FRESH LOGIC: PROOF-THEORY AND SEMANTICS FOR FM AND NOMINAL TECHNIQUES

MURDOCH J. GABBAY

Abstract. In this paper we introduce **Fresh Logic**, a natural deduction style first-order logic extended with term-formers and quantifiers derived from the FM-sets model of names and binding in abstract syntax. Fresh Logic can be classical or intuitionistic depending on whether we include a law of excluded middle; we present a proof-normalisation procedure for the intuitionistic case and a semantics based on Kripke models in FM-sets for which it is sound and complete.

Contents

Date: October 9, 2005.

This work was inspired by conversations with Dale Miller and Alwen Tiu during a meeting on 11/10/02 at INRIA Rocquencourt. It was funded by INRIA FUTURS, Paris.

1. Introduction

In this paper we introduce **Fresh Logic**, a natural deduction style first-order logic extended with term-formers and quantifiers derived from the FM-sets model of names and binding in abstract syntax [9, 11, 12], developed by the author with A.M. Pitts. We show a proof-normalisation algorithm exists, and demonstrate soundness and completeness with respect to a model in FM-sets.

'FM' stands for 'Fraenkel-Mostowski', after two authors of papers on set theories very similar to what we call FM-sets (they were interested in proving independence of axioms of set theory [2], in particular the axiom of choice). Recently 'Nominal' has replaced 'FM' as an umbrella term for the logics and other systems derived from the original FM sets model, probably because it has fewer syllables and no umlauts.

Since this paper is a rock-hard proof-theoretic/semantic macho sets-fest, we keep the umlauts too.

FM-sets models names by a dedicated countably infinite set of names $a, b, c \in \mathbb{A}$, we give them a technical nomenclature **atoms**. This appears in Fresh Logic as a dedicated **type of atoms** $\mathbb{A}$ and a countably infinite set of atoms constant symbols $a, b, c, \ldots$ (they are not necessarily quite constant symbols, but we shall discuss that in due course).

FM-sets have a swapping action allowing us to rename (by swapping) atoms in sets. Fresh Logic has a corresponding term-former whose behaviour in the deduction system is to swap atoms constant symbols in terms. It is a fact that semantic swapping in FM-sets has excellent logical properties [12, 26]—Fresh Logic is designed to take advantage of this in its proof-theory.

Finally, FM-sets give rise to a derived quantifier $\mathsf{И}$ meaning 'for all but a finite set of names'. Fresh Logic adds a quantifier $\mathsf{И}$ to the usual $\forall$ and $\exists$ of first-order

logic. We shall see how swapping is vital to proof-normalisation for $\mathsf{И}$, lending proof-theoretic support to the FM-sets model.

Fresh Logic can be classical or intuitionistic depending on whether we include a law of excluded middle; we present a proof-normalisation procedure for the intuitionistic case and a semantics based on Kripke models in FM-sets for which it is sound and complete. The core of Fresh Logic is first-order logic, unchanged.

The style of FM techniques has been "to be within ϵ of standard practice". The most developed FM system is FreshML [6]. FreshML programs look just like the informal specifications we would normally write; issues of α-equivalence and renaming are smoothly delegated to the compiler. In the same spirit, we want

> a logic whose judgements look like normal First-Order Logic, whose language is augmented with the FM $\mathsf{И}$-quantifier, and with a good proof theory and semantics.

'Good' proof theory here means proof normalisation, Theorem 8.9, and also perhaps that the proof be simple and follow standard lines. Similarly for semantics. Soundness and completeness Theorem 7.7 and Theorem 9.12 follow standard lines, up to a point.

Outline of the paper.

- §2 We introduce a selection of relevant basic notions of FM-sets semantics (developed in detail in [12, 28]); semantic atoms $\mathbb{A}$, sets with a swapping action, support as a syntax-free notion of 'free names of', the finite support property, the $\mathsf{И}$ quantifier, the difference set of two permutations of $\mathbb{A}$, equivariant functions, and freshness $\#$.
- §3 We define the terms, predicates, and contexts of Fresh Logic.
- §4 Fresh Logic is intuitionistic. Its semantics consists of Kripke models of so-called 'possible worlds', which we call 'frames'; we define their structure.
- §5 We define judgements, and inductively define valid judgements. This includes intro- and elim-rules for the $\mathsf{И}$ quantifier.
- §6 We give some example derivations.
- §7 We develop a notion of validity for judgements with respect to a Kripke model semantics whose possible worlds are frames and whose accessibility is a notion of frame map.
- §8 We give a proof-normalisation algorithm for Fresh Logic.
- §9 We discuss how to build a standard model of a Fresh Logic theory out its syntax and use this model to prove completeness. We discover that completeness demands we add an extra axiom (Small), which we analyse.
- §10 We discuss alternatives to the many design decisions involved in making Fresh Logic.

Aside from a conclusions and related work section, there are also two appendices: Appendix A includes some definitions deferred from the text, Appendix B is a sequent-style presentation of Fresh Logic. We have not proved cut-elimination for the sequent presentation in full detail, but we have investigated it and expect it to work smoothly.

2. Semantics I

Definition 2.1 (The basics)**.** Fix a countably infinite **set of atoms** $a, b, c, \ldots \in \mathbb{A}$. For $a, b \in \mathbb{A}$ a **swapping** $(a\ b)$ is a function from $\mathbb{A}$ to $\mathbb{A}$ defined by

$$\begin{array}{lll} (b\ a) \cdot a & \stackrel{\text{def}}{=} & b \\ (b\ a) \cdot b & \stackrel{\text{def}}{=} & a \\ (b\ a) \cdot c & \stackrel{\text{def}}{=} & c \qquad c \not\equiv a, b. \end{array} \tag{1}$$

Write $P_{\mathbb{A}}$ for **the set of finite permutations of** $\mathbb{A}$; the group generated by the $(a\ b)$ with functional composition $\circ$ as the group action. Write **Id** for the identity function on $\mathbb{A}$, which is of course the unit of the group $(P_{\mathbb{A}}, \circ)$.

This concludes the basic notation. We now rehearse some basic FM theory:

Definition 2.2 (Nominal Set)**.** A **Nominal Set** is a pair $\langle X, \cdot \rangle$ of an underlying set X and **permutation action** $\cdot$ (written infix) of type $P_{\mathbb{A}} \times X \to X$. The permutation action satisfies

$$\forall \pi, \pi', x.\, \pi \cdot (\pi' \cdot x) = \pi \circ \pi' \cdot x \quad \text{and} \quad \mathbf{Id} \cdot x = x \tag{2}$$

(the standard rules for a permutation action) along with a **"smallness condition"**

$$\forall x \in X.\, \exists S \in \mathrm{Fin}\mathbb{A}.\, S \textit{ supports } x. \tag{3}$$

We may write just X for $\langle X, \cdot \rangle$.

Here we write $\mathrm{Fin}\mathbb{A}$ for the **set of finite subsets** $A, B, C, S \subseteq \mathbb{A}$. We define

$$S \textit{ supports } x \quad \text{means} \quad \forall a, b \notin S.\, (a\ b) \cdot x = x.$$

So (3) says **every** x **has a finite supporting set**, (3) is also called the **finite support property**. It can be proved [8, 12] that there is then a smallest supporting set. We call this the **support of** x and write it $S(x)$. As we would expect we can construct it as $\bigcap \{S \mid S \textit{ supports } x\}$.

- $\mathbb{A}$ with the natural functional action of permutations, is an FM set. $S(a) = \{a\}$.
- The set of terms t of the λ-calculus up to α-equivalence with variables modelled by atoms in the α-equivalence classes of abstract syntax, is an FM set: $S(t)$ coincides with $fv(t)$. An α-equivalence class of terms does not contain bound names, in the sense that the equivalence class quotients over them.
- The set of terms t of the λ-calculus *not* up to α-equivalence with variables modelled by atoms in the abstract syntax, is an FM set: $S(t)$ coincides with the names occurring free or bound in t.
- Any set of finitely branching trees with nodes labelled by elements of FM sets, is an FM set. The support is the union of the support of the labels, the swapping action acts pointwise on the labels. We shall see many examples of these in this paper: terms of Fresh Logic, formulae of Fresh Logic, derivations (proofs) of Fresh Logic, even models of Fresh Logic.
- The set of finite subsets of an FM set is an FM set. Swapping acts pointwise on the elements, the support of a finite set is the union of the support of its elements (proof omitted).
- $\mathrm{Fin}\mathbb{A}$ is an FM set with the pointwise set action. For $U \subseteq \mathbb{A}$ a finite set of atoms, $S(U)$ is equal to U.
- By calculation, for $U \subseteq \mathbb{A}$ as above, $S(\mathbb{A} \setminus U)$ is also equal to U.

We now introduce the **FM $\mathcal{N}$-quantifier**: write

$$\mathcal{N}a.\, \Phi(a) \quad \text{for} \quad \exists S \in \mathrm{Fin}\mathbb{A}.\, \forall a \notin S.\, \Phi(a). \tag{4}$$

Fresh Logic has a syntactic new-quantifier, also written $\mathsf{И}$. Its semantics will be this. (3) can be rephrased as $\mathsf{И}a, b.\,(a\ b)\cdot x = x$, by simple calculation using the fact that if S and S' are finite, so is $S \cup S'$.

