delta_R, \mathcal{G}} \quad \text{or} \quad \frac{\Theta_L, \Theta_R, y \in b \vdash \Delta'_L, \varphi[y/x], \Delta_R, \mathcal{G}}{\Theta_L, \Theta_R \vdash \Delta'_L, \forall x \in b.\varphi, \Delta_R, \mathcal{G}}$$

In both cases, we again apply the inductive hypothesis according to the obvious splitting of contexts and sets of formulas to get a NRC definition E^{IH} and a formula θ^{IH} that satisfy the desired semantic property. We set $E := E^{\text{IH}}$ and $\theta := \theta^{\text{IH}}$.

- Now we consider the case where the last rule applied is \exists . Here we have two main subcases, according to whether \mathcal{G} is the main formula or not. The case where the main formula of \mathcal{G} was sketched in the body of the paper, but we repeat it in more detail here.
 - If $\mathcal{G} = \exists y \in_p r \forall z \in c. (\lambda(z) \leftrightarrow \rho(z, y))$ is the main formula, Δ_L, Δ_R is necessarily existential-leading and the proof necessarily has shape

$$\begin{array}{c} \vee \\ \wedge \end{array} \frac{\frac{\Theta_L, \Theta_R, x \in c \vdash \Delta_L, \Delta_R, \neg\rho(x, w), \lambda(x), \mathcal{G}}{\Theta_L, \Theta_R, x \in c \vdash \Delta_L, \Delta_R, \rho(x, w) \rightarrow \lambda(x), \mathcal{G}} \quad \vee \frac{\Theta_L, \Theta_R, x \in c \vdash \Delta_L, \Delta_R, \neg\lambda(x), \rho(x, w), \mathcal{G}}{\Theta_L, \Theta_R, x \in c \vdash \Delta_L, \Delta_R, \lambda(x) \rightarrow \rho(x, w), \mathcal{G}}}{\frac{\Theta_L, \Theta_R, x \in c \vdash \Delta_L, \Delta_R, \lambda(x) \leftrightarrow \rho(x, w), \mathcal{G}}{\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \forall z \in c. (\lambda(z) \leftrightarrow \rho(z, w)), \mathcal{G}}} \quad \exists \quad \frac{}{\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \mathcal{G}}$$

where x is a fresh variable. So in particular, we have two strict subproofs with respective conclusions

$$\Theta_L, x \in c, \Theta_R, x \in c \vdash \lambda(x), \Delta_L, \neg\rho(x, w), \Delta_R, \mathcal{G} \quad \text{and} \quad \Theta_L, x \in c, \Theta_R, x \in c \vdash \neg\lambda(x), \Delta_L, \rho(x, w), \Delta_R, \mathcal{G}$$

Applying the inductive hypothesis, we obtain NRC expressions $E_1^{\text{IH}}, E_2^{\text{IH}}$ and formulas $\theta_1^{\text{IH}}, \theta_2^{\text{IH}}$ which contain free variables in $C \cup \{x\}$ such that all of the following hold

$$\Theta_L, x \in c \models_{\text{nested}} \lambda(x), \Delta_L, \theta_1^{\text{IH}} \vee \Lambda \in E_1^{\text{IH}} \quad (\text{v})$$

$$\text{and} \quad \Theta_L, x \in c \models_{\text{nested}} \neg\lambda(x), \Delta_L, \theta_2^{\text{IH}} \vee \Lambda \in E_2^{\text{IH}} \quad (\text{vi})$$

$$\text{and} \quad \Theta_R \models_{\text{nested}} \neg\rho(x, w), \Delta_R, \neg\theta_1^{\text{IH}} \quad (\text{vii})$$

$$\text{and} \quad \Theta_R \models_{\text{nested}} \rho(x, w), \Delta_R, \neg\theta_2^{\text{IH}} \quad (\text{viii})$$

With this in hand, we set

$$\theta := \exists x \in c. \theta_1^{\text{IH}} \wedge \theta_2^{\text{IH}} \quad \text{and} \quad E := \left\{ \{x \in c \mid \theta_2^{\text{IH}}\} \right\} \cup \bigcup \left\{ E_1^{\text{IH}} \cup E_2^{\text{IH}} \mid x \in c \right\}$$

Note in particular that the free variables of E and θ are contained in C , since we bind x . The bindings of x have radically different meaning across the two main components $E_1 := \{ \{x \in c \mid \theta_2^{\text{IH}}\} \}$ and $E_2 := \{ E_1^{\text{IH}} \cup E_2^{\text{IH}} \mid x \in c \}$ of $E = E_1 \cup E_2$. E_1 consists of a single definition corresponding to the restriction of c to θ_2^{IH} , and there x plays the role of an element being defined. On the other hand, E_2 corresponds to the joining of all the definitions obtained inductively, which may contain an $x \in c$ as a parameter. So we have two families of potential definitions for Λ indexed by $x \in c$ that we join together. Now let us show that we have the desired semantic properties. First we need to show that E contains a definition for Λ under the right hypotheses, i.e.,

$$\Theta_L \models_{\text{nested}} \Delta_L, \exists x \in c. \theta_1^{\text{IH}} \wedge \theta_2^{\text{IH}}, \Lambda \in \left(\left\{ \{x \in c \mid \theta_2^{\text{IH}}\} \right\} \cup \bigcup \left\{ E_1^{\text{IH}} \cup E_2^{\text{IH}} \mid x \in c \right\} \right) \quad (\text{ix})$$

which can be rephrased as

$$\Theta_L \models_{\text{nested}} \Delta_L, \exists x \in c. \theta_1^{\text{IH}} \wedge \theta_2^{\text{IH}}, \Lambda = \{x \in c \mid \theta_2^{\text{IH}}\}, \exists x \in c. \Lambda \in E_1^{\text{IH}} \cup E_2^{\text{IH}}$$

Now concentrate on the statement $\Lambda = \{x \in c \mid \theta_2^{\text{IH}}\}$. It would follow from the two inclusions $\Lambda \subseteq \{x \in c \mid \theta_2^{\text{IH}}\}$ and $\{x \in c \mid \theta_2^{\text{IH}}\} \subseteq \Lambda$, so, recalling that $\Lambda = \{x \in c \mid \lambda(x)\}$, the overall conclusion would follow from having

$$\Theta_L, x \in c \models_{\text{nested}} \Delta_L, \exists x \in c. \theta_1^{\text{IH}} \wedge \theta_2^{\text{IH}}, \lambda(x) \rightarrow \theta_2^{\text{IH}}, \exists x \in c. \Lambda \in E_1^{\text{IH}} \cup E_2^{\text{IH}}$$

$$\text{and} \quad \Theta_L, x \in c \models_{\text{nested}} \Delta_L, \exists x \in c. \theta_1^{\text{IH}} \wedge \theta_2^{\text{IH}}, \theta_2^{\text{IH}} \rightarrow \lambda(x), \exists x \in c. \Lambda \in E_1^{\text{IH}} \cup E_2^{\text{IH}}$$

Those in turn follow from the following two statements

$$\begin{aligned} & \Theta_L, x \in c \models_{\text{nested}} \Delta_L, \theta_1^{\text{IH}} \wedge \theta_2^{\text{IH}}, \neg \lambda(x), \theta_2^{\text{IH}}, \Lambda \in E_1^{\text{IH}} \cup E_2^{\text{IH}} \\ \text{and} \quad & \Theta_L, x \in c \models_{\text{nested}} \Delta_L, \theta_1^{\text{IH}} \wedge \theta_2^{\text{IH}}, \neg \theta_2^{\text{IH}}, \lambda(x), \Lambda \in E_1^{\text{IH}} \cup E_2^{\text{IH}} \end{aligned}$$

which are straightforward consequences of vi and v respectively. This concludes the proof of ix.

Now we only need to prove a final property, which is

$$\Theta_R \models_{\text{nested}} \Delta_R, \forall x \in c. \neg \theta_1^{\text{IH}} \vee \neg \theta_2^{\text{IH}}$$

which is equivalent to the validity of

$$\Theta_R, x \in c \models_{\text{nested}} \Delta_R, \neg \theta_1^{\text{IH}} \vee \neg \theta_2^{\text{IH}}$$

which can be obtained by combining vii and viii with excluded middle for $\rho(x, w)$.

- If \mathcal{G} is not the main formula, then we have two subcases corresponding to whether the main formula under consideration occurs in Δ_L or Δ_R . In both cases, the restriction on the \exists -rule imposed by the focused proof system is not particularly relevant and the block quantification hinders readability. Let us treat the equivalent of the case of a more general rule with a single quantifier in an auxiliary lemma.

LEMMA 11. Suppose that we have a formula θ^{IH} and a NRC expression E^{IH} such that

$$\Theta_L \models_{\text{nested}} \varphi[w/x], \exists x \in r. \varphi, \Delta_L, \theta^{\text{IH}} \vee \Lambda \in E^{\text{IH}} \quad \text{and} \quad \Theta_R \models_{\text{nested}} \Delta_R, \neg \theta^{\text{IH}}$$

with $FV(E^{\text{IH}}, \theta^{\text{IH}}) \subseteq FV(\Theta_L, \Delta_L, \varphi[w/x]) \cap FV(\Theta_R, \Delta_R)$. and that we additionally that $w \in t$ is part of Θ_L, Θ_R . Then we can define θ and E with $FV(E, \theta) \subseteq FV(\Theta_L, \Delta_L, \varphi) \cap FV(\Theta_R, \Delta_R)$ and

$$\Theta_L \models_{\text{nested}} \exists x \in t. \varphi, \Delta_L, \theta \vee \Lambda \in E \quad \text{and} \quad \Theta_R \models_{\text{nested}} \Delta_R, \neg \theta$$

PROOF. There are two subcases to consider:

- * If $w \in t$ is part of Θ_L , we can conclude immediately by setting $\theta := \theta^{\text{IH}}$ and $E := E^{\text{IH}}$.
- * Otherwise $w \in t$ is part of Θ_R , and it might be the case that w is a free variable which is not part of $\Theta_L; \Delta_L, \exists x \in t. \varphi$, so θ^{IH} and E^{IH} may feature w as a free variable. In that case, we know that the free variables of t are common and we can set $\theta := \forall x \in t. \theta[x/w]$ and $E := \bigcup \{E \mid w \in t\}$.

□

A dual lemma where the existential formula is located on the right side of the partition can be proven in a completely analogous way. We can use those lemmas to prove the result by induction on the size of the quantifier block for the focused \exists rule.