We now write

$$a\#x \quad \text{for} \quad \mathsf{И}b.\,(b\ a)\cdot x = x \tag{5}$$

and read this 'a **is fresh for** x'. Fresh Logic also has a # connective, with semantics this one. This definition here is derived in (**D5**) and (**D6**) (for # in Fresh Logic) in Figure 4.

Another construction of $S(x)$ is $\{a \mid \neg a\#x\}$, and (3) may be rephrased as

$$\mathsf{И}a.\,a\#x. \tag{6}$$

In view of this, and of the examples above, we see that freshness gives a generalised notion of 'not in the free variable names of' and the smallness condition a generalised notion of 'everything mentions finitely many variables' — 'generalised', because it depends in no way on x being syntax and indeed will be most notably important when this is *not* the case, for example in the discussion of frames in §4.1.

Definition 2.3. A function $f : X \to Y$ is **equivariant** when

$$\forall x \in X, a, b \in \mathbb{A}.\,(a\ b)\cdot(f(x)) = f((a\ b)\cdot x). \tag{7}$$

Let NOM be the category with objects Nominal Sets and arrows equivariant functions between them.

(It is easy to verify that this is indeed a category.)

Any type T of binary labelled trees (say by $a \in \mathbb{A}$ and $n \in \mathbb{N}$) is a Nominal Set, with the action on the labels—trivially on $n \in \mathbb{N}$ and functionally on $a \in \mathbb{A}$. Then $\pi \cdot \langle t, t'\rangle = \langle \pi \cdot t, \pi \cdot t'\rangle$, so the tree-pairing constructor is equivariant. We see generalising this argument that constructors of all such types are equivariant.

It is a useful observation, which we shall use often, that if $a\#x$ and f is equivariant, then $a\#f(x)$.

One final useful concept is the **difference set** of $\pi, \pi' \in P_{\mathbb{A}}$.

$$ds(\pi, \pi') \stackrel{\text{def}}{=} \{n \mid \pi(n) \neq \pi'(n)\}. \tag{8}$$

For example, $ds((a\ b), \mathbf{Id}) = \{a, b\}$.

If S is a finite collection of atoms then write $S\#x$ for $a\#x$ for every $a \in S$. If S is empty, $S\#x$ is always true. We shall use the following lemma:

Lemma 2.4. *If $ds(\pi, \pi')\#x$ then $\pi \cdot x = \pi' \cdot x$.*

Proof. By calculation. □

It is interesting to observe this lemma in action for the example of terms up to α-equivalence above. If $t = \lambda f, x.fx$ then $(f\ x)\cdot[\lambda f, x.fx]_{=_\alpha}$ (is given by the action on representatives and) equals $[\lambda x, f.xf]_{=_\alpha}$, using an obvious notation for equivalence classes.

This concludes all we need of semantics. We use it heavily in the parts of this paper regarding semantics of Fresh Logic.

3. Basic definitions and notation

3.1. **Sorts.** We assume a set $\mathtt{X}, \mathtt{Y}, \ldots \in$ Sorts of **sorts**. We assume a distinguished **sort of atoms** $\mathtt{A} \in$ Sorts.

There may be sort-formers, e.g. pair-sorts $\mathtt{A} \times \mathtt{B}$ or function-sorts $\mathtt{X} \to \mathtt{X}$.

3.2. **Variables and atoms.** We assume countably infinite sets $a, b, c, \ldots \in \mathbb{A}$ of **atoms** and $x, y, z, \ldots \in \mathbb{V}$ of **variables** $\mathbb{V}$.

Since Fresh Logic has equality we may write identity on $\mathbb{A}$ and $\mathbb{V}$ as $\equiv$. We assume variables are inherently sorted, so $x \in \mathbb{V}$ has a unique sort $\mathtt{X} \in \text{Sorts}$. We may write $x : \mathtt{X}$, or we may leave sorts implicit. Atoms $a \in \mathbb{A}$ also have a sort, which is always $\mathtt{A}$.

3.3. **(Syntactic) permutations.** For $a, b \in \mathbb{A}$ a **(syntactic) swapping** $(a\ b)$ is a *pair* $\langle a, b\rangle$. Write $\pi, \pi', \kappa, \ldots \in P^{stx}_{\mathbb{A}}$ for the set of finite lists of swappings and call them **(syntactic) permutations**. Given $\pi, \pi' \in P^{stx}_{\mathbb{A}}$ write $\pi \circ \pi'$ for their list concatenation. Write **Id** for the empty list. Write $\mathrm{A}(\pi)$ for the atoms $a \in \mathbb{A}$ appearing in π.

Recall the set of semantic permutations in FM which we wrote $P_{\mathbb{A}}$. A syntactic permutation maps to a semantic permutation in the way obviously suggested by the notation. For $a \in \mathbb{A}$ and π a syntactic permutation, write $\pi(a)$ for the atom obtained via this semantic interpretation. Recall the notation $ds(\pi, \pi')$ from (8). We may write this also for syntactic permutations, using this semantic interpretation. Thus $ds((a\ b) \circ (a\ b), \mathbf{Id}) = \emptyset$, though $\mathrm{A}((a\ b) \circ (a\ b)) = \{a, b\}$.

3.4. **Moderated variables.** A **moderated variable** is a pair $\langle \pi, x\rangle \in P^{stx}_{\mathbb{A}} \times \mathbb{V}$, written $\pi \cdot x$.

Intuitively $\pi \cdot x$ is 'x with the permutation π applied to it' — but, since we do not yet know what x is, π remains 'in suspension' (*moderating* it) until we find out what x is.

A meaning to 'finding out what x is' is provided in the syntax by substitution (40).[1] Moderations lead to an interaction between substitution and moderations on variables, which is formally defined in (40), (42), and in the text below. We give one canonical example now:

$$((a\ b) \cdot x)\{x \mapsto \langle a, b, c, y\rangle\} \equiv \langle b, a, c, (a\ b) \cdot y\rangle.$$

Define $FV(\pi \cdot x) \stackrel{\text{def}}{=} \{x\}$ ('variables in') and $\mathrm{A}(\pi \cdot x) \stackrel{\text{def}}{=} \mathrm{A}(\pi)$ ('atoms in'). Call a moderated variable $\pi \cdot x$ **trivially moderated** when $\pi \equiv \mathbf{Id}$ and write x for $\mathbf{Id} \cdot x$. If $x : \mathtt{X}$ then $\pi \cdot x : \mathtt{X}$.

The notions of moderated variable and difference set are due to Urban [28].

3.5. **Terms. Terms** are defined by the following grammar:

$$s, t, \ldots ::= a \mid \pi \cdot x \mid \mathsf{c}(ts). \tag{9}$$

Here we introduce a shorthand we shall use again frequently, of writing ts for some list of terms. c has an arity and here (and in the future) we assume the terms have sorts appropriate to that arity in $\mathsf{c}(ts)$. We omit the typing rules for terms which are as usual.

We introduce notions $\mathrm{A}(t)$ of 'atoms of t' and $FV(t)$ of 'variables of t'. We gather these all similar definitions in an appendix; see (39) and (38). For example:

(10)
$$t \stackrel{\text{def}}{=} \mathsf{c}(a, b, x, (a\ b) \cdot x, y, (c\ d) \cdot y, z) \qquad \mathrm{A}(t) = \{a, b, c, d\} \qquad FV(t) = \{x, y, z\}.$$

We introduce a **substitution action** $t\{x \mapsto s\}$ defined in the standard way (s and x must have the same sort).

$$\begin{array}{llll} a\{x \mapsto s\} & = a & (\pi \cdot x)\{y \mapsto s\} & = \pi \cdot x \quad x \not\equiv y \\ (\pi \cdot x)\{x \mapsto s\} & = \pi \cdot_s s & \mathsf{c}(ts)\{x \mapsto s\} & = \mathsf{c}(ts\{a \mapsto s\}), \end{array}$$

[1] ... in the logic by rules involving substitution such as $\forall$-elimination rule, and in the model by the valuations.

See (40) in the appendix for a fully commented version.