- The case of the $=$ rule can be handled exactly as the \top rule.
- For the \neq rule, we distinguish several subcases:
 - If we have $\Delta_L = y \neq_{\text{U}} z, \alpha[y/x], \Delta'_L$ or $\Delta_R = y \neq_{\text{U}} z, \alpha[y/x], \Delta'_R$, so that the last step has either shape

$$\frac{\Theta_L, \Theta_R \vdash y \neq_{\text{U}} z, \alpha[y/x], \alpha[z/x], \Delta'_L, \Delta'_R, \mathcal{G}}{\Theta_L, \Theta_R \vdash y \neq_{\text{U}} z, \alpha[y/x], \Delta'_L, \Delta'_R, \mathcal{G}} \quad \text{or} \quad \frac{\Theta_L, \Theta_R \vdash \Delta'_L, y \neq_{\text{U}} z, \alpha[y/x], \alpha[z/x], \Delta'_R, \mathcal{G}}{\Theta_L, \Theta_R \vdash \Delta'_L, y \neq_{\text{U}} z, \alpha[y/x], \Delta'_R, \mathcal{G}}$$

we can apply the induction hypothesis to obtain some θ^{IH} and E^{IH} such that setting $\theta := \theta^{\text{IH}}$ and $E := E^{\text{IH}}$ solves this subcase; we leave checking the additional properties to the reader.

- Otherwise, if we have $\Delta_L = y \neq_{\text{U}} z, \Delta'_L, \Delta_R = \alpha[y/x], \Delta'_R$ and a last step of shape

$$\frac{\Theta_L, \Theta_R \vdash y \neq_{\text{U}} z, \Delta'_L, \alpha[y/x], \alpha[z/x], \Delta'_R, \mathcal{G}}{\Theta_L, \Theta_R \vdash y \neq_{\text{U}} z, \Delta'_L, \alpha[y/x], \Delta'_R, \mathcal{G}}$$

In that case, the inductive hypothesis gives θ^{IH} and E^{IH} with free variables in $C \cup \{z\}$ such that

$$\Theta_L \models_{\text{nested}} y \neq_{\text{U}} z, \Delta'_L, \theta^{\text{IH}}, \Lambda \in E^{\text{IH}} \quad \text{and} \quad \Theta_R \models_{\text{nested}} \alpha[y/x], \alpha[z/x], \Delta'_R, \neg \theta^{\text{IH}}$$

We then have two subcases according to whether $z \in C$ or not

- * If $z \in C$, we can take $\theta := \theta^{\text{IH}} \wedge y =_{\text{U}} z$ and $E := E^{\text{IH}}$. Their free variables are in C and we only need to check

$$\Theta_L \models_{\text{nested}} y \neq_{\text{U}} z, \Delta'_L, \theta^{\text{IH}} \wedge y =_{\text{U}} z, \Lambda \in E^{\text{IH}} \quad \text{and} \quad \Theta_R \models_{\text{nested}} \alpha[y/x], \Delta'_R, \neg \theta^{\text{IH}}, y \neq_{\text{U}} z$$

which follow easily from the induction hypothesis.

- * Otherwise, we take $\theta := \theta^{\text{IH}}[y/z]$ and $E := E^{\text{IH}}[y/z]$. In that case, note that we have $\alpha[z/x][y/z] = \alpha[y/x]$ (which would not be necessarily the case if z belonged to C). This allows to conclude that we have

$$\Theta_L \models_{\text{nested}} y \neq_{\text{U}} z, \Delta'_L, \theta^{\text{IH}}[y/z], \Lambda \in E^{\text{IH}}[y/z] \quad \text{and} \quad \Theta_R \models_{\text{nested}} \alpha[y/x], \Delta'_R, \neg\theta^{\text{IH}}[y/z]$$

directly from the induction hypothesis.

- Otherwise, if we have $\Delta_L = \alpha[y/x], \Delta'_L, \Delta_R = y \neq_{\text{U}} z, \Delta'_R$ and a last step of shape

$$\frac{\Theta_L, \Theta_R \vdash \alpha[y/x], \alpha[z/x], \Delta'_L, y \neq_{\text{U}} z, \Delta'_R, \mathcal{G}}{\Theta_L, \Theta_R \vdash \alpha[y/x], \Delta'_L, y \neq_{\text{U}} z, \Delta'_R, \mathcal{G}}$$

In that case, the inductive hypothesis gives θ^{IH} and E^{IH} with free variables in $C \cup \{z\}$ such that

$$\Theta_L \models_{\text{nested}} \alpha[y/x], \alpha[z/x], \Delta'_L, \theta^{\text{IH}}, \Lambda \in E^{\text{IH}} \quad \text{and} \quad \Theta_R \models_{\text{nested}} y \neq_{\text{U}} z, \Delta'_R, \neg\theta^{\text{IH}}$$

We then have two subcases according to whether $z \in C$ or not

- * If $z \in C$, we can take $\theta := \theta^{\text{IH}} \vee y \neq_{\text{U}} z$ and $E := E^{\text{IH}}$. Their free variables are in C and we only need to check

$$\Theta_L \models_{\text{nested}} \alpha[y/x], \Delta'_L, \theta^{\text{IH}} \vee y \neq_{\text{U}} z, \Lambda \in E^{\text{IH}} \quad \text{and} \quad \Theta_R \models_{\text{nested}} y \neq_{\text{U}} z, \Delta'_R, \neg\theta^{\text{IH}} \wedge y =_{\text{U}} z$$

which follow easily from the induction hypothesis.

- * Otherwise, we take $\theta := \theta^{\text{IH}}[y/z]$ and $E := E^{\text{IH}}[y/z]$. In that case, note that we have $\alpha[z/x][y/z] = \alpha[y/x]$ (which would not be necessarily the case if z belonged to C). This allows to conclude that we have

$$\Theta_L \models_{\text{nested}} \alpha[y/x], \Delta'_L, \theta^{\text{IH}}[y/z], \Lambda \in E^{\text{IH}}[y/z] \quad \text{and} \quad \Theta_R \models_{\text{nested}} y \neq_{\text{U}} z, \Delta'_R, \neg\theta^{\text{IH}}[y/z]$$

directly from the induction hypothesis.

- If the last rule applied is \times_{η} , the proofs has shape

$$\frac{\Theta_L[\langle x_1, x_2 \rangle / x], \Theta_R[\langle x_1, x_2 \rangle / x] \vdash \Delta_L[\langle x_1, x_2 \rangle / x], \Delta_R[\langle x_1, x_2 \rangle / x], \mathcal{G}[\langle x_1, x_2 \rangle / x]}{\Theta_L, \Theta_R \vdash \Delta_L, \Delta_R, \mathcal{G}}$$

and one applies the inductive hypothesis as expected to get θ^{IH} and E^{IH} such that

$$\Theta_L[\langle x_1, x_2 \rangle / x] \models_{\text{nested}} \Delta_L[\langle x_1, x_2 \rangle / x], \theta^{\text{IH}}, \Lambda[\langle x_1, x_2 \rangle / x] \in E \quad \text{and} \quad \Theta_R[\langle x_1, x_2 \rangle / x] \models_{\text{nested}} \Delta_R[\langle x_1, x_2 \rangle / x], \neg\theta^{\text{IH}}$$

and with free variables included in C if $x \notin C$ or $(C \cup \{x_1, x_2\}) \setminus \{x\}$ otherwise. In both cases, it is straightforward to check that taking $\theta := \theta^{\text{IH}}[\pi_1(x)/x_1, \pi_2(x)/x_2]$ and $E := E^{\text{IH}}[\pi_1(x)/x_1, \pi_2(x)/x_2]$ will yield the desired result.

- Finally, if the last rule applied is the \times_{β} rule, it has shape

$$\frac{\Theta_L[x_i/x], \Theta_R[x_i/x] \vdash \Delta_L[x_i/x], \Delta_R[x_i/x], \mathcal{G}'[x_i/x]}{\Theta_L[\pi_i(\langle x_1, x_2 \rangle)/x], \Theta_R[\pi_i(\langle x_1, x_2 \rangle)/x] \vdash \Delta_L[\pi_i(\langle x_1, x_2 \rangle)/x], \Delta_R[\pi_i(\langle x_1, x_2 \rangle)/x], \mathcal{G}'[\pi_i(\langle x_1, x_2 \rangle)/x]}$$

and we can apply the induction hypothesis to get satisfactory θ^{IH} and E^{IH} (moving from \mathcal{G} to $\mathcal{G}'[x_i/x]$ is unproblematic, as we can assume the lemma works for \mathcal{G} with arbitrary subformulas λ and ρ); it is easy to see that we can set $\theta := \theta^{\text{IH}}$ and $E := E^{\text{IH}}$ and conclude.

This completes the proof of the lemma.

F PROOFS OF POLYTIME ADMISSIBILITY

The goal of this section is to prove most claims of polytime admissibility made in the body of the paper, crucially those of Section 6. Recall that a rule

$$\frac{\Theta \vdash \Delta}{\Theta' \vdash \Delta'}$$

is *polytime admissible* if we can compute in polynomial time a proof of the conclusion $\Theta' \vdash \Delta'$ from a proof of the antecedent $\Theta \vdash \Delta$, and *polytime derivable* if there is a polynomial-size poof tree with dangling leaves labelled by the antecedent.

Throughout this section we deal with the focused proof system of Figure 3.

F.1 Standard rules

Here we collect some useful standard sequent calculi rules, which are all polytime admissible in our system. The arguments for these rules are straightforward.

LEMMA 12. *The following weakening rule is polytime admissible:*

$$\frac{\Theta \vdash \Delta}{\Theta' \vdash \Delta, \Delta'}$$

LEMMA 13. *The following inference, witnessing the invertibility of the \wedge rule, is polytime admissible for both $i \in \{1, 2\}$:*

$$\frac{\Theta \vdash \varphi_1 \wedge \varphi_2, \Delta}{\Theta \vdash \varphi_i, \Delta}$$

LEMMA 14. *The following, witnessing the invertibility of the \forall rule, is polytime admissible:*

$$\frac{\Theta \vdash \forall x \in t. \varphi, \Delta}{\Theta, x \in t \vdash \varphi, \Delta}$$

LEMMA 15. *The following generalization of the \exists rule is polytime admissible:*

$$\frac{\Theta, \Theta_2 \vdash \varphi', \varphi, \Delta^{\text{EL}} \quad \varphi' \text{ is a spec of } \varphi \text{ wrt an ordering of } \Theta_2}{\Theta, \Theta_2 \vdash \varphi, \Delta^{\text{EL}}}$$

LEMMA 16. *The following substitution rule is polytime admissible:*

$$\frac{\Theta \vdash \Delta}{\Theta[t/x] \vdash \Delta[t/x]}$$

F.2 Admissibility of generalized congruence

Recall the admissibility claim concerning the rule related to congruence:

LEMMA 17. *The following generalized congruence rule is polytime admissible:*

$$\frac{\Theta[t/x, t/y] \vdash \Delta[t/x, t/y]}{\Theta[t/x, u/y] \vdash \neg(t \equiv u), \Delta[t/x, u/y]}$$

Recall that in a two-side reading of this, the hypothesis is $t \equiv u; \Theta[t/x, u/y] \vdash \Delta[t/x, u/y]$. So the rule says that if we Θ entails Δ where both contain t , then if we assume $t \equiv u$ and substitute some occurrences of t with u in Θ , we can conclude the corresponding substitution of Δ .

To prove Lemma 17 in the case where the terms t and u are of type $\text{Set}(T)$, we will need a more general statement. We are going to generalize the statement to treat tuples of terms and use EL formulas instead of $\neg(t \equiv u)$ to simplify the inductive invariant.

Given two terms t and u of type T , define by induction the set of formulas $\mathcal{E}_{t,u}$:

- If $T = \mathbf{U}$, then $\mathcal{E}_{t,u}$ is $t \neq_{\mathbf{U}} u$
- If $T = T_1 \times T_2$, then $\mathcal{E}_{t,u}$ is $\mathcal{E}_{\pi_1(t), \pi_1(u)}, \mathcal{E}_{\pi_2(t), \pi_2(u)}$
- If $T = \text{Set}(T')$, $\mathcal{E}_{t,u}$ is $\neg(t \subseteq_{T'} u), \neg(u \subseteq_{T'} t)$

The reader can check that $\mathcal{E}_{t,u}$ is essentially $\neg(t \equiv u)$.

LEMMA 18. *The following rule is polytime admissible:*

$$\frac{\Theta \vdash \Delta, \mathcal{E}_{t,u}}{\Theta \vdash \Delta, \neg(t \equiv u)}$$

PROOF. Straightforward induction over T . □

Since we will deal with multiple equivalences, we will adopt vector notation $\vec{t} = t_1, \dots, t_n$ and $\vec{x} = x_1, \dots, x_n$ for lists of terms and variables. Call $\mathcal{E}_{\vec{t}, \vec{u}}$ the union of the \mathcal{E}_{t_i, u_i} . We can now state our more general lemma:

LEMMA 19. *The following generalized n -ary congruence rule for set variables is polytime admissible:*

$$\frac{\Theta[\vec{t}/\vec{x}, \vec{t}/\vec{y}] \vdash \Delta[\vec{t}/\vec{x}, \vec{t}/\vec{y}]}{\Theta[\vec{t}/\vec{x}, \vec{u}/\vec{y}] \vdash \Delta[\vec{t}/\vec{x}, \vec{u}/\vec{y}], \mathcal{E}_{\vec{t}, \vec{u}}}$$

PROOF OF LEMMA 19. We proceed by induction over the proof of $\Theta[\vec{t}/\vec{x}, \vec{t}/\vec{y}] \vdash \Delta[\vec{t}/\vec{x}, \vec{u}/\vec{y}]$,

- If the last rule applied is the $=$ rule, i.e. we have

$$\overline{\Theta[\vec{t}/\vec{x}, \vec{t}/\vec{y}] \vdash a[\vec{t}/\vec{x}, \vec{t}/\vec{y}] =_{\mathbf{U}} b[\vec{t}/\vec{x}, \vec{t}/\vec{y}], \Delta[\vec{t}/\vec{x}, \vec{t}/\vec{y}]}$$

and $a[\vec{t}/\vec{x}, \vec{t}/\vec{y}] = b[\vec{t}/\vec{x}, \vec{t}/\vec{y}]$, with a, b variables. Now if a and b are equal, or if they belong both to either \vec{x} or \vec{y} , it is easy to derive the desired conclusion with a single application of the $=$ rule. Otherwise, assume $a = x_i$ and $b = y_j$ (the symmetric case is handled similarly). In such a case, we have that $\mathcal{E}_{\vec{t}, \vec{u}}$ contains $t_i \neq_{\mathbf{U}} t_j$. So the desired proof

$$\overline{\Theta[\vec{t}/\vec{x}, \vec{t}/\vec{y}] \vdash t_i =_{\mathbf{U}} u_j, \Delta[\vec{t}/\vec{x}, \vec{t}/\vec{y}], \mathcal{E}_{\vec{t}, \vec{u}}, t_i \neq_{\mathbf{U}} t_j}$$

follows from the polytime admissibility of the axiom rule.

- Suppose the last rule applied is the \top rule:

$$\frac{}{\Theta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] \vdash \top, \Delta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}]}$$

Then we do not need to apply the induction hypothesis. Instead we can immediately apply the \top rule to obtain

$$\frac{}{\Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}] \vdash \top, \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}}}$$

- If the last rule applied is the \wedge rule

$$\frac{\Theta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] \vdash \varphi_1[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] \quad \Theta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] \vdash \varphi_2[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}]}{\Theta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] \vdash (\varphi_1 \wedge \varphi_2)[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}]}$$

then the induction hypothesis gives us proofs of

$$\Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}] \vdash \varphi_i[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}}$$

for both $i \in \{1, 2\}$. So we can apply the \wedge rule to conclude that we have

$$\Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}] \vdash (\varphi_1 \wedge \varphi_2)[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}}$$

as desired.

- The cases of the rules \vee, \forall and \times_η are equally straightforward and left to the reader.
- Now, let us handle the case of the \exists rule. To simplify notation, we just treat the case where there is only one leading existential in the formula.

$$\frac{\Theta, t \in u \vdash \varphi[t/x], \Delta^{\text{EL}}}{\Theta, t \in u \vdash \exists x \in u. \varphi, \Delta^{\text{EL}}}$$

The generalization to multiple existentials can be obtained by translating the general \exists rule into a sequence of applications of the rule above and applying the same argument to each instance (note that this is possible because we do not require φ to be AL in the rule for a single existential).

So assume that z is fresh wrt $\tilde{x}, \tilde{y}, \tilde{t}, \tilde{u}, a, b$ and that the last step of the proof is

$$\frac{\Theta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}], a[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] \in b[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] \vdash \varphi[a/z][\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}], \exists z \in c[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}]. \varphi[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}]}{\Theta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}], a[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] \in b[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] \vdash \exists z \in c[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}]. \varphi[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}]}$$

with $b[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}] = c[\tilde{t}/\tilde{x}, \tilde{t}/\tilde{y}]$. Set $a' = a[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}]$, $b' = b[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}]$, $c' = c[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}]$. We have three subcases:

- If we have that $b' = c'$, using the induction hypothesis, we have a proof of

$$\Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], a' \in b' \vdash \varphi[a/z][\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \exists z \in c'. \varphi[\tilde{u}/\tilde{x}, \tilde{u}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}}$$

we can simply apply an \exists rule to that proof and we are done.

- Otherwise, if we have that $b' = t_i$ and $c' = u_j$ for some $i, j \leq n$. In that case, extending the tuples \tilde{t} and \tilde{u} with a' and a fresh variable z' (the substitutions under consideration would be, we can apply the induction hypothesis to obtain a proof of

$$\Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}, a'/z], a' \in t_i, z' \in u_j \vdash \varphi[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}, z'/z], \exists z \in u_j. \varphi[\tilde{u}/\tilde{x}, \tilde{u}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}}, \mathcal{E}_{a', z'}$$

Note that $\mathcal{E}_{\tilde{t}, \tilde{u}}$ contains an occurrence of $\neg(t_i \subseteq u_j)$, which expands to $\exists z \in t_i. \forall z' \in u_j. \neg(z \equiv z')$, so we can construct the partial derivation

$$\begin{array}{c} \frac{\Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}, a'/z], a' \in t_i, z' \in u_j \vdash \varphi[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}, z'/z], \exists z \in u_j. \varphi[\tilde{u}/\tilde{x}, \tilde{u}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}}, \mathcal{E}_{a', z'}}{\exists} \\ \text{Lemma 18} \frac{\Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], a' \in t_i, z' \in u_j \vdash \exists z \in u_j. \varphi[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}}, \mathcal{E}_{a', z'}}{\wedge} \\ \frac{\Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}, a'/z], a' \in t_i, z' \in u_j \vdash \exists z \in u_j. \varphi[\tilde{u}/\tilde{x}, \tilde{u}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}}, a' \equiv z'}{\vee} \\ \frac{\Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}, a'/z], a' \in t_i \vdash \exists z \in c'. \varphi[\tilde{u}/\tilde{x}, \tilde{u}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}}, \forall z' \in u_j. a' \equiv z'}{\exists} \\ \Theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}, a'/z], a' \in t_i \vdash \exists z \in c'. \varphi[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \Delta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}], \mathcal{E}_{\tilde{t}, \tilde{u}} \end{array}$$

whose conclusion matches what we want.

- Otherwise, we are in a similar case where $b' = u_j$ and $c' = t_i$. We proceed similarly, except that we use the formula $\neg(u_j \subseteq t_i)$ of \mathcal{E}_{t_i, u_j} instead of $\neg(t_i \subseteq u_j)$.
- For the rule \times_β , which has general shape

$$\frac{\Theta[z_i/z] \vdash \Delta[z_i/z]}{\Theta[\pi_i(\langle z_1, z_2 \rangle)/z] \vdash \Delta[\pi_i(\langle z_1, z_2 \rangle)/z]}$$

we can assume, without loss of generality, that z occurs only once in Θ, Δ . Let us only sketch the case where z occurs in a formula φ and the rule has shape

$$\frac{\Theta \vdash \varphi[z_i/z], \Delta}{\Theta \vdash \varphi[\pi_i(\langle z_1, z_2 \rangle)/z], \Delta}$$

We have that $\varphi[\pi_i(\langle z_1, z_2 \rangle)/z]$ is also of the shape $\psi[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}]$ in our situation. We can also assume without loss of generality that each variable in \tilde{x} and \tilde{y} occur each a single time in Θ, Δ . Now if we have a couple of situations:

- * If the occurrence of z do not interfere with the substitution $[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}]$, i.e., there is a formula θ such that

$$\varphi[\pi_i(\langle z_1, z_2 \rangle)/z] = \psi[\tilde{t}/\tilde{x}, \tilde{u}] = \theta[\tilde{t}/\tilde{x}, \tilde{u}/\tilde{y}, \langle z_1, z_2 \rangle/z]$$

we can simply apply the induction hypothesis on the subproof and conclude with one application of \times_η .

- * If we have that z clashes with a variable of \tilde{x}, \tilde{y} , say x_j , but that $t_j = v[\pi_i(\langle z_1, z_2 \rangle)/x_j]$ for some term v . Then we can apply the induction hypothesis with the matching tuples of terms \tilde{t}, x_i and \tilde{u}, t_i and conclude by applying the β rule.
- * Otherwise, the occurrence of z does interfere with the substitution in such a way that we have, say $t_j = \langle z_1, z_2 \rangle$. Then we can apply the induction hypothesis on the subproof with the matching tuples of terms \tilde{x}, z_i and $\tilde{y}, \pi_i(u_j)$ and conclude by applying the β rule.

□

One easy consequence of the above is Lemma 17:

PROOF OF LEMMA 17. Combine Lemma 19 and Lemma 18.

□

Another consequence is the following corollary, which will be used later in this section:

COROLLARY 20. *The following rule is polytime admissible:*

$$\frac{\Theta, t \in u \vdash \Delta}{\Theta \vdash \neg t \hat{=} u, \Delta}$$

PROOF. Recall that $t \hat{=} u$ expands to $\exists x \in u. x \equiv t$, so that $\neg t \hat{=} u$ is $\forall x \in u. \neg(x \equiv t)$. So we have

$$\text{Lemma 17} \frac{\Theta, t \in u \vdash \Delta}{\Theta, x \in u \vdash \neg(x \equiv t), \Delta} \quad \forall \frac{}{\Theta \vdash \neg t \hat{=} u, \Delta}$$

□

F.3 Proof of Lemma 6

We now recall the claim of admissibility concerning rules for “moving down in an equivalence”. Recall that these use the notation for quantifying on a path below an object, defined in the body.