The clause for moderated variables in the definition of substitution above uses a **permutation action** $\pi \cdot_s t$ (we mentioned this must happen in the last subsection). It is defined by:

$$\begin{array}{llll} \pi \cdot_s a & = \pi(a) & \pi \cdot_s (\kappa \cdot x) & = \pi \circ \kappa \cdot x \\ \pi \cdot_s (\kappa \cdot x) & = \kappa \cdot x & \pi \cdot_s \mathsf{c}(ts) & = \mathsf{c}(\pi \cdot_s ts). \end{array}$$

$\pi \cdot_s t$ is an operation on terms whose result is the term obtained by 'pushing π down to the moderated variables and atoms'. The fully commented definition is in (42). (The s is for 'sugar' since $(a\ b) \cdot_s t$ operates on t.) For example:

$$\begin{array}{c} (a\ b) \cdot_s x \equiv (a\ b) \cdot x \\ (a\ b) \cdot_s \langle a, b, c, y\rangle \equiv \langle b, a, c, (a\ b) \cdot y\rangle \\ t \stackrel{\text{def}}{=} \mathsf{c}(a, b, x, (a\ b) \cdot x, y, (c\ d) \cdot y, z) \\ (a\ b) \cdot_s t \equiv \mathsf{c}(b, a, (a\ b) \cdot x, (a\ b) \circ (a\ b) \cdot x, (a\ b) \cdot y, (a\ b) \circ (c\ d) \cdot y, (a\ b) \cdot z). \end{array}$$

Permutations act 'top-down' and substitutions act 'bottom-up'. Formally we have the following useful lemma:

Lemma 3.1. $\pi \cdot_s (t\{x{\mapsto}s\}) = (\pi \cdot_s t)\{x{\mapsto}s\}$ *always.*

Proof. By induction on terms. The crucial base case is $\pi \cdot_s (x\{x{\mapsto}t\}) = (\pi \cdot x)\{x{\mapsto}t\}$, see (40). □

3.6. **Propositions or formulae.** We assume **predicate constant symbols** $p, q, r \ldots \in \mathbb{P}$, each with an arity a list of sorts, for example $(\mathtt{X}, \mathtt{X})$, describing the number and sorts of its arguments (two arguments both of sort $\mathtt{X}$). For each $\mathtt{X} \in$ Sorts we assume distinguished constants $=\!:(\mathtt{X}, \mathtt{X})$ **equality** and $\#\!:(\mathtt{A}, \mathtt{X})$ **freshness**. Fresh Logic includes special deduction rules for them which we shall come to in due course.

Propositions or **formulae** are generated by the grammar

$$P ::= p(ts) \mid P \wedge P \mid P \vee P \mid P \Rightarrow P \mid \top \mid \bot \mid \forall x.\, P \mid \exists x.\, P \mid \text{И} n.\, P. \tag{11}$$

$P \Leftrightarrow Q$ is shorthand for $P \Rightarrow Q \wedge Q \Rightarrow P$ and $\neg P$ is shorthand for $P \Rightarrow \bot$.

Like terms, propositions have notions of free variables and atoms $FV(P)$ and $\mathrm{A}(P)$, formally defined in (38) and (39), substitution $P\{x{\mapsto}s\}$ defined in (41), and a permutation action defined in (43). In brief, $\forall$ binds x in $\forall x.\, P$, $\exists$ binds x in $\exists x.\, P$, and И binds n in $\text{И} n.\, P$, and we take formulae up to α-equivalence of bound variables $\forall/\exists x$ *and* bound atoms $\text{И} n$. Substitutions and permutations are capture-avoiding for these bindings, we include the core of the definition without comment:

$$\begin{array}{llll} \pi{\cdot_s}\, p(ts) & = p(\pi{\cdot_s}\, ts) & \pi{\cdot_s}\, (P \wedge Q) & = \pi{\cdot_s}\, P \wedge \pi{\cdot_s}\, Q \\ & \ldots & \pi{\cdot_s}\, \forall x.\, P & = \forall x.[\![\pi{\cdot_s}\, P\{x{\mapsto}\pi^{-1}{\cdot}x\}]\!]_x \\ \pi{\cdot_s}\, \exists x.\, P & = \exists x.[\![\pi{\cdot_s}\, P\{x{\mapsto}\pi^{-1}{\cdot}x\}]\!]_x & \pi{\cdot_s}\, \text{И} n.\, P & = \text{И} n.\, \pi{\cdot_s}\, P. \end{array}$$

$$\begin{array}{llll} p(ts)\{x{\mapsto}t\} & = p(ts\{x{\mapsto}t\}) & (P \wedge Q)\{x{\mapsto}t\} & = P\{x{\mapsto}t\} \wedge Q\{x{\mapsto}t\} \\ & \ldots & (\forall x'.\, P)\{x{\mapsto}t\} & = \forall x'.\, (P\{x{\mapsto}t\}) \\ (\exists x'.\, P)\{x{\mapsto}t\} & = \exists x'.\, (P\{x{\mapsto}t\}) & (\text{И} n.\, P)\{x{\mapsto}t\} & = \text{И} n.\, (P\{x{\mapsto}t\}). \end{array}$$

For examples, consider $P = \text{И} c.\, \forall x.\, p(c, (a\ b) \cdot x, (a\ c) \circ (a\ c) \cdot y)$. Then $FV(P) = \{y\}$, $\mathrm{A}(P) = \{a, b\}$, and $P\{x{\mapsto}t\} = P$.

Also $(d\ a) \cdot P = \text{И} c.\, \forall x.\, p(c, (d\ b) \cdot x, (d\ a) \circ (a\ c) \circ (a\ c) \cdot y)$. Note how the permutation acts differently on moderations of bound and free variables. This is necessary for §8.4 to work.

3.7. **Contexts.** A **(logical) context** is a multiset of propositions, usually written Γ.

Write $\mathrm{A}(\Gamma)$ and $FV(\Gamma)$ for the obvious extensions to many formulae. We may also write $\Gamma\{x{\mapsto}s\}$ and $\pi \cdot_s \Gamma$.

We shall give judgements and deduction rules in §5. First however we develop part of the semantics of Fresh Logic.

4. Semantics I: Frames

4.1. **Frames.** The semantics of Fresh Logic is a Kripke structure [29] of classical models. A classical model is an FM set (or rather a collection of FM sets, since Fresh Logic is typed), along with extra structure. We call this set with structure a **frame**, it is defined as follows:

Definition 4.1. A **frame** α for a Fresh Logic language $\mathcal{L}$ is assignements:

(i) $\mathtt{X} \mapsto [\![\mathtt{X}]\!] \in \mathbf{Obj}(NOM)$ of each sort to a Nominal Set ($\mathtt{X}$ a sort, not necessarily primitive). To $[\![\mathtt{X} \to \mathtt{Y}]\!]$ is associated a standard injection into $[\![\mathtt{Y}]\!]^{[\![\mathtt{X}]\!]}$. To $[\![\mathtt{A}]\!]$ is associated a standard bijection with $\mathbb{A}$.
(ii) $\mathtt{c} : \mathtt{X} \mapsto [\![\mathtt{c}]\!] \in [\![\mathtt{X}]\!]$ ($\mathtt{c}$ a constant symbol). $a : \mathtt{A} \mapsto a = [\![a]\!] \in \mathbb{A}$.
(iii) $p : \mathtt{X} \mapsto [\![p]\!] \subseteq [\![\mathtt{X}]\!]$ (p a predicate constant symbol).
(iv) $[\![\top]\!] = \{*\}$, and $[\![\bot]\!] = \emptyset$.
(v) A finite set of variable symbols $U_\alpha \subseteq \mathbb{V}$.
(vi) A **valuation** $x \in U_\alpha : \mathtt{X} \mapsto [\![x]\!]_\alpha \in [\![\mathtt{X}]\!]_\alpha$.

These assignements must satisfy:

(i) $[\![c]\!]$ must be equivariant: $(a\ b) \cdot ([\![c]\!](us)) = [\![c]\!]((a\ b) \cdot us)$ ($\mathtt{c}$ a constant symbol).
(ii) $[\![p]\!]$ must be equivariant: $u \in [\![p]\!] \implies (a\ b) \cdot u \in [\![p]\!]$.
(iii) Let $[\![\mathtt{f}]\!] : [\![\mathtt{X} \to \mathtt{Y}]\!]$ map to f under the standard injection. Then for any $\mathtt{t} : \mathtt{X}$, $f[\![t]\!] = [\![\mathtt{f}\,\mathtt{t}]\!]$.

A frame is an instance of an 'applicative structure' or 'prestructure' in the category of FM sets [21, 7, 27].

4.2. **Semantics of atoms and terms.** A frame α gives rise to a total map on t such that $FV(t) \subseteq U_\alpha$, defined inductively by

$$[\![\mathtt{c}(ts)]\!]_\alpha = [\![\mathtt{c}]\!]_\alpha([\![ts]\!]_\alpha).$$

It is easy to show by induction that:

$$(12)\quad \frac{[\![t]\!] = [\![t']\!]}{[\![s\{x{\mapsto}t\}]\!] = [\![s\{x{\mapsto}t'\}]\!]} \qquad \frac{[\![t]\!] = [\![t']\!]}{\langle [\![ts]\!]\{x{\mapsto}t\}\rangle \in [\![p]\!] \iff \langle [\![ts]\!]\{x{\mapsto}t'\}\rangle \in [\![p]\!]}$$

$$\frac{\bigwedge a \in ds(\pi,\pi').\ a\#u}{\langle [\![ts]\!]\{x{\mapsto}\pi \cdot_s t\}\rangle \in [\![p]\!] \iff \langle [\![ts]\!]\{x{\mapsto}\pi' \cdot_s t\}\rangle \in [\![p]\!]} \qquad \pi \cdot [\![t]\!] = [\![\pi \cdot_s t]\!]$$

We drop the α subscript in $[\![\cdot]\!]$ where convenient. We shall use the natural semantics of a syntactic permutation and write for example $\pi \cdot [\![t]\!]$ above instead of a stricter $[\![\pi]\!] \cdot [\![t]\!]$.

4.3. **Comments on the semantics.** The typical semantics for intuitionistic First-Order Logic is by collecting classical models together in a Kripke (or Beth) model [29]. The domains of these models may increase along accessibility so excluded middle is lost because 'new' elements can appear 'later'. Each classical model is called a 'world' and inclusions of models is called 'accessibility'.