LEMMA 6. *Assume p is a subtype occurrence for the type of the term o' . The following is polytime admissible*

$$\frac{\Theta \vdash \Delta, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}{\Theta, z \in r \vdash \Delta, \exists z' \in_{mp} o'. z \equiv_{T'} z'}$$

Furthermore, the size of the output proof is at most the size of the input proof.

PROOF. We proceed by induction over the input proof of $\Theta \vdash \Delta, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'$ and make a case distinction according to which rule was applied last. All cases are straightforward, save for one: when a \exists rule is applied on the formula $\exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'$. Let us only detail that one.

In that case, the last step has shape

$$\frac{\Theta \vdash \Delta^{\text{EL}}, r \equiv_{\text{Set}(T')} w, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}{\Theta \vdash \Delta^{\text{EL}}, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}$$

and because $r \equiv_{\text{Set}(T')} w$ is AL, we can further infer that the corresponding proof tree starts as follows

$$\frac{\frac{\Theta, z \in r \vdash \Delta^{\text{EL}}, z \hat{=} w, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r' \quad \vdots}{\Theta \vdash \Delta^{\text{EL}}, r \subseteq w, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'} \quad \frac{}{\Theta \vdash \Delta^{\text{EL}}, w \subseteq r, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}}{\exists \frac{\Theta \vdash \Delta^{\text{EL}}, r \equiv_{\text{Set}(T')} w, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}{\Theta \vdash \Delta^{\text{EL}}, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}}$$

In particular, we have a strictly smaller subproof of

$$\Theta, z \in r \vdash \Delta^{\text{EL}}, z \hat{=} w, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'$$

Applying the induction hypothesis, we get a proof of

$$\Theta, z \in r \vdash \Delta^{\text{EL}}, z \hat{=} w, \exists z' \in_{mp} o'. z \equiv_{\text{Set}(T')} z'$$

Recall that there is a max specialization of $\exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'$. Using that same specialization and the admissibility of the \exists rule that allows to perform a non-maximal specialization (Lemma 15), we can then obtain a proof with conclusion

$$\Theta, z \in r \vdash \Delta^{\text{EL}}, \exists r' \in_p o'. z \hat{=} r, \exists z' \in_{mp} o'. z \equiv_{\text{Set}(T')} z'$$

which concludes our argument, since $\exists r \in_p o'. z \hat{=} r$ and $\exists z' \in_{mp} o'. z \equiv_{\text{Set}(T')} z'$ are syntactically the same. \square

F.4 Proof of Lemma 7

LEMMA 7. *The following is polytime admissible (where p is a subtype occurrence of the type of o')*

$$\frac{\Theta \vdash \Delta, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}{\Theta \vdash \Delta, \exists r' \in_p o'. \forall z \in a. z \hat{=} r \leftrightarrow z \hat{=} r'}$$

PROOF. By induction over the shape of the input derivation, making a case distinction according to the last rule applied. All cases are trivial, except of the case of the rule \exists where $\exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'$ is the main formula. So let us focus on that one.

In that case, the proof necessarily has shape

$$\frac{\frac{\frac{\Theta, y \in w \vdash \Delta^{\text{EL}}, y \hat{=} r, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}{\Theta \vdash \Delta^{\text{EL}}, w \subseteq_{T'} r, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'} \quad \vee \quad \frac{\Theta, x \in r \vdash \Delta^{\text{EL}}, x \hat{=} w, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}{\Theta \vdash \Delta^{\text{EL}}, r \subseteq_{T'} w, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'}}{\wedge \quad \Theta \vdash \Delta^{\text{EL}}, r \equiv_{\text{Set}(T')} w, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'} \quad \exists \quad \Theta \vdash \Delta^{\text{EL}}, \exists r' \in_p o'. r \equiv_{\text{Set}(T')} r'$$

Applying the induction hypothesis to the leaves of that proof, we obtain two proofs of

$$\Theta, y \in w \vdash \Delta^{\text{EL}}, y \hat{=} r, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r' \quad \text{and} \quad \Theta, x \in r \vdash \Delta^{\text{EL}}, x \hat{=} w, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'$$

and we can conclude using the admissibility of weakening and Corollary 20 twice and replaying the $\wedge/\vee/\exists$ steps in the appropriate order (half of the proof derivation is elided below to save space):

$$\begin{array}{c} \text{Corollary 20} \quad \frac{\Theta, y \in w \vdash \Delta^{\text{EL}}, y \hat{=} r, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'}{\Theta \vdash \Delta^{\text{EL}}, \neg(y \hat{=} w), y \hat{=} r, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'} \\ \vee \quad \frac{\Theta \vdash \Delta^{\text{EL}}, \neg(y \hat{=} w), y \hat{=} r, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'}{\Theta \vdash \Delta^{\text{EL}}, y \hat{=} w \rightarrow y \hat{=} r, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'} \\ \wedge \quad \frac{\Theta \vdash \Delta^{\text{EL}}, y \hat{=} w \rightarrow y \hat{=} r, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'}{\Theta \vdash \Delta^{\text{EL}}, y \hat{=} r \leftrightarrow y \hat{=} w, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'} \\ \text{Lemma 12} \quad \frac{\Theta \vdash \Delta^{\text{EL}}, y \hat{=} r \leftrightarrow y \hat{=} w, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'}{\Theta, y \in a \vdash \Delta^{\text{EL}}, y \hat{=} r \leftrightarrow y \hat{=} w, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'} \\ \vee \quad \frac{\Theta \vdash \Delta^{\text{EL}}, \forall z \in a. z \hat{=} r \leftrightarrow z \hat{=} w, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'}{\Theta \vdash \Delta^{\text{EL}}, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r'} \\ \exists \quad \Theta \vdash \Delta^{\text{EL}}, \exists r' \in_p o'. \forall z \in a, z \hat{=} r \leftrightarrow z \hat{=} r' \end{array}$$

\square

G PROOF OF THE MAIN THEOREM FOR NON-SET TYPES

Recall again our main result:

THEOREM 2 (EFFECTIVE IMPLICIT TO EXPLICIT FOR NESTED DATA). *Given a witness for an implicit definition of o in terms of \vec{i} up to extensionality relative to Δ_0 $\varphi(\vec{i}, \vec{a}, o)$, one can compute NRC expression E such that for any \vec{i}, \vec{a} and o , if $\varphi(\vec{i}, \vec{a}, o)$ then $E(\vec{i}) = o$. Furthermore, if the witness is focused, this can be done in polynomial time.*

In the body of the paper, we gave a proof for the case where the type of the defined object is $\text{Set}(T)$ for any T . We now discuss the remaining cases: the base case and the inductive case for product types.

So assume we are given an implicit definition $\varphi(\vec{i}, \vec{a}, o)$ and a focused witness, and proceed by induction over the type o :

- If o has type Unit , then, since there is only one inhabitant in type Unit , then we can take our explicit definition to be the corresponding NRC expression $()$.
- If o has type \mathbb{U} , then using interpolation on the entailment $\varphi(\vec{i}, \vec{a}, o) \rightarrow (\varphi(\vec{i}, \vec{a}', o') \rightarrow o =_{\mathbb{U}} o')$, we obtain $\theta(\vec{i}, o)$ with $\varphi(\vec{i}, \vec{a}, o) \rightarrow \theta(\vec{i}, \vec{a}, o)$ and $\theta(\vec{i}, o) \wedge \varphi(\vec{i}, \vec{a}, o') \rightarrow o =_{\mathbb{U}} o'$. But then we know that φ implies o is a subobject of \vec{i} : otherwise we could find a model that contradicts the entailment. There is a NRC definition $A(\vec{i})$ that collects all of the \mathbb{U} -elements lying beneath \vec{i} . We can then take $E(\vec{i}) = \text{GET}_T(\{x \in A(\vec{i}) \mid \theta(\vec{i}, x)\})$ as our $\text{NRC}[\text{GET}]$ definition of o . The correctness of E follows from the properties of θ above.
- If o has type $T_1 \times T_2$, recalling the definition of $\equiv_{T_1 \times T_2}$, we have a derivation of

$$\varphi(\vec{i}, \vec{a}, o) \wedge \varphi(\vec{i}, \vec{a}, o') \vdash \pi_1(o) \equiv_{T_1} \pi_1(o') \wedge \pi_2(o) \equiv_{T_2} \pi_2(o')$$

By Lemma 13, we have proofs of

$$\varphi(\vec{i}, \vec{a}, o) \wedge \varphi(\vec{i}, \vec{a}, o') \vdash \pi_i(o) \equiv_{T_i} \pi_i(o')$$

for $i \in \{1, 2\}$. Take o_1 and o_2 to be fresh variables of types T_1 and T_2 . Take $\tilde{\varphi}(\vec{i}, \vec{a}, o_1, o_2)$ to be $\varphi(\vec{i}, \vec{a}, \langle o_1, o_2 \rangle)$. By substitutivity (the admissible rule given by Lemma 16) and applying the \times_β rule, we have focused proofs of

$$\varphi(\vec{i}, \vec{a}, \langle o_1, o_2 \rangle) \wedge \varphi(\vec{i}, \vec{a}, \langle o'_1, o'_2 \rangle) \vdash o_i \equiv_{T_i} o'_i$$

We can apply our inductive hypothesis to obtain a definition $E_i^{\text{IH}}(\vec{i})$ for both $i \in \{1, 2\}$. We can then take our explicit definition to be $\langle E_1^{\text{IH}}(\vec{i}), E_2^{\text{IH}}(\vec{i}) \rangle$.

H VARIANT OF PARAMETER COLLECTION THEOREM, THEOREM 8, FOR PARAMETERIZED DEFINABILITY IN FIRST-ORDER LOGIC

Our paper has focused on the setting of nested relations, phrasing our results in terms of the language NRC. We indicated in the conclusion of the paper that there is a variant of the NRC parameter collection theorem, Theorem 8, for the broader context of first-order logic. In fact, this first-order version of the result provided the intuition for the theorem. In this section we present this variant.

We consider first-order logic with equality and without function symbols, which also excludes nullary function symbols, that is, individual constants, whose role is just taken by free individual variables. Specifically, we consider first-order formulas with the following syntax

$$\varphi, \psi ::= P(\vec{x}) \mid \neg P(\vec{x}) \mid x = y \mid x \neq y \mid \top \mid \perp \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \forall x \varphi \mid \exists x \varphi.$$

On top of this, we give some “syntactic sugar”. We define $\neg\varphi$ by induction over φ , dualizing every connective, including the quantifiers, and removing doubled negation. We define implication $\varphi \rightarrow \psi$ as an abbreviation of $\neg\varphi \vee \psi$ and bi-implication $\varphi \leftrightarrow \psi$ as an abbreviation of $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. The set of free variables occurring in a formula φ is denoted by $FV(\varphi)$ and the set of predicates occurring in φ by $PRED(\varphi)$.