We use this technique: we could extend the valuation on terms above to a valuation on predicates, and see that a frame extends to a classical semantics for Fresh Logic.

Frames are built in FM Sets and not ZF Sets. FM Sets interpret names and binding and are otherwise remarkably ZF-like. The match between transposition and freshness in the model and logic is invited by our notation. The FM Sets semantics though non-standard is very natural.

We have been particularly careful about valuations: a frame α comes packaged with an evaluation $x \mapsto [\![x]\!]_\alpha \in [\![\mathtt{X}]\!]_\alpha$ on a specific finite set $U_\alpha \subseteq \mathbb{V}$. This is useful for rigorously treating binding in $\forall$ and $\exists$, where we need to choose names for the bound variable symbols and would like to do so 'fresh'.

In intuitionistic logic terms can become equal moving along worlds $\alpha \leq \beta$; equality is treated in the model as a binary relation $[\![=]\!]_\alpha \subseteq [\![\mathtt{X}]\!]^2_\alpha$ which may grow, but the maps of underlying domains $[\![\mathtt{X}]\!]_\alpha \mapsto [\![\mathtt{X}]\!]_\beta$ are set inclusions. A completeness proof builds a model out of the syntax of a theory by $[\![\mathtt{X}]\!] = \{t : \mathtt{X}\}$ and interprets equality by a relation of provable equality.

A problem now arises. t and $(a\ b) \cdot_s t$ are not usually textually identical terms even if they are provably equal. Using the standard method outlined above, $[\![t : \mathtt{X}]\!]$ would not have finite support and $[\![\mathtt{X}]\!]$ would not be an FM set.

Therefore, our notion of model interprets equality and apartness as literal equality and literal apartness, and maps of underlying domains may be non-injective. Similar considerations explain why each frame comes equipped with a standard isomorphism $[\![\mathtt{A}]\!] \cong \mathbb{A}$.

We shall see that we still get sensible and reasonably familiar-looking soundness and completeness proofs.

5. Judgements

5.1. **Inductive definition.** A **judgement** is a pair $\langle \Gamma, P\rangle$ of a context and predicate written $\Gamma \vdash P$. Elements of Γ are called hypotheses and P the conclusion.

The **valid** or **derivable** judgements are inductively defined by the rules of Figure 1, Figure 2, and Figure 3. We discuss notation and semantics below.

5.2. **Notations used in the inductive definition.** In (New$\mathbb{A}$) $FV(\Gamma, C)$ denotes $\bigcup \{FV(P) \mid P \text{ in } \Gamma \text{ or } P \equiv C\}$; all variables occurring free in Γ or C. ts is a list of terms $t_1, t_2, \ldots$ (using an already-established convention) and $a\#ts$ is a list of assumptions $a\#t_1, a\#t_2, \ldots$.

In (πdiff) $ds(\pi, \pi')\#t$ denotes a list of predicates $a_1\#t, a_2\#t, \ldots$ for $ds(\pi, \pi') = \{a_1, a_2, \ldots\}$, and $\Gamma \vdash a\#ds(\pi, \pi')$ a list of judgements.

In ($\mathsf{И}$I) and ($\mathsf{И}$E) $P\{n{\mapsto}a\}$ denotes that formula obtained by (capture-avoiding) replacing every n in P by a.

In (Exhaust$\mathbb{A}$) the side conditions insist we have a proof of $\Gamma\{x{\mapsto}a\} \vdash P\{x{\mapsto}a\}$ for every $a \in \mathrm{A}(\Gamma, P)$, as well as at least one fresh atom n.

5.3. **Intuition behind the novel aspects of the deduction rules.** Fresh Logic contains intuitionistic first-order logic with equality, and we can see from the rules that this core is unchanged.

(#I) *and* (#E)*:* The semantics defined in §4 interprets syntactic equality judgements as literal equality, and freshness judgements as literal freshness. Atoms are their own semantics and freshness on atoms is inequality.

(πdiff)*:* We know from (12) that $[\![\pi \cdot_s t]\!] = \pi \cdot [\![t]\!]$. This rule formalises Lemma 2.4 inside Fresh Logic.

(Axiom) $\dfrac{}{\Gamma, P \vdash P}$

($\wedge$I) $\dfrac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}$

($\wedge$E1) $\dfrac{\Gamma \vdash P \wedge Q}{\Gamma \vdash P}$

($\wedge$E2) $\dfrac{\Gamma \vdash P \wedge Q}{\Gamma \vdash Q}$

($\vee$I1) $\dfrac{\Gamma \vdash P}{\Gamma \vdash P \vee Q}$

($\vee$I2) $\dfrac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q}$

($\vee$E) $\dfrac{\Gamma \vdash P \vee Q \quad \Gamma, P \vdash C \quad \Gamma, Q \vdash C}{\Gamma \vdash C}$

($\Rightarrow$I) $\dfrac{\Gamma, P \vdash Q}{\Gamma \vdash P \Rightarrow Q}$

($\Rightarrow$E) $\dfrac{\Gamma \vdash P \Rightarrow Q \quad \Gamma \vdash P}{\Gamma \vdash Q}$

($\bot$E) $\dfrac{\Gamma \vdash \bot}{\Gamma \vdash P}$

($\top$I) $\dfrac{}{\Gamma \vdash \top}$

($\forall$I) $\dfrac{\Gamma \vdash P}{\Gamma \vdash \forall x.\, P} \quad x \notin FV(\Gamma)$

($\forall$E) $\dfrac{\Gamma \vdash \forall x.\, P}{\Gamma \vdash P\{x{\mapsto}t\}} \quad x \notin FV(\Gamma)$

($\exists$I) $\dfrac{\Gamma \vdash P\{x{\mapsto}t\}}{\Gamma \vdash \exists x.\, P} \quad x \notin FV(\Gamma)$

($\exists$E) $\dfrac{\Gamma \vdash \exists x.\, P \quad \Gamma, P \vdash C}{\Gamma \vdash C} \quad x \notin FV(\Gamma, C)$

FIGURE 1. Core (standard) rules of Fresh Logic

(πI) $\dfrac{\Gamma \vdash P}{\Gamma \vdash \pi \cdot_s P}$

(πdiff) $\dfrac{\Gamma \vdash P\{x{\mapsto}\pi' \cdot_s t\} \quad \Gamma \vdash ds(\pi,\pi')\#t}{\Gamma \vdash P\{x{\mapsto}\pi \cdot_s t\}} \quad x \notin FV(t)$

($И$I) $\dfrac{\Gamma \vdash P\{n{\mapsto}a\} \quad \Gamma \vdash a\#t_i \ (i = 1,\ldots,k)}{\Gamma \vdash Иn.\, P} \quad \begin{array}{l} P/n = P' \bullet_{y_i} (t_i)_1^k \\ a \notin \mathrm{A}(P') \end{array}$

($И$E) $\dfrac{\Gamma \vdash Иn.\, P \quad \Gamma \vdash a\#t_i \ (i = 1,\ldots,k)}{\Gamma \vdash P\{n{\mapsto}a\}} \quad \begin{array}{l} P/n = P' \bullet_{y_i} (t_i)_1^k \\ a \notin \mathrm{A}(P') \end{array}$

FIGURE 2. π and $И$ rules of Fresh Logic

$$\text{(EqE)}\quad \frac{\Gamma \vdash P\{x{\mapsto}t'\} \quad \Gamma \vdash t = t'}{\Gamma \vdash P\{x{\mapsto}t\}} \qquad x \notin FV(t,t')$$

$$\text{(EqI)}\quad \frac{}{\Gamma \vdash t = t}$$

$$\text{(\#I)}\quad \frac{}{\Gamma \vdash a\#b}$$

$$\text{(\#c)}\quad \frac{\Gamma \vdash a\#ts}{\Gamma \vdash a\#\mathsf{c}(ts)}$$

$$\text{(\#E)}\quad \frac{\Gamma \vdash a\#a}{\Gamma \vdash C}$$

$$\text{(New}\mathbb{A}\text{)}\quad \frac{\Gamma, a\#ts \vdash C}{\Gamma \vdash C} \qquad a \notin \mathrm{A}(\Gamma, C)$$

$$\text{(Exhaust}\mathbb{A}\text{)}\quad \frac{\bigwedge n \in S.\, \Gamma\{x{\mapsto}n\} \vdash P\{x{\mapsto}n\}}{\Gamma\{x{\mapsto}t\} \vdash P\{x{\mapsto}t\}} \quad \mathrm{A}(\Gamma, P) \subsetneq S$$

FIGURE 3. $=$, $\#$, and $\mathbb{A}$ rules of Fresh Logic

(New$\mathbb{A}$)*:* Read bottom-up this rule says we can always introduce a fresh atom a. The idea of 'we can always find a fresh widget' turns up frequently ([14], [5], side-conditions on 'where x is fresh' in $\forall$-introduction rules, and a 'trick' by Krivine [16, Chapter 2] which is precisely a use of $\mathsf{И}$, are just some examples). A very important *difference* here is that (New$\mathbb{A}$) lets us choose a fresh for the current context without having to explicitly say what that context is, or even necessarily what freshness is with respect to that context; this turns out to be very useful.

(πI)*:* This is a version of (12) for predicates. We prove its validity in Lemma 7.5.

(Exhaust$\mathbb{A}$)*:* Here $\subsetneq$ just means $\subset$, but we reinforce the point that we mean here *proper* subset, so $S \setminus \mathrm{A}(\Gamma, C)$ is not empty. Since $\mathrm{A}(\Gamma, C)$ is finite, we can also restrict S to be finite (see §9).

This rule states that the atoms $a \in \mathbb{A}$ exhaust terms of sort $\mathtt{A}$ up to equality. To test $\Gamma \vdash P$ it suffices to test it on all the atoms mentioned in Γ or P, plus (at least) one fresh one acting as a fixed but arbitrary constant. See §5.5 for a further discussion.

The NEW intro- and elim-rules: ($\mathsf{И}$I) and ($\mathsf{И}$E) are symmetric and we consider just the first ($\mathsf{И}$I). A deduction rule should be closed under substitution $\{x{\mapsto}s\}$, to give us Lemma 8.3 and from it Theorem 8.2 the essential case of $\forall$.

So at first we take $P = P'$ and $t_i = y_i$ for all i. This rule becomes

$$\text{(13)}\quad \frac{\Gamma \vdash P\{n{\mapsto}a\} \quad \Gamma \vdash a\#y_i \ (i = 1, \ldots, k)}{\Gamma \vdash \mathsf{И}n.\, P} \qquad a \notin \mathrm{A}(P).$$

Thus "if $a\#y_i$ for all $y_i \in FV(P)$, and $a \notin \mathrm{A}(P)$, then a is fresh, and if $P(a)$ then $\mathsf{И}n.\, P(n)$". (Similarly the elim-rule says "if $a\#y_i$ for all $y_i \in FV(P)$, and $a \notin \mathrm{A}(P)$, then a is fresh, and if $\mathsf{И}n.\, P(n)$ then $P(a)$".) We discuss this some/any property from Prop 4.10 onwards in [12]. We now discuss this pair of rules in more detail:

5.4. **Slices.** We use **slices** to specify the deduction rules for the $\mathsf{И}$ quantifier ($\mathsf{И}$I) and ($\mathsf{И}$E).

Definition 5.1. A **slice** of a proposition P is a tuple $(P, k, (y_i)_{i=1}^k, P', (t_i)_{i=1}^k)$ of P, a number k, and:

(i) A choice of variable symbols $y_1, \ldots, y_k$, taken fresh (for the variables in P, and where convenient for any other variables we do not want accidental name-clashes with).
(ii) A proposition P' with $FV(P') = \{y_1, \ldots, y_k\}$.
(iii) A list of terms $t_1, \ldots, t_k$ such that $P'\{y_1 \mapsto t_1\} \ldots \{y_k \mapsto t_k\} \equiv P$.

A slice of P **covers** $n \in \mathbb{A}$ **in** P if $n \notin \mathrm{A}(t_i)$ for any $1 \leq i \leq k$.

It is useful to write

$$P' \bullet_{y_i} (t_i)_1^k \text{ or just } P' \bullet (t_i) \text{ for } P'\{y_1 \mapsto t_1\} \ldots \{y_k \mapsto t_k\}. \tag{14}$$

(A more suggestive notation for a slice is arguably $P \equiv (\lambda y_1, \ldots, y_k.P')t_1 \ldots t_k$; this treats slices as a very specific form of β-expansion. Then (ИI) states that if $a \notin \mathrm{A}(\lambda y_i.P')$ and $a\#t_i$ then a is apart from the application P and holds for any fresh a by FM swapping. We see this argument made formal in §8.4. We prefer the $\bullet$ notation partly to not suggest Fresh Logic is second-order, partly because $\bullet$ is typographically more convenient.)

Here are two examples of slices:

$$p(a, b, (a\ b) \circ (a\ b) \cdot x, y) \quad \text{slices as} \quad p(a, b, (a\ b) \cdot y_1, y_2) \bullet_{y_1, y_2} ((a\ b) \cdot x,\ y).$$
$$\forall x.\, p(a, b, (a\ b) \circ (a\ b) \cdot x, y) \quad \text{slices as} \quad \forall x.\, p(a, b, (a\ b) \cdot (a\ b) \cdot x, y_1) \bullet_{y_1} (y).$$

Slices induce a relation on propositions given by $P' \leq P$ if for suitable terms t_i, $P \equiv P' \bullet_{y_i} (t_i)$. This order is well-founded up to choices of the y_i which we shall as discussed assume are chosen fresh, and otherwise ignore.

Given two slices $P', P'' \leq P$ we can see by the structural properties of abstract syntax trees that there is a natural notion of their intersection (greatest lower bound) $P''' \leq P', P'' \leq P$.

For fixed P (and n) the set of slices of P (covering n) inherits this order from the P'. By the observations above and the fact that syntax is finite, this set also has a least slice with respect to $\leq$.

Write the least slice of P covering n as P/n. By abuse of notation write $P/n = P' \bullet_{y_i} (t_i)$ for "the least slice of P covering n is $P', (y_i), (t_i)$". This corresponds to the Caires-Cardelli notion of 'free term' [4, p.7], [5, p.5, §2].

(ИI) and (ИE) take least slices and this is important for §8.4.

(ИI) and (ИE) write '$P\{n \mapsto a\}$'. We easily see that all the substitutions are really happening in the P':

Lemma 5.2. *Suppose $P/n = P' \bullet (t_i)_1^k$. Then for any a,*

$$(P' \bullet (t_i))\{n \mapsto a\} = (P'\{n \mapsto a\}) \bullet (t_i).$$

While on the subject we also notice by inspection that in substitutions $P\{x \mapsto s\}$, if n is suitably fresh for s then the substitution there must really be happening in the terms t_i:

Lemma 5.3. *Suppose s is a term and x a variable and $n \notin \mathrm{A}(s)$. Suppose $P/n = P' \bullet (t_i)$. Then*

$$(P\{x \mapsto s\})/n = P' \bullet_{y_i} (t_i\{x \mapsto s\}).$$

5.5. **Significance of** (Exhaust$\mathbb{A}$)**.** (Exhaust$\mathbb{A}$) formalises the following feature of FM theory: to test a predicate on atoms, it suffices to test it on the (finite number of) atoms in its support, and on one fresh atom. This is implied by (3) the *smallness condition* on page 4, but does not imply it (see also §9.1).

Rather remarkably (Exhaust$\mathbb{A}$) seems to suffice to derive all of the practically useful consequences of (3). We now explore some derivable judgements which emerge from it:

Lemma 5.4. *Let p be a predicate symbol of kind* $\mathtt{A}$. $И n.\, p(n) \ \vdash\ \forall x.\, p(x)$ *is a theorem of Fresh Logic.*

The intuition why this should be so is that p has kind $\mathtt{A}$ and so uses no other free variables or atoms for which n must be fresh.

Proof. $P/n = p(n) \bullet_{\emptyset} ()$ so easily by ($И$E), for any a, $И n.\, p(n) \ \vdash\ p(a)$. Since this is for any a we use (Exhaust$\mathbb{A}$) to derive $И n.\, p(n) \ \vdash\ p(x)$ for a variable symbol x, and using ($\forall$I) $И n.\, p(n) \ \vdash\ \forall x.\, p(x)$ as required. □

In intuitionistic logic a **separation predicate** is one which states explicitly inequality of two elements. # is a separation predicate on $\mathtt{A}$ (this does not follow directly from (#E) and (#I) because they address only atoms, not variables).

Fresh Logic is intuitionistic. But equality on $\mathtt{A}$ is decidable:

Lemma 5.5. *For* $x, y : \mathtt{A}$*,*

$$x\#y \ \vdash\ x \neq y, \quad x \neq y \ \vdash\ x\#y, \quad \textit{and} \quad \vdash\ x = y \vee x \neq y$$

are derivable. As a corollary of the first two judgements, # is symmetric.

Proof. It is easy to derive $a\#b, a = b \ \vdash\ \bot$ and hence $a\#b \ \vdash\ a \neq b$, and also $a\#a \ \vdash\ a \neq a$. Using (Exhaust$\mathbb{A}$) twice we derive the first judgement.

It is easy to derive $a \neq b \ \vdash\ a\#b$ using (#I), and also $a \neq a \ \vdash\ a\#a$. As in the previous case we use (Exhaust$\mathbb{A}$) to derive the second judgement.

A proof of the third judgement can be constructed similarly. □

We can verify by calculation that the only equivariant $f : \mathbb{A} \to \mathbb{A}$ in NOM is the identity $\lambda a.a$. The logic reflects this:

Lemma 5.6. *For any constant symbol* $\mathtt{f} : \mathtt{A} \to \mathtt{A}$ *in a language $\mathcal{L}$ of Fresh Logic,* $\vdash\ x = \mathtt{f}(x)$ *is derivable.*

Proof. Using the previous lemma $a \neq \mathtt{f}(a) \ \vdash\ a\#\mathtt{f}(a)$. Using ($\#\mathtt{c}$) we can also deduce $a \neq \mathtt{f}(a) \ \vdash\ b\#\mathtt{f}(a)$. By (Exhaust$\mathbb{A}$) we conclude $a \neq \mathtt{f}(a) \ \vdash\ \mathtt{f}(a)\#\mathtt{f}(a)$, and again using the previous lemma we conclude $a \neq \mathtt{f}(a) \ \vdash\ \mathtt{f}(a) \neq \mathtt{f}(a)$, which is absurd.