Figure 4 shows our proof system for first-order logic. It is identical to a system from the prior literature² Like the focused proof system we used in the body of the paper for Δ_0 formulas, it is a 1-sided calculus. The formulas other than Γ in the premise are the *active formulas* of the rule, while the *principal formulas* are the other formulas in its conclusion. The complementary principal formulas in Ax have to be literals. The replacement of symbols induced by equality with REPL is only performed on negative literals.

$$\begin{array}{c} \text{AX} \frac{}{\vdash \Gamma, \varphi, \neg\varphi} \quad \varphi \text{ AN ATOM} \quad \top \frac{}{\vdash \Gamma, \top} \quad \wedge \frac{\vdash \Gamma, \varphi_1 \quad \vdash \Gamma, \varphi_2}{\vdash \Gamma, \varphi_1 \wedge \varphi_2} \quad \vee \frac{\vdash \Gamma, \varphi_1, \varphi_2}{\vdash \Gamma, \varphi_1 \vee \varphi_2} \\ \\ \forall \frac{\vdash \Gamma, \varphi[y/x]}{\vdash \Gamma, \forall x \varphi} \quad y \notin FV(\Gamma, \forall x \varphi) \quad \exists \frac{\vdash \Gamma, \varphi[t/x], \exists x \varphi}{\vdash \Gamma, \exists x \varphi} \\ \\ \text{REF} \frac{\vdash t \neq t, \Gamma}{\vdash \Gamma} \quad \text{REPL} \frac{\vdash t \neq u, \varphi[u/x], \varphi[t/x], \Gamma}{\vdash t \neq u, \varphi[t/x], \Gamma} \quad \varphi \text{ A NEGATIVE LITERAL} \end{array}$$

Figure 4: One-sided sequent calculus for first-order logic with equality.

As in the body of the paper, a proof tree or derivation is a tree whose nodes are labelled with sequents, such that the labels of the children of a given node and that of the node itself are the premises and conclusion, resp., of an instance of a rule from Figure 4. The *conclusion* of a proof tree is the sequent that labels its root. The proof system is closed under cut, weakening and contraction. Closure under contraction in particular makes it suited as basis for “root-first” proof search. Read in this “bottom-up” way, the \exists rule states that a disjunction with an existentially quantified formula can be proven if the extension of the disjunction by a copy of the formula where the formerly quantified variable x is instantiated with an arbitrary variable t can be proven. The existentially quantified formula is retained in the premise and may be used to add further instances by applying \exists again in the course of the proof.

Soundness of the rules is straightforward. For example the \exists rule could be read as stating that if we deduce a disjunction in which one disjunct is a formula φ with t in it, then we can deduce the same disjunction but with some occurrences of t replaced in that disjunct with an existentially quantified variable. Completeness of the proof system can also be proven by a standard Henkin-style construction: indeed, since this is really ordinary first-order logic, there are proofs in the literature for systems that are very similar to this one: [28, 38].

The system in Figure 4 is the analog of our higher-level system in the body of the paper. We also have a restricted notion of proof, which admit more efficient algorithms, that we refer to as *focused proofs*. They are analogous to the focused proof system for Δ_0 formulas in the body of the paper. But in this first-order context we can define focused more easily, as an extra condition on proofs in the system. We characterize a proof as *FO-focused* if no application of AX, \top , \exists , REF, REPL contains in its conclusion a formula whose top-level connective is

²G3c+REF+REPL [29, 38], in the one-sided form of GS3, discussed in Chapter 3 of [38], which reduces the number of rules.

\vee , \wedge or \forall . This property may be either incorporated directly into a “root-first” proof procedure by constraining rule applications or it may be ensured by converting an arbitrary given proof tree to a FO-focused proof tree with the same ultimate consequence. This conversion is quite straightforward, but may increase the proof size exponentially.

We now discuss our generalization of the NRC Parameter Collection Theorem from the body of the paper to this first-order setting. The concept of explicit definition can be generalized to *definition up to parameters and disjunction*: A family of formulas $\chi_i(\vec{z}, \vec{y}, \vec{r})$, $1 \leq i \leq n$, provides an *explicit definition up to parameters and disjunction* of a formula $\lambda(\vec{z}, \vec{l})$ relative to a formula φ if

$$\varphi \models \bigvee_{i=1}^n \exists \vec{y} \forall \vec{z} (\lambda(\vec{z}, \vec{l}) \leftrightarrow \chi_i(\vec{z}, \vec{y}, \vec{r})). \quad (\star)$$

The entailment (\star) is considered with restrictions on the predicates and variables permitted to occur in the χ_i . In the simplest case, $\lambda(\vec{z}, \vec{l})$ is an atomic formula $p(\vec{z})$ with a predicate that is permitted in φ but not in the χ_i . The predicate p is then said to be *explicitly definable up to parameters and disjunction* with respect to φ [8].

The disjunction over a finite family of formulas χ_i can be consolidated into a single quantified biconditional as long as the domain has size at least 2 in every model of φ . Notice that if φ has only finite models, then by the compactness theorem of first-order logic, the size of models must be bounded. In such cases every formula λ is definable with sufficiently many parameters.

We can now state our analog of the Parameter Collection Theorem, Theorem 8.

THEOREM 21. *Let φ , ψ , $\lambda(\vec{z}, \vec{l})$, and $\rho(\vec{z}, \vec{y}, \vec{r})$ be first-order formulas such that*

$$\varphi \wedge \psi \models \exists \vec{y} \forall \vec{z} (\lambda(\vec{z}, \vec{l}) \leftrightarrow \rho(\vec{z}, \vec{y}, \vec{r})).$$

Then there exist first-order formulas $\chi_i(\vec{z}, \vec{v}_i, \vec{c}_i)$, $1 \leq i \leq n$, such that

- (1) $\varphi \wedge \psi \models \bigvee_{i=1}^n \exists \vec{v}_i \forall \vec{z} (\lambda(\vec{z}, \vec{l}) \leftrightarrow \chi_i(\vec{z}, \vec{v}_i, \vec{c}_i))$,
- (2) $\vec{c}_i \subseteq (FV(\varphi) \cup \vec{l}) \cap (FV(\psi) \cup \vec{r})$,
- (3) $PRED(\chi_i) \subseteq (PRED(\varphi) \cup PRED(\lambda)) \cap (PRED(\psi) \cup PRED(\rho))$.

Moreover, given a FO-focused proof of the precondition with the system of Fig. 4, a family of formulas χ_i , $1 \leq i \leq n$, with the claimed properties can be computed in polynomial time in the size of the proof tree.

In the theorem statement, the free variables of λ and ρ are \vec{z} , \vec{y} and \vec{l} . The precondition supposes an explicit definition ρ of λ up to parameters with respect to a conjunction $\varphi \wedge \psi$. The conclusion then claims that one can effectively compute another definition of λ with respect to $\varphi \wedge \psi$ that is up to parameters and disjunction and has a constrained signature: free variables and predicates must occur in at least one of the “left side” formulas φ and λ and also in at least one of the “right side” formulas ψ and ρ . In other words, the theorem states that if SIG_L and SIG_R are “left” and “right” signatures such that φ and λ are over SIG_L , ψ and ρ are over SIG_R , and ρ provides an explicit definition of λ up to parameters with respect to $\varphi \wedge \psi$, then one can effectively compute another definition of λ with respect to $\varphi \wedge \psi$ that is up to parameters and disjunction and is just over the intersection of the signatures SIG_L and SIG_R .

We now prove Theorem 21 by induction on the depth of the proof tree, generalizing the constructive proof method for Craig interpolation often called Maehara’s method [33, 35, 38]. To simplify the presentation we assume that the tuples \vec{z} and \vec{y} in the theorem statement each consist of a single variable z and y , respectively. The generalization of our argument to tuples of variables is straightforward.

To specify conveniently the construction steps of the family of formulas χ_i we introduce the following concept: A *pre-defining equivalence up to parameters and disjunction* (briefly PDEPD) for a formula $\lambda(z, \vec{l})$ is a formula δ built up from formulas of the form $\forall z (\lambda(z, \vec{l}) \leftrightarrow \chi(z, \vec{p}))$ (where the left side is always the same formula $\lambda(z, \vec{l})$ but the right sides $\chi(z, \vec{p})$ may differ) and a finite number of applications of disjunction and existential quantification upon variables from the vectors \vec{p} of the right sides. The empty disjunction \perp is allowed as a special case of a PDEPD. By rewriting with the equivalence $\exists v (\delta_1 \vee \delta_2) \equiv \exists v \delta_1 \vee \exists v \delta_2$, any PDEPD for λ can be efficiently transformed into the form $\bigvee_{i=1}^n \exists \vec{v}_i \forall z (\lambda(z, \vec{l}) \leftrightarrow \chi_i(z, \vec{v}_i, \vec{r}_i))$ for some natural number $n \geq 0$. That is, although a PDEPD has in general not the syntactic form of the disjunction of quantified biconditionals in the theorem statement (thus “pre-”), it corresponds to such a disjunction. The more generous syntax will be convenient in the induction. The sets of additional variables \vec{l} and \vec{p} in the biconditionals $\lambda(z, \vec{l}) \leftrightarrow \chi(z, \vec{p})$ can overlap, but the overlap will be top-level variables that never get quantified. Although we have defined the notion of PDEPD for a general λ , in the proof we just consider PDEPDs for the formula λ from the theorem statement.

For PDEPDs δ we provide analogs to FV and $PRED$ that only yield free variables and predicates of δ that occur in a right side of its biconditionals, which helps to express the restrictions by definability properties that constrains the signature of exactly those right sides. Recall that we refer of these right sides as subformulas $\chi(z, \vec{p})$. For PDEPD δ , define $PRED^{RHS}(\delta)$ as the set of the predicate symbols that occur in a subformula $\chi(z, \vec{p})$ of δ and define $FV^{RHS}(\delta)$ as the set of all variables that occur in a subformula $\chi(z, \vec{p})$ of δ and are free in δ . In other words, $FV^{RHS}(\delta)$ is the set of all variables p in the vectors \vec{p} of the subformulas $\chi(z, \vec{p})$ that have an occurrence in δ which is not in the scope of a quantifier $\exists p$. If, for example $\delta = \bigvee_{i=1}^n \exists \vec{v}_i \forall z (\lambda(z, \vec{l}) \leftrightarrow \chi_i(z, \vec{v}_i, \vec{r}_i))$, then $FV^{RHS}(\delta)$ is the set of all variables in the vectors \vec{r}_i , for $1 \leq i \leq n$.

To build up PDEPDs we provide an operation that only affects the right sides of the biconditionals, a restricted form of existential quantification. It is for use in interpolant construction to convert a free variable in right sides that became illegal into an existentially quantified parameter. For variables p, y define $\delta[y/p]^{RHS}$ as δ after substituting all occurrences of p that are within a right side $\chi(z, \vec{p})$ and

are free in δ with y . Define $\exists^{\text{RHS}} p \delta$ as shorthand for $\exists y \delta[y/p]^{\text{RHS}}$, where y is a fresh variable. Clearly $\delta \models \exists^{\text{RHS}} p \delta$ and p has no free occurrences in $\exists^{\text{RHS}} p \delta$ that are in any of the right side formulas $\chi(z, \vec{p})$, i.e., $p \notin FV^{\text{RHS}}(\exists^{\text{RHS}} p \delta)$. Occurrences of p in the left sides $\lambda(z, \vec{l})$, if p is a member of \vec{l} , are untouched in $\exists^{\text{RHS}} p \delta$. If p is not in \vec{l} , then $\exists^{\text{RHS}} p \delta$ reduces to ordinary existential quantification $\exists y \delta[y/p]$.

We introduce the following symbolic shorthand for the parametric definition on the right side in the theorem's precondition.

$$\mathcal{G} := \exists y \forall z (\lambda(z, \vec{l}) \leftrightarrow \rho(z, y, \vec{r})).$$

Note that our proof rules are such that if we have a proof (FO-focused or not) that our original top-level “global” parametric definition is implied by some formula, then every one-sided sequent in the proof must include that parametric definition in it. This is because the rules that eliminate a formula when read “bottom-up” cannot apply to that parametric definition, whose outermost logic operator is the existential quantifier. Thus, in our inductive argument, we can assume that \mathcal{G} is always present.

We write

$$\vdash \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle,$$

where $\vdash \Gamma_L, \Gamma_R, \mathcal{G}$ is a sequent, partitioned into three components, multisets Γ_L and Γ_R of formulas and the formula \mathcal{G} from the theorem's hypothesis, θ is a formula and \mathcal{D} is a PDEPD, to express that the following properties hold:

- I1. $\vdash \Gamma_R \vee \theta$.
- I2. $\vdash \neg \theta \vee \Gamma_L \vee \mathcal{D}$.
- I3. $PRED(\theta) \subseteq (PRED(\Gamma_L) \cup PRED(\lambda)) \cap (PRED(\Gamma_R) \cup PRED(\rho))$.
- I4. $FV(\theta) \subseteq (FV(\Gamma_L) \cup \vec{l}) \cap (FV(\Gamma_R) \cup \vec{r})$.
- I5. \mathcal{D} is a PDEPD for $\lambda(z, \vec{l})$.
- I6. $PRED^{\text{RHS}}(\mathcal{D}) \subseteq (PRED(\Gamma_L) \cup PRED(\lambda)) \cap (PRED(\Gamma_R) \cup PRED(\rho))$.
- I7. $FV^{\text{RHS}}(\mathcal{D}) \subseteq (FV(\Gamma_L) \cup \vec{l}) \cap (FV(\Gamma_R) \cup \vec{r})$.

For a given proof with conclusion $\vdash \neg \varphi, \neg \psi, \mathcal{G}$, corresponding to the hypothesis $\varphi \wedge \psi \models \mathcal{G}$ of the theorem, we show the construction of a formula θ and PDEPD \mathcal{D} such that

$$\vdash \neg \varphi; \neg \psi; \mathcal{G} : \langle \theta, \mathcal{D} \rangle.$$

From properties 1–2 and 5–7 it is then straightforward to read off that the formula $\bigvee_{i=1}^n \exists \vec{v}_i \forall \vec{z} (\lambda(\vec{z}, \vec{l}) \leftrightarrow \chi_i(\vec{z}, \vec{v}_i, \vec{r}_i))$ obtained from \mathcal{D} by propagating existential quantifiers inwards is as claimed in the theorem's conclusion.

Formula θ plays an auxiliary role in the induction. For the overall conclusion of the proof it is like a Craig interpolant of ψ and $\varphi \rightarrow \mathcal{D}$, but slightly weaker syntactically constrained by taking λ and ρ into account: $FV(\theta) \subseteq ((FV(\varphi) \cup \vec{l}) \cap (FV(\psi) \cup \vec{r}))$ and $PRED(\theta) \subseteq (PRED(\varphi) \cup PRED(\lambda)) \cup PRED(\psi) \cup PRED(\rho)$.

As basis of the induction, we have to show constructions of θ and \mathcal{D} such that $\vdash \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle$ holds for Ax and \top , considering each possibility in which the principal formula(s) can be in Γ_L or Γ_R . For the induction step, there are a number of subcases, according to which rule is last applied and which of the partitions Γ_L, Γ_R or \mathcal{G} contain the principal formula(s). We first discuss the most interesting case, the induction step where \mathcal{G} is the principal formula. This case is similar to the most interesting case in the NRC Parameter Collection Theorem, covered in the body of the paper.

Case where the principal formula is \mathcal{G} . We now give more detail on the most complex case. If the principal formula of a conclusion $\vdash \Gamma_L; \Gamma_R; \mathcal{G}$ is \mathcal{G} , then the rule that is applied must be \exists . From the FO-focused property of the proof it follows that the derivation tree ending in $\vdash \Gamma_L; \Gamma_R; \mathcal{G}$ must have the following shape, for some $u \notin FV(\Gamma_L, \Gamma_R, \mathcal{G})$ and $w \neq u$. Note that u could be either a top-level variable from λ , i.e., a member of \vec{l} , or one introduced during the proof.

$$\begin{array}{c} \vee \frac{\vdash \Gamma_L, \neg \lambda(u, \vec{l}), \Gamma_R, \rho(u, w, \vec{r}), \mathcal{G}}{\vdash \Gamma_L, \Gamma_R, \lambda(u, \vec{l}) \rightarrow \rho(u, w, \vec{r}), \mathcal{G}} \quad \vee \frac{\vdash \Gamma_L, \lambda(u, \vec{l}), \Gamma_R, \neg \rho(u, w, \vec{r}), \mathcal{G}}{\vdash \Gamma_L, \lambda(u, \vec{l}), \Gamma_R, \rho(u, w, \vec{r}) \rightarrow \lambda(u, \vec{l}), \mathcal{G}} \\ \wedge \frac{}{\vdash \Gamma_L, \Gamma_R, \lambda(u, \vec{l}) \leftrightarrow \rho(u, w, \vec{r}), \mathcal{G}} \\ \vee \frac{}{\vdash \Gamma_L, \Gamma_R, \forall z (\lambda(z, \vec{l}) \leftrightarrow \rho(z, w, \vec{r})), \mathcal{G}} \\ \exists \frac{}{\vdash \Gamma_L, \Gamma_R, \mathcal{G}} \end{array}$$

The important point is that the two “leaves” of the above tree are both sequents where we can apply our induction hypothesis. Taking into account the partitioning of the sequents at the bottom conclusion and the top premises in this figure, we can express the induction step in the form of a “macro” rule that specifies the how we constructed the required θ and \mathcal{D} for the conclusion, making use of the θ_1, \mathcal{D}_1 and θ_2, \mathcal{D}_2 that we get by applying the induction hypothesis to each of the two premises.

$$\frac{\vdash \Gamma_L, \neg \lambda(u, \vec{l}); \Gamma_R, \rho(u, w, \vec{r}); \mathcal{G} : \langle \theta_1, \mathcal{D}_1 \rangle \quad \vdash \Gamma_L, \lambda(u, \vec{l}); \Gamma_R, \neg \rho(u, w, \vec{r}); \mathcal{G} : \langle \theta_2, \mathcal{D}_2 \rangle}{\vdash \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle},$$

where u is as above. The values of θ and \mathcal{D} – the new formula and definition that we are building – will depend on occurrences of w , and we give their construction in cases below:

(i) If $w \notin FV(\Gamma_L) \cup \vec{l}$ or $w \in FV(\Gamma_R) \cup \vec{r}$, then

$$\begin{aligned}\theta &:= \forall u (\theta_1 \vee \theta_2). \\ \mathcal{D} &:= \exists^{\text{RHS}} u \mathcal{D}_1 \vee \exists^{\text{RHS}} u \mathcal{D}_2 \vee \forall z (\lambda(z, \vec{l}) \leftrightarrow \theta_2[z/u]).\end{aligned}$$

(ii) Else it holds that $w \in FV(\Gamma_L) \cup \vec{l}$ and $w \notin FV(\Gamma_R) \cup \vec{r}$. Then

$$\begin{aligned}\theta &:= \forall w \forall u (\theta_1 \vee \theta_2). \\ \mathcal{D} &:= \exists^{\text{RHS}} w (\exists^{\text{RHS}} u \mathcal{D}_1 \vee \exists^{\text{RHS}} u \mathcal{D}_2 \vee \forall z (\lambda(z, \vec{l}) \leftrightarrow \theta_2[z/u])).\end{aligned}$$

We now verify that $\vdash \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta', \mathcal{D}' \rangle$, that is, properties I1–I7, hold. The proofs for the individual properties are presented in tabular form, with explanations annotated in the side column, where *IH* stands for *induction hypothesis*. We concentrate on the case (i) and indicate the modifications of the proofs for case (ii) in remarks, where we refer to the values of θ and \mathcal{D} for that case in terms of the values for the case (i) as $\forall w \theta$ and $\exists^{\text{RHS}} w \mathcal{D}$. In the proofs of the semantic properties I1 and I2 we let sequents stand for the disjunction of their members.

Property I1:

- (1) $\vdash \Gamma_R \vee \rho(u, w, \vec{r}) \vee \theta_1$. IH
- (2) $\vdash \Gamma_R \vee \neg \rho(u, w, \vec{r}) \vee \theta_2$. IH
- (3) $\vdash \Gamma_R \vee \theta_1 \vee \theta_2$. by (1) and (2)
- (4) $\vdash \Gamma_R \vee \forall u (\theta_1 \vee \theta_2)$. by (3) since $u \notin FV(\Gamma_R)$
- (5) $\vdash \Gamma_R \vee \theta$. by (4) and def. of θ

For case (ii), it follows from the precondition $w \notin FV(\Gamma_R)$ and step (5) that $\vdash \Gamma_R \vee \forall w \theta$.