□

6. Example deductions

(**D1**) connects $И$ and # within Fresh Logic. In the instance of ($И$I) $P/n = (n\#y) \bullet_y (x)$. (**D2**) derives (3) within Fresh Logic.

In (**D3**) $FV(A)$ and $FV(B)$ are the variables of A and B and $a\#FV(A, B)$ is a list of freshness assumptions $a\#x$ for each $x \in FV(A) \cup FV(B)$. In the last instance of ($И$I) the slice is $A \wedge B/n = (A \wedge B)\sigma \bullet (FV(A), FV(B))$, where σ renames $x \in FV(A, B)$ to y_i, as slices demand. We leave it to the reader to calculate the other slices and suchlike.

The reverse entailment $И n.\, (P \wedge Q) \ \vdash\ И n.\, P \wedge И n.\, Q$ is similar. We can also derive similar equivalences for the other connectives such as $\Rightarrow$ and $\vee$.

(**D4**) is one direction of a well-known commutativity property $\vdash\ (И a.\, \forall x.\, a\#x \Rightarrow P) \iff \forall x.\, И a.\, P$. In that proof, vs represents the set of all free variables of P except for x.

(**D5**) and (**D6**) are another well-known relation between #, $И$, and equality [12, Prop. 4.10]:

$$\vdash\ a\#x \iff И b.\, (b\ a) \cdot x = x.$$

(**D1**)
$$\dfrac{\dfrac{}{a\#x \ \vdash \ a\#x}\,(\text{Axiom})}{a\#x \ \vdash \ \text{И}n.\, n\#x}\,(\text{И I})}{\vdash \ \text{И}n.\, n\#x}\,(\text{New}\mathbb{A}).$$

(**D2**)
$$\dfrac{\dfrac{\dfrac{\dfrac{}{a, b\#x \ \vdash \ x = x}\,(\text{EqI})}{a, b\#x \ \vdash \ (a\ b)\cdot x = x}\,(\pi\text{diff})}{a, b\#x \ \vdash \ \text{И}m.\,(a\ m)\cdot x = x}\,(\text{И I})}{a, b\#x \ \vdash \ \text{И}n.\,\text{И}m.\,(n\ m)\cdot x = x}\,(\text{И I})}{\vdash \ \text{И}n.\,\text{И}m.\,(n\ m)\cdot x = x}\,(\text{New}\mathbb{A}), (\text{New}\mathbb{A}).$$

(**D5**)
$$\dfrac{\dfrac{\dfrac{}{a\#x, b\#x \ \vdash \ x = x}\,(\text{EqI})}{a\#x, b\#x \ \vdash \ (b\ a)\cdot x = x}\,(\pi\text{diff})}{a\#x \ \vdash \ \text{И}b.\,(b\ a)\cdot x = x}\,(\text{И I}), (\text{New}\mathbb{A})$$

(**D3**)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a\#FV(A), FV(B),\ \text{И}n.\, A\wedge\text{И}n.\, B \ \vdash \ \text{И}n.\, A\wedge\text{И}n.\, B}\,(\text{Axiom})}{a\#FV(A), FV(B),\ \text{И}n.\, A\wedge\text{И}n.\, B \ \vdash \ \text{И}n.\, B}\,(\wedge\text{E2})}{a\#FV(A), FV(B),\ \text{И}n.\, A\wedge\text{И}n.\, B \ \vdash \ B\{n\mapsto a\}}\,(\text{И E})}{a\#FV(A), FV(B),\ \text{И}n.\, A\wedge\text{И}n.\, B \ \vdash \ A\{n\mapsto a\}\wedge B\{n\mapsto a\}}\,(\wedge\text{I})}{a\#FV(A, B),\ \text{И}n.\, A\wedge\text{И}n.\, B \ \vdash \ A\{n\mapsto a\}\wedge B\{n\mapsto a\}}\,(\text{И I})}{a\#FV(A, B),\ \text{И}n.\, A\wedge\text{И}n.\, B \ \vdash \ \text{И}n.\,(A\wedge B)}\,(\text{New}\mathbb{A}).}{\text{И}n.\, A\wedge\text{И}n.\, B \ \vdash \ \text{И}n.\,(A\wedge B)}$$

(**D4**)
$$\dfrac{\dfrac{\dfrac{\dfrac{}{(\text{И}a.\,\forall x.\, a\#x\Rightarrow P),\ a\#x, vs \ \vdash \ a\#x\Rightarrow P}\,(\text{И E}), (\forall\text{E})}{(\text{И}a.\,\forall x.\, a\#x\Rightarrow P),\ a\#x, vs \ \vdash \ P}\,(\Rightarrow\text{E})}{\vdash \ (\text{И}a.\,\forall x.\, a\#x\Rightarrow P)\Rightarrow\forall x.\,\text{И}a.\, P}\,(\text{И I}), (\text{New}\mathbb{A}), (\forall\text{I}), (\Rightarrow\text{I})}$$

(**D6**)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{b\#x, \text{И}b.\,(b\ a)\cdot x = x \ \vdash \ \text{И}b.\,(b\ a)\cdot x = x}\,(\text{Axiom})}{b\#x, \text{И}b.\,(b\ a)\cdot x = x \ \vdash \ (b\ a)\cdot x = x}\,(\text{И E})}{b\#x, \text{И}b.\,(b\ a)\cdot x = x \ \vdash \ b\#(b\ a)\cdot x}\,(\pi\text{I})\ (\pi = (b\ a))}{b\#x, \text{И}b.\,(b\ a)\cdot x = x \ \vdash \ a\#(b\ a)\circ(b\ a)\cdot x}\,(\pi\text{diff})}{b\#x, \text{И}b.\,(b\ a)\cdot x = x \ \vdash \ a\#x}\,(\text{New}\mathbb{A})}{\text{И}b.\,(b\ a)\cdot x = x \ \vdash \ a\#x}$$
$$\dfrac{\dfrac{}{b\#x, \text{И}b.\,(b\ a)\cdot x = x \ \vdash \ b\#x}\,(\text{Axiom})}{b\#x, \text{И}b.\,(b\ a)\cdot x = x \ \vdash \ b\#(b\ a)\cdot x}\,(\text{EqE})$$

FIGURE 4. Example deductions

7. Validity and soundness

7.1. **Semantics II: Models and validity.** A **frame map** $\alpha \leq \beta$ is a collection of functions $\tau : [\![X]\!]_\alpha \to [\![X]\!]_\beta$ such that

(i) $U_\alpha \subseteq U_\beta$.
(ii) $\bigwedge x \in U_\alpha.\ \tau[\![x]\!]_\alpha = [\![x]\!]_\beta$.
(iii) $\bigwedge u \in [\![\mathtt{X}]\!]_\alpha.\ [\![\mathtt{c}]\!]_\beta(\tau u) = \tau[\![\mathtt{c}(u)]\!]_\alpha$.
(iv) $\bigwedge u \in [\![\mathtt{X}]\!]_\alpha.\ \tau u \in [\![p]\!]_\beta \iff u \in [\![p]\!]_\alpha$.

τ must commute with the isomorphisms $[\![\mathtt{A}]\!] \cong \mathbb{A}$ associated to the frame. It need not be injective.

Definition 7.1. A **model** $\mathcal{M}$ of Fresh Logic is a set of frames partially ordered by a set of frame maps, satisfying the following closure condition:

If $\alpha \in \mathcal{M}$, $x{:}\mathtt{X} \in \mathbb{V} \backslash U_\alpha$, and $u \in [\![\mathtt{X}]\!]_\alpha$, then the frame α' written $\alpha, x \mapsto u$ is in $\mathcal{M}$, where $U_{\alpha'} = U_\alpha \cup \{x\}$ and $[\![x]\!]_{\alpha'} = u$. Also, if $\alpha \leq_\tau \beta$ then $\alpha, x \mapsto u \leq_\tau \beta, x \mapsto \tau(u)$.

Definition 7.2. Define a relation $\alpha \Vdash P$ inductively on P such that $FV(P) \subseteq U_\alpha$ by

(i) $\alpha \Vdash p(t_1, \ldots, t_n)$ when $\langle\!|[\![t_1]\!]_\alpha, \ldots \rangle\!| \in [\![p]\!]_\alpha$.
(ii) Amongst the ps above are $=$ and $\#$. We insist additionally that $\alpha \Vdash t = t'$ when $[\![t]\!] = [\![t']\!]$ and $\alpha \Vdash t\#t'$ when $[\![t]\!]\#[\![t']\!]$.
(iii) $\alpha \Vdash P \vee Q$ when $\alpha \Vdash P$ or $\alpha \Vdash Q$. Similarly for $\wedge$. $\alpha \Vdash \top$ always.
(iv) $\alpha \Vdash P \Rightarrow Q$ when $\forall \beta \geq \alpha.$ if $\beta \Vdash P$ then $\beta \Vdash Q$.
(v) $\alpha \Vdash \forall x : \mathtt{X}.\, P$ when $\forall \beta \geq \alpha. \forall v \in [\![\mathtt{X}]\!]_\beta.\ \beta, x \mapsto v \Vdash P$. Here $x \notin U_\alpha$.
(vi) $\alpha \Vdash \exists x : \mathtt{X}.\, P$ when $\exists u \in [\![\mathtt{X}]\!].\ \alpha, x \mapsto u \Vdash P$. Here $x \notin U_\alpha$.
(vii) $\alpha \Vdash \mathit{N}n.\, P$ when $\alpha \Vdash P$. Here $n\#\alpha$ (we can α-convert to guarantee this because α satisfies (3)).