Property I2:

- (1) $\vdash \neg \theta_1 \vee \Gamma_L \vee \neg \lambda(u, \vec{l}) \vee \mathcal{D}_1$. IH
- (2) $\vdash \neg \theta_2 \vee \Gamma_L \vee \lambda(u, \vec{l}) \vee \mathcal{D}_2$. IH
- (3) $\vdash \neg (\theta_1 \vee \theta_2) \vee \Gamma_L \vee \neg \lambda(u, \vec{l}) \vee \mathcal{D}_1 \vee \mathcal{D}_2$. by (1)
- (4) $\vdash \neg (\theta_1 \vee \theta_2) \vee \Gamma_L \vee \neg \lambda(u, \vec{l}) \vee \mathcal{D}_1 \vee \mathcal{D}_2 \vee \theta_2$. by (3)
- (5) $\vdash \neg \forall u (\theta_1 \vee \theta_2) \vee \Gamma_L \vee \neg \lambda(u, \vec{l}) \vee \mathcal{D}_1 \vee \mathcal{D}_2 \vee \theta_2$. by (4)
- (6) $\vdash \neg \forall u (\theta_1 \vee \theta_2) \vee \neg \theta_2 \vee \Gamma_L \vee \lambda(u, \vec{l}) \vee \mathcal{D}_1 \vee \mathcal{D}_2$. by (2)
- (7) $\vdash \neg \forall u (\theta_1 \vee \theta_2) \vee \Gamma_L \vee \mathcal{D}_1 \vee \mathcal{D}_2 \vee (\lambda(u, \vec{l}) \leftrightarrow \theta_2)$. by (6) and (5)
- (8) $\vdash \neg \forall u (\theta_1 \vee \theta_2) \vee \Gamma_L \vee \exists^{\text{RHS}} u \mathcal{D}_1 \vee \exists^{\text{RHS}} u \mathcal{D}_2 \vee (\lambda(u, \vec{l}) \leftrightarrow \theta_2)$. by (7)
- (9) $\vdash \neg \forall u (\theta_1 \vee \theta_2) \vee \Gamma_L \vee \exists^{\text{RHS}} u \mathcal{D}_1 \vee \exists^{\text{RHS}} u \mathcal{D}_2 \vee \forall z (\lambda(z, \vec{l}) \leftrightarrow \theta_2[z/u])$. by (8) since $u \notin FV(\Gamma_L, \lambda(z, \vec{l}))$
- (10) $\vdash \neg \theta \vee \Gamma_L \vee \mathcal{D}$. by (9) and defs. of θ, \mathcal{D}

That $u \notin FV(\lambda(z, \vec{l}))$ follows from the precondition $u \notin FV(\mathcal{G})$. It is used in step (9) to justify that the substitution $[z/u]$ has only to be applied to θ_2 and not to $\lambda(z, \vec{l})$ and, in addition, to justify that $u \notin FV(\exists^{\text{RHS}} u \mathcal{D}_1)$ and $u \notin FV(\exists^{\text{RHS}} u \mathcal{D}_2)$, which follow from $u \notin FV(\lambda(z, \vec{l}))$ and the induction hypotheses that property I5 applies to \mathcal{D}_1 and \mathcal{D}_2 .

For case (ii), it follows from step (10) that $\vdash \neg \forall w \theta \vee \Gamma_L \vee \exists^{\text{RHS}} w \mathcal{D}$.

Property I3:

- (1) $\text{PRED}(\theta_1) \subseteq (\text{PRED}(\Gamma_L, \neg \lambda(u, \vec{l})) \cup \text{PRED}(\lambda)) \cap (\text{PRED}(\Gamma_R, \rho(u, w, \vec{r})) \cup \text{PRED}(\rho))$. IH
- (2) $\text{PRED}(\theta_2) \subseteq (\text{PRED}(\Gamma_L, \lambda(u, \vec{l})) \cup \text{PRED}(\lambda)) \cap (\text{PRED}(\Gamma_R, \neg \rho(u, w, \vec{r})) \cup \text{PRED}(\rho))$. IH
- (3) $\text{PRED}(\forall u (\theta_1 \vee \theta_2)) \subseteq (\text{PRED}(\Gamma_L) \cup \text{PRED}(\lambda)) \cap (\text{PRED}(\Gamma_R) \cup \text{PRED}(\rho))$. by (1), (2)
- (4) $\text{PRED}(\theta) \subseteq (\text{PRED}(\Gamma_L) \cup \text{PRED}(\lambda)) \cap (\text{PRED}(\Gamma_R) \cup \text{PRED}(\rho))$. by (3) and def. of θ

For case (ii) the property follows since $\text{PRED}(\theta) = \text{PRED}(\forall w \theta)$.

Property I4:

- (1) $FV(\theta_1) \subseteq (FV(\Gamma_L, \neg \lambda(u, \vec{l})) \cup \vec{l}) \cap (FV(\Gamma_R, \rho(u, w, \vec{r})) \cup \vec{r})$. IH
- (2) $FV(\theta_2) \subseteq (FV(\Gamma_L, \lambda(u, \vec{l})) \cup \vec{l}) \cap (FV(\Gamma_R, \neg \rho(u, w, \vec{r})) \cup \vec{r})$. IH
- (3) $FV(\theta_1) \subseteq (FV(\Gamma_L) \cup \vec{l} \cup \{u\}) \cap (FV(\Gamma_R) \cup \vec{r} \cup \{u, w\})$. by (1)
- (4) $FV(\theta_1) \subseteq (FV(\Gamma_L) \cup \vec{l} \cup \{u\}) \cap (FV(\Gamma_R) \cup \vec{r} \cup \{u, w\})$. by (2)
- (5) $FV(\forall u (\theta_1 \vee \theta_2)) \subseteq (FV(\Gamma_L) \cup \vec{l}) \cap (FV(\Gamma_R) \cup \vec{r})$. by (3), (4) and the precondition. $w \notin FV(\Gamma_L) \cup \vec{l}$ or $w \in FV(\Gamma_R) \cup \vec{r}$
- (6) $FV(\theta) \subseteq (FV(\Gamma_L) \cup \vec{l}) \cap (FV(\Gamma_R) \cup \vec{r})$. by (5) and def. of θ

For case (ii) step (5) has to be replaced by

$$FV(\forall w \forall u (\theta_1 \vee \theta_2)) \subseteq (FV(\Gamma_L) \cup \vec{l}) \cap (FV(\Gamma_R) \cup \vec{r}),$$

which follows just from (3) and (4). Instead of step (6) we then have $FV(\forall w \theta) \subseteq (FV(\Gamma_L) \cup \vec{l}) \cap (FV(\Gamma_R) \cup \vec{r})$.

Property I5: Immediate from the induction hypothesis and the definition of \mathcal{D} .

Property I6:

- (1) $PRED^{RHS}(\mathcal{D}_1) \subseteq (PRED(\Gamma_L, \neg\lambda(u, \vec{I})) \cup PRED(\lambda)) \cap (PRED(\Gamma_R, \rho(u, w, \vec{r})) \cup PRED(\rho)).$ IH
- (2) $PRED^{RHS}(\mathcal{D}_2) \subseteq (PRED(\Gamma_L, \lambda(u, \vec{I})) \cup PRED(\lambda)) \cap (PRED(\Gamma_R, \neg\rho(u, w, \vec{r})) \cup PRED(\rho)).$ IH
- (3) $PRED(\theta_2) \subseteq (PRED(\Gamma_L, \lambda(u, \vec{I})) \cup PRED(\lambda)) \cap (PRED(\Gamma_R, \neg\rho(u, w, \vec{r})) \cup PRED(\rho)).$ IH
- (4) $PRED^{RHS}(\exists^{RHS} u \mathcal{D}_1 \vee \exists^{RHS} u \mathcal{D}_2 \vee \forall z (\lambda(z, \vec{I}) \leftrightarrow \theta_2[z/u])) \subseteq$
 $(PRED(\Gamma_L) \cup PRED(\lambda)) \cap (PRED(\Gamma_R) \cup PRED(\rho))$ by (1)–(3)
- (5) $PRED^{RHS}(\mathcal{D}) \subseteq (PRED(\Gamma_L) \cup PRED(\lambda)) \cap (PRED(\Gamma_R) \cup PRED(\rho))$ by (4) and def. of \mathcal{D}

For case (ii) the property follows since $PRED^{RHS}(\mathcal{D}) = PRED^{RHS}(\exists^{RHS} w \mathcal{D})$.

Property I7:

- (1) $FV^{RHS}(\mathcal{D}_1) \subseteq (FV(\Gamma_L, \neg\lambda(u, \vec{I})) \cup \vec{I}) \cap (FV(\Gamma_R, \rho(u, w, \vec{r})) \cup \vec{r}).$ IH
- (2) $FV^{RHS}(\mathcal{D}_2) \subseteq (FV(\Gamma_L, \lambda(u, \vec{I})) \cup \vec{I}) \cap (FV(\Gamma_R, \neg\rho(u, w, \vec{r})) \cup \vec{r}).$ IH
- (3) $FV(\theta_2) \subseteq (FV(\Gamma_L, \lambda(u, \vec{I})) \cup \vec{I}) \cap (FV(\Gamma_R, \neg\rho(u, w, \vec{r})) \cup \vec{r}).$ IH
- (4) $FV^{RHS}(\mathcal{D}_1) \subseteq (FV(\Gamma_L) \cup \vec{I} \cup \{u\}) \cap (FV(\Gamma_R) \cup \vec{r} \cup \{u, w\}).$ by (1)
- (5) $FV^{RHS}(\mathcal{D}_2) \subseteq (FV(\Gamma_L) \cup \vec{I} \cup \{u\}) \cap (FV(\Gamma_R) \cup \vec{r} \cup \{u, w\}).$ by (2)
- (6) $FV(\theta_2) \subseteq (FV(\Gamma_L) \cup \vec{I} \cup \{u\}) \cap (FV(\Gamma_R) \cup \vec{r} \cup \{u, w\}).$ by (3)
- (7) $FV^{RHS}(\exists^{RHS} u \mathcal{D}_1 \vee \exists^{RHS} u \mathcal{D}_2 \vee \forall z (\lambda(z, \vec{I}) \leftrightarrow \theta_2[z/u])) \subseteq$ by (4), (5), (6) and the precondition. $w \notin FV(\Gamma_L) \cup \vec{I}$ or
 $(FV(\Gamma_L) \cup \vec{I}) \cap (FV(\Gamma_R) \cup \vec{r}).$ $w \in FV(\Gamma_R) \cup \vec{r}$
- (8) $FV^{RHS}(\mathcal{D}) \subseteq (FV(\Gamma_L) \cup \vec{I}) \cap (FV(\Gamma_R) \cup \vec{r}).$ by (7) and def. of \mathcal{D}

For case (ii) step (7) has to be replaced by

$$FV^{RHS}(\exists^{RHS} w (\exists^{RHS} u \mathcal{D}_1 \vee \exists^{RHS} u \mathcal{D}_2 \vee \forall z (\lambda(z, \vec{I}) \leftrightarrow \theta_2[z/u]))) \subseteq \\ (FV(\Gamma_L) \cup \vec{I}) \cap (FV(\Gamma_R) \cup \vec{r}),$$

which follows just from (4), (5), (6). Instead of step (8) we then have $FV^{RHS}(\exists^{RHS} w \mathcal{D}) \subseteq (FV(\Gamma_L) \cup \vec{I}) \cap (FV(\Gamma_R) \cup \vec{r})$.

This completes the verification of correctness, and thus ends our discussion of this case.

We now turn to the base of the induction along with the other inductive cases.

Cases where the principal formulas are in the Γ_L or Γ_R partition. The inductive cases where the principal formulas are in the Γ_L or Γ_R partition can be conveniently specified in the form of rules that lead from induction hypotheses of the form $\vdash \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle$ as premises to an induction conclusion of the same form. Base cases can there be taken just as such rules without premises. The axioms and rules shown below correspond to the those of the calculus, but replicated for each possible way in which the partitions Γ_L , Γ_R or \mathcal{G} of the conclusion may contain the principal formula(s). To verify that properties I1–I7 are preserved by each of the shown constructions is in general straightforward, such that we only have annotated a few subtleties that may not be evident.