The condition that $FV(P) \subseteq U_\alpha$ is merely a well-formedness condition to ensure that, for example, $[\![t_1]\!]_\alpha$ is defined. Since $\Vdash$ is derived inductively and no rule derives $\alpha \Vdash \bot$, we know $\alpha \nVdash \bot$ for all α. Recalling also that $\neg P$ is sugar for $P \Rightarrow \bot$, we obtain the following:

$$\alpha \Vdash \neg P \text{ when } \forall \beta \geq \alpha.\text{ if } \beta \not\Vdash P.$$

The condition $x \notin U_\alpha$ formally states that x is chosen fresh. Similarly for the condition $n\#\alpha$. Calculated explicitly for the structure of frames α this condition means that $n\#[\![x]\!]_\alpha$ for every $x \in U_\alpha$.

Note that the semantics of equality is literal identity. We see a frame map need not be an inclusion on underlying sets. In Kripke models elements can 'become' equal moving along the accessibility relation, just as any other predicate may 'become' true.

It is standard in some communities that accessibility be an inclusion on underlying sets ([1, §6 p416 (3)] and [29, p249 3.3(i)]) and equality is not identity but a relation satisfying very special coherence conditions so that the semantics of term-formers and predicates cannot distinguish related elements. Other communities ([7, 27], [22, p6]) allow accessibility to be a non-injective function, so that equality can be modelled by literal equality in the model.

Thus Fresh Logic semantics follow the latter tradition (see for example the second clause in Definition 7.2 above, for equality). A semantics in the former tradition is possible but the completeness proof breaks for it, see §9.3.

7.2. **Basic properties of validity.** We may write $\alpha \Vdash \Gamma$ for $\bigwedge P \in \Gamma.\ \alpha \Vdash P$.

Lemma 7.3. (i) *Cofinitely many b satisfy $\alpha \Vdash b\#t$.*
(ii) *If $\alpha \Vdash P$ and $\beta \geq \alpha$ then $\beta \Vdash P$ (call this property **persistence**).*
(iii) *If $\alpha \Vdash P$ and $x : \mathtt{X} \notin U_\alpha$ then $\alpha, x \mapsto u \Vdash P$ for any $u \in [\![\mathtt{X}]\!]_\alpha$.*

(iv) $\alpha, x \mapsto u \Vdash P$ *if and only if* $\alpha, x \mapsto \pi \cdot u \Vdash P\{x \mapsto \pi^{-1} \cdot x\}$.
(v) *Suppose* $x \notin U_\alpha$ *and* $FV(P) \subseteq U_\alpha \cup \{x\}$. *Then* $\alpha, x \mapsto [\![t]\!]_\alpha \Vdash P$ *if and only if* $\alpha \Vdash P\{x \mapsto t\}$.
(vi) *Suppose* $x \notin U_\alpha$ *and* $FV(P) \subseteq U_\alpha$. *Then if* $\alpha, x \mapsto u \Vdash P$ *then* $\alpha \Vdash P$.

Proof. (i) $[\![t]\!]$ is an element of a Nominal Set and so has finite support, so there are cofinitely many b such that $b\#[\![t]\!]_{\alpha,\epsilon}$.
(ii) By induction on P. We only really need to consider the base cases $p(t_1, \ldots, t_k)$, $t = t'$, and $t\#t'$, since for the other cases this property is built-in or trivial. We just consider $p(t_1, \ldots, t_k)$. Suppose $\langle [\![t_1]\!]_{\alpha,\epsilon}, \ldots \rangle \in [\![p]\!]_\alpha$. Associated to $\beta \geq \alpha$ is a frame map so $[\![p]\!]_\alpha \subseteq [\![p]\!]_\beta$ and $[\![t_i]\!]_{\alpha,\epsilon} = [\![t_i]\!]_{\beta,\epsilon}$, and evidently $\langle [\![t_1]\!]_{\beta,\epsilon}, \ldots \rangle \in [\![p]\!]_\beta$.
(iii) By very easy induction on P, omitted.
(iv) By very easy induction on P, omitted.
(v) By induction on P. The previous two parts are not corollaries of this one, because we cannot assume $u = [\![t]\!]_\alpha$ for some t.
(vi) By very easy induction on P, omitted.

□

7.3. **Validity for some structural rules.** We prove results which are not quite but but amount to soundness (EqI), (#I), (#E), (#c), (EqE), and (πdiff). This is very simple.

Lemma 7.4. *The following deduction rules are (trivially) valid:*

$$\alpha \Vdash t = t \qquad \alpha \Vdash a\#b \qquad \frac{\alpha \Vdash a\#a}{\alpha \Vdash \bot} \qquad \frac{\alpha \Vdash a\#ts}{\alpha \Vdash a\#\mathsf{c}(ts)}$$

$$\frac{\alpha \Vdash P\{x \mapsto t'\},\ t = t'}{\alpha \Vdash P\{x \mapsto t\}} \qquad \frac{\alpha \Vdash P\{x \mapsto t'\},\ t' = t}{\alpha \Vdash P\{x \mapsto t\}} \tag{15}$$

$$\frac{\alpha \Vdash P\{x \mapsto \pi \cdot_s t\},\ ds(\pi, \pi')\#t}{\alpha \Vdash P\{x \mapsto \pi' \cdot_s t\}}.$$

Proof. Equality is interpreted by literal identity and $[\![t]\!]_\alpha = [\![t]\!]_\alpha$ always. Freshness is modelled by literal freshness and on atoms this coincides with inequality, so $a\#b$ always (when $a \not\equiv b$) and $a\#a$ never. By definition $[\![\mathsf{c}(ts)]\!] = [\![\mathsf{c}]\!]([\![ts]\!])$ and $[\![\mathsf{c}]\!]$ is equivariant. The equality elimination rules are easy, given that equality is interpreted as literal equality. (πdiff) is easy from Lemma 2.4. □

We now prove soundness for (πI):

Lemma 7.5. *The following deduction rule is valid:*

$$\frac{\alpha \Vdash P}{\alpha \Vdash \pi \cdot_s P} \tag{16}$$

Proof. By induction on P.
(i) $P = p(t_1, \ldots, t_k)$: By (12) $\pi \cdot \langle [\![t_1]\!], \ldots, [\![t_k]\!] \rangle = \langle [\![\pi \cdot_s t_1]\!], \ldots, [\![\pi \cdot_s t_k]\!] \rangle$, and by construction $[\![p]\!]$ is equivariant.
(ii) $P = P_1 \Rightarrow P_2$: Suppose $\alpha \Vdash P_1 \Rightarrow P_2$. $\pi \cdot_s P = \pi \cdot_s P_1 \Rightarrow \pi \cdot_s P_2$, so suppose $\beta \Vdash \pi \cdot_s P_1$. By induction hypothesis $\beta \Vdash \pi^{-1} \cdot_s \pi \cdot_s P_1$. We know $\Vdash$ is closed under (πdiff) by the previous result so $\beta \Vdash P_1$, whence $\beta \Vdash P_2$. By induction hypothesis $\beta \Vdash \pi \cdot_s P_2$. This suffices to establish $\alpha \Vdash \pi \cdot_s P$.
(iii) $P = \forall x : \mathtt{X}.\, P'$: Take some $\beta \geq \alpha$ and $u \in [\![\mathtt{X}]\!]_\beta$; then $\alpha, x \mapsto \pi^{-1} \cdot u \Vdash P'$. By hypothesis

$$\alpha, x \mapsto \pi^{-1} \cdot u \Vdash \pi \cdot_s P' \quad \text{so} \quad \alpha, x \mapsto u \Vdash \pi \cdot_s P'\{x \mapsto \pi^{-1} \cdot x\}.$$

Since u and β were arbitrary, $\alpha \Vdash \pi \cdot_s \forall x.\, P$.

(iv) $P = \exists x : \mathtt{X}.\, P'$: There is some $u \in [\![\mathtt{X}]\!]_\alpha$ such that $\alpha, x \mapsto u \Vdash P'$. By hypothesis

$$\alpha, x \mapsto u \Vdash \pi \cdot_s P' \quad \text{so} \quad \alpha, x \mapsto \pi \cdot u \Vdash \pi \cdot_s P'\{x \mapsto \pi^{-1} \cdot x\}$$

This suffices to conclude $\alpha \Vdash \pi \cdot_s \exists x.\, P$.