- (1) Ax $\frac{}{\vdash \Gamma_L, \varphi, \neg\varphi; \Gamma_R; \mathcal{G} : \langle \top, \perp \rangle}$ φ an atom
- (2) Ax $\frac{}{\vdash \Gamma_L; \Gamma_R, \varphi, \neg\varphi; \mathcal{G} : \langle \perp, \perp \rangle}$ φ an atom
- (3) Ax $\frac{}{\vdash \Gamma_L, \varphi; \Gamma_R, \neg\varphi; \mathcal{G} : \langle \varphi, \perp \rangle}$ φ a literal
- (4) \top $\frac{}{\vdash \Gamma_L, \top; \Gamma_R; \mathcal{G} : \langle \top, \perp \rangle}$
- (5) \top $\frac{}{\vdash \Gamma_L; \Gamma_R, \top; \mathcal{G} : \langle \perp, \perp \rangle}$
- (6) \vee $\frac{\vdash \Gamma_L, \varphi_1, \varphi_2; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \Gamma_L, \varphi_1 \vee \varphi_2; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}$
- (7) \vee $\frac{\vdash \Gamma_L; \Gamma_R, \varphi_1, \varphi_2; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \Gamma_L; \Gamma_R, \varphi_1 \vee \varphi_2; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}$

$$(8) \quad \wedge \frac{\vdash \Gamma_L, \varphi_1; \Gamma_R; \mathcal{G} : \langle \theta_1, \mathcal{D}_1 \rangle \quad \vdash \Gamma_L, \varphi_2; \Gamma_R; \mathcal{G} : \langle \theta_2, \mathcal{D}_2 \rangle}{\vdash \Gamma_L, \varphi_1 \wedge \varphi_2; \Gamma_R; \mathcal{G} : \langle \theta_1 \wedge \theta_2, \mathcal{D}_1 \vee \mathcal{D}_2 \rangle}$$

$$(9) \quad \wedge \frac{\vdash \Gamma_L; \Gamma_R, \varphi_1; \mathcal{G} : \langle \theta_1, \mathcal{D}_1 \rangle \quad \vdash \Gamma_L; \Gamma_R, \varphi_2; \mathcal{G} : \langle \theta_2, \mathcal{D}_2 \rangle}{\vdash \Gamma_L; \Gamma_R, \varphi_1 \wedge \varphi_2; \mathcal{G} : \langle \theta_1 \vee \theta_2, \mathcal{D}_1 \vee \mathcal{D}_2 \rangle}$$

$$(10) \quad \exists \frac{\vdash \Gamma_L, \varphi[t/x], \exists x \varphi; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \Gamma_L, \exists x \varphi; \Gamma_R; \mathcal{G} : \langle \theta', \mathcal{D}' \rangle},$$

where the values of θ' and \mathcal{D}' depend on occurrences of t :

- If $t \in FV(\Gamma_L, \exists x \varphi) \cup \tilde{l}$, then $\theta' := \theta$ and $\mathcal{D}' := \mathcal{D}$.
- Else it holds that $t \notin FV(\Gamma_L, \exists x \varphi) \cup \tilde{l}$. Then $\theta' := \exists t \theta$ and $\mathcal{D}' := \exists^{\text{RHS}}_t \mathcal{D}$.

$$(11) \quad \exists \frac{\vdash \Gamma_L; \Gamma_R, \varphi[t/x], \exists x \varphi; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \Gamma_L; \Gamma_R, \exists x \varphi; \mathcal{G} : \langle \theta', \mathcal{D}' \rangle},$$

where the values of θ' and \mathcal{D}' depend on occurrences of t :

- If $t \in FV(\Gamma_R, \exists x \varphi) \cup \tilde{r}$, then $\theta' := \theta$ and $\mathcal{D}' := \mathcal{D}$.
- Else it holds that $t \notin FV(\Gamma_R, \exists x \varphi) \cup \tilde{r}$. Then $\theta' := \forall t \theta$ and $\mathcal{D}' := \exists^{\text{RHS}}_t \mathcal{D}$.

$$(12) \quad \forall \frac{\vdash \Gamma_L, \varphi[y/x]; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \Gamma_L, \forall x \varphi; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle} \quad y \notin FV(\Gamma_L, \forall x \varphi, \Gamma_R, \mathcal{G})$$

$$(13) \quad \forall \frac{\vdash \Gamma_L; \Gamma_R, \varphi[y/x]; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \Gamma_L; \Gamma_R, \forall x \varphi; \mathcal{G} : \langle \theta, \mathcal{D} \rangle} \quad y \notin FV(\Gamma_L, \Gamma_R, \forall x \varphi, \mathcal{G})$$

$$(14) \quad \text{REF} \frac{\vdash t \neq t, \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle} \quad t \in FV(\Gamma_L) \cup \tilde{l}$$

$$(15) \quad \text{REF} \frac{\vdash \Gamma_L; t \neq t, \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle} \quad t \notin FV(\Gamma_L) \cup \tilde{l}$$

$$(16) \quad \text{REPL} \frac{\vdash t \neq u, \varphi[u/x], \varphi[t/x], \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash t \neq u, \varphi[t/x], \Gamma_L; \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle} \quad \varphi \text{ a negative literal}$$

$$(17) \quad \text{REPL} \frac{\vdash t \neq u, \Gamma_L; \varphi[u/x], \varphi[t/x], \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash t \neq u, \Gamma_L; \varphi[t/x], \Gamma_R; \mathcal{G} : \langle \theta', \mathcal{D}' \rangle}, \quad \varphi \text{ a negative literal}$$

where the values of θ' and \mathcal{D}' depend on occurrences of t and u :

- If $t \notin FV(\varphi[t/x], \Gamma_R) \cup \tilde{r}$, then $\theta' := \theta$ and $\mathcal{D}' := \mathcal{D}$. In this subcase the precondition $t \notin FV(\varphi[t/x])$ implies that $x \notin FV(\varphi)$ and thus $\varphi[u/x] = \varphi[t/x]$.
- If $t, u \in FV(\varphi[t/x], \Gamma_R) \cup \tilde{r}$, then $\theta' := \theta \vee t \neq u$ and $\mathcal{D}' := \mathcal{D}$.
- Else it holds that $t \in FV(\varphi[t/x], \Gamma_R) \cup \tilde{r}$ and $u \notin FV(\varphi[t/x], \Gamma_R) \cup \tilde{r}$. Then $\theta' := \theta[t/u]$ and $\mathcal{D}' := \mathcal{D}[t/u]^{\text{RHS}}$. For this subcase, to derive property I1 it is used that the precondition $u \notin \varphi[t/x]$ implies that $\varphi[u/x][t/u] = \varphi[t/x]$.

$$(18) \quad \text{REPL} \frac{\vdash \Gamma_L; t \neq u, \varphi[u/x], \varphi[t/x], \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \Gamma_L; t \neq u, \varphi[t/x], \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle} \quad \varphi \text{ a negative literal}$$

$$(19) \quad \text{REPL} \frac{\vdash \varphi[u/x], \varphi[t/x], \Gamma_L; t \neq u, \Gamma_R; \mathcal{G} : \langle \theta, \mathcal{D} \rangle}{\vdash \varphi[t/x], \Gamma_L; t \neq u, \Gamma_R; \mathcal{G} : \langle \theta', \mathcal{D}' \rangle}, \quad \varphi \text{ a negative literal}$$

where the values of θ' and \mathcal{D}' depend on occurrences of t and u :

- If $t \notin FV(\varphi[t/x], \Gamma_L) \cup \tilde{l}$, then $\theta' := \theta$ and $\mathcal{D}' := \mathcal{D}$. In this subcase the precondition $t \notin FV(\varphi[t/x])$ implies that $x \notin FV(\varphi)$ and thus $\varphi[u/x] = \varphi[t/x]$.
- If $t, u \in FV(\varphi[t/x], \Gamma_L) \cup \tilde{l}$, then $\theta' := \theta \wedge t = u$ and $\mathcal{D}' := \mathcal{D}$.
- Else it holds that $t \in FV(\varphi[t/x], \Gamma_L) \cup \tilde{l}$ and $u \notin FV(\varphi[t/x], \Gamma_L) \cup \tilde{l}$. Then $\theta' := \theta[t/u]$ and $\mathcal{D}' := \mathcal{D}[t/u]^{\text{RHS}}$. To derive property I2 for this subcase, that is, $\models \neg \theta[t/u] \vee \varphi[t/x], \Gamma_L \vee \mathcal{D}[t/u]^{\text{RHS}}$, it is required that $u \notin FV(\mathcal{D}[t/u]^{\text{RHS}})$, which follows from the precondition $u \notin \tilde{l}$ of the subcase.

I FROM UNRESTRICTED PROOFS TO FOCUSED PROOFS

Recall that in Section H we gave a standard first-order proof system — see Figure 4. We then introduced the first-order version of focused proofs. A proof is called *focused* if no application of Ax , \top , \exists , REF , REPL contains in its conclusion a formula whose top-level connective is \vee , \wedge or \forall .

THEOREM 22. *Any proof tree (in the calculus presented in Fig. 4) can be turned into a focused proof tree, in exponential time.*

A counterexample to the focused property is a node in the proof tree where one of the rules Ax , \top , \exists , REF or REPL is applied and the conclusion contains a formula with \vee , \wedge or \forall as top-level operator. We eliminate the counterexample by converting the proof of the conclusion depending on the rule and the top-level operator. We iterate this until there is no counterexample. The steps involving connectives other than \wedge increase the tree size only linearly, while converting a \wedge step may double the size. The exponential time bound follows from this. In the following, we show the particular conversion for some chosen cases of counterexamples. For the remaining cases they are analogous. First, we consider the case where \wedge appears as top-level operator in the conclusion of an application of \exists . The proof of the conclusion then has the form shown on the left below and is converted to the form shown on the right.

where proof P' is like proof P except that all applications of \wedge where the principal formula is $\psi_1 \wedge \psi_2$ and corresponds (i.e., is passed down) to the shown occurrence of $\psi_1 \wedge \psi_2$ are removed and all occurrences of $\psi_1 \wedge \psi_2$ that correspond to the shown occurrence are replaced by ψ_1 . Proof P'' is defined like P' , except that the occurrences of $\psi_1 \wedge \psi_2$ that correspond to the shown occurrence are replaced by ψ_2 instead of ψ_1 . In the case where \forall appears as top-level operator in the conclusion of an application of \exists , the conversion is as follows.

where y is a fresh variable and proof P' is like proof P except that all applications of \forall where the principal formula is $\forall u \psi$ and corresponds to the shown occurrence of $\forall u \psi$ are removed and all occurrences of $\forall u \psi$ that correspond to the shown occurrence are replaced by $\psi[y/u]$. Finally, we show the case where \vee appears as top-level operator in the conclusion of an application of Ax .

One can observe that the translation step increases the size of a proof by a factor of at most 2. The exponential bound follows from this.