(v) $P = \mathit{N}n.\, P'$: n is fresh for π so $\pi \cdot_s \mathit{N}n.\, P' = \mathit{N}n.\, \pi \cdot_s P'$. By induction hypothesis $\alpha \Vdash \pi \cdot_s P'$. This suffices to conclude $\alpha \Vdash \pi \cdot_s \mathit{N}n.\, P'$.

We omit the remaining cases.

□

7.4. **Meta-level swapping.** Our treatment of Fresh Logic uses FM sets for models $\mathcal{M}$, frames α, syntax, and so on. We therefore enjoy a 'meta-level' swapping on all of these. We wrote it $\cdot$. This is distinct from the swapping action $\cdot_s$ defined in (42) and (43) which acts on terms and formulae.

From equivariance of FM sets [12, Lemma 4.7] meta-level swapping commutes with constructors and term-formers; because, in the notation of that Lemma, they can be specified by some ϕ in the language of FM logic. Thus the property of being a frame Definition 4.1 can be encoded as some ϕ for a frame α, the set $(a\ b) \cdot \alpha$ is also a frame, identical to α except that $x \in U_{(a\ b)\cdot\alpha} = U_\alpha$ maps to $(a\ b) \cdot [\![x]\!]_\alpha$.

For a formula P, the set $(a\ b) \cdot P$ is also a formula, obtained by syntactically swapping a and b.

Relations $\Vdash$ and $\vdash$ are equivariant with respect to $\cdot$, so

$$\Gamma \vdash P \Leftrightarrow (a\ b)\cdot\Gamma \vdash (a\ b)\cdot P \quad \text{and} \quad \alpha \Vdash P \Leftrightarrow (a\ b)\cdot\alpha \vdash (a\ b)\cdot P.$$

Very useful corollaries, which we shall use in the next subsection, are as follows:

$$(a\ b)\cdot\Gamma \vdash P \Leftrightarrow \Gamma \vdash (a\ b)\cdot P \quad \text{and} \quad (a\ b)\cdot\alpha \Vdash P \Leftrightarrow \alpha \vdash (a\ b)\cdot P.$$

7.5. **Entailment relative to a model and soundness.**

Notation 7.6. Let $\mathcal{M}$ be a model of Fresh Logic. Write $\Gamma \Vdash_{\mathcal{M}} P$ when $\forall \alpha \in \mathcal{M}.\, \alpha \Vdash \Gamma \Rightarrow \alpha \Vdash P$.

In the proof of soundness below we shall work relative to a fixed but arbitrary $\mathcal{M}$ and abbreviate $\Gamma \Vdash_{\mathcal{M}} P$ to $\Gamma \Vdash P$. Later on in the proof of completeness we shall write $\Gamma \Vdash P$ with a different meaning, that $\forall \mathcal{M}.\, \Gamma \Vdash_{\mathcal{M}} P$. We will always make clear which of the two notations we mean. In this subsection, we mean the former.

Theorem 7.7 (Soundness)**.** *Fix some model $\mathcal{M}$. If $\Gamma \vdash P$ then $\Gamma \Vdash_{\mathcal{M}} P$.*

Proof. By induction on the proof of $\Gamma \vdash P$.

(i) The rule ($\Rightarrow$I): Suppose $\Gamma, P \Vdash Q$ and $\alpha \Vdash \Gamma$. For any $\beta \geq \alpha$, if $\beta \Vdash P$ then using persistence $\beta \Vdash \Gamma, P$ so $\beta \Vdash Q$.

(ii) The rule ($\forall$I): Suppose $\Gamma \Vdash P$ and $x : \mathtt{X} \notin FV(\Gamma)$. Suppose $\alpha \Vdash \Gamma$. For any $\beta \geq \alpha$ by persistence $\beta, x \mapsto u \Vdash \Gamma$ for any $u \in [\![\mathtt{X}]\!]_\beta$. By assumption $\beta, x \mapsto u \Vdash P$, and this suffices to prove $\alpha \Vdash \forall x.\, P$.

(iii) The rule ($\forall$E): Suppose $\Gamma \Vdash \forall x.\, P$ and choose $x : \mathtt{X} \notin FV(\Gamma)$. Suppose $\alpha \Vdash \Gamma, \forall x.\, P$. Then for any t, $\alpha, x \mapsto [\![t]\!]_\alpha \Vdash P$ and by Lemma 7.3 $\alpha \Vdash P\{x \mapsto t\}$.

(iv) The rule ($\exists$I): Suppose $\Gamma \Vdash P\{x \mapsto t\}$. Suppose $\alpha \Vdash \Gamma$. Then $\alpha, x \mapsto [\![t]\!]_\alpha \Vdash P$, so $\alpha, x \mapsto [\![t]\!]_\alpha \Vdash \exists x.\, P$, so $\alpha \Vdash \exists x.\, P$.

(v) The rule ($\exists$E): Suppose $\Gamma \Vdash \exists x : \mathtt{X}.\, P$ and $\Gamma, P \Vdash C$. Now suppose $\alpha \Vdash \Gamma$, so $\alpha \Vdash \exists x.\, P$, so for some $u \in [\![\mathtt{X}]\!]$, $\alpha, x \mapsto u \Vdash P$. Also using Lemma 7.3 $\alpha, x \mapsto u \Vdash \Gamma$, so $\alpha, x \mapsto u \Vdash C$ and finally $\alpha \Vdash C$.

(vi) (EqI), (#I), (#E), (#$\mathtt{c}$), and (EqE), follow by Lemma 7.4.

(vii) The rule ($\mathit{И}$I): Suppose $P/n = P' \bullet (t_i)$ and $\Gamma \Vdash P\{n{\mapsto}a\}$, $a\#t_i$. Suppose $\alpha \Vdash \Gamma$ so $\alpha \Vdash P\{n{\mapsto}a\}$, $a\#t_i$. n is fresh so by Lemma 7.3 $\alpha \Vdash n\#t_i$ and $n \notin \mathrm{A}(t_i)$ for all i.

By clauses of the induction hypothesis and Lemma 7.5 $\alpha \Vdash (n\ a) \cdot_s P\{n{\mapsto}a\}$ so $\alpha \Vdash P' \bullet ((n\ a) \cdot_s t_i)$, so $\alpha \Vdash P' \bullet (t_i) \equiv P$. We deduce $\alpha \Vdash \mathit{И}n.\, P$.

(viii) The rule (Exhaust$\mathbb{A}$): Suppose $\Gamma\{x{\mapsto}a\} \Vdash P\{x{\mapsto}a\}$ for each $a \in \mathrm{A}(\Gamma, P)$, and for one atom n not in $\mathrm{A}(\Gamma, P)$. Suppose $\alpha \Vdash \Gamma$.

If $[\![x]\!]_\alpha = a \in \mathrm{A}(\Gamma, P)$ then we use Lemma 7.3 to conclude $\alpha \Vdash \Gamma\{x{\mapsto}a\}$ so $\alpha \Vdash P\{x{\mapsto}a\}$ so $\alpha \Vdash P$.

Suppose $[\![x]\!]_\alpha = n' \notin \mathrm{A}(\Gamma, P)$. Using notation and results from §7.4 $[\![x]\!]_{(n\ n')\cdot\alpha} = n$ and $(n\ n') \cdot \alpha \Vdash (n\ n') \cdot \Gamma \equiv \Gamma$. We conclude $(n\ n') \cdot \alpha \Vdash \Gamma\{x{\mapsto}n\}$ so $(n\ n') \cdot \alpha \Vdash P\{x{\mapsto}n\}$ so $(n\ n') \cdot \alpha \Vdash P$ and $\alpha \Vdash (n\ n') \cdot P \equiv P$.

Since we have $\alpha \Vdash P$ for all atoms, and thus all possible values of $[\![x]\!]_\alpha$, we are done.

□

8. Proof normalisation

8.1. **Overview of the proof.** The proof-normalisation proof follows standard lines. We establish terminology and sketch it.

As a matter of notation we let Π and Π' vary over proofs (for example, Π might be[2]

$$\dfrac{\dfrac{}{A \wedge B \vdash A \wedge B}\,(\text{Axiom})}{A \wedge B \vdash A}\,(\wedge\text{E1})\quad .$$

We shall most usually only care about the final rule or few rules of a proof, in which case we write, for example $\dfrac{\Pi'}{A \wedge B \vdash A}\,(\wedge\text{E1})$. Later on, for brevity and fitting proofs neatly into a line of text, we may use notation which would represent the proof just given as $\Pi', (\wedge\text{E1})$ or $(\text{Axiom}), (\wedge\text{E1})$.

In a proof Π a **critical pair** is a pair of intro-rule followed after n **intervening rules** by an elim-rule for the *same* connective in the formulae. It is **unseparated** when $n = 0$ (the elim-rule follows immediately) and **separated** otherwise. This critical pair is separated by a single intervening rule ($\vee$E) (so $n = 1$):

$$(17)\quad \dfrac{\Gamma \vdash P \vee Q \qquad \dfrac{\Gamma, P \vdash A \quad \Gamma, P \vdash B}{\Gamma, P \vdash A \wedge B}\,(\wedge\text{I}) \qquad \dfrac{\Gamma, Q \vdash A \quad \Gamma, Q \vdash B}{\Gamma, Q \vdash A \wedge B}\,(\wedge\text{I})}{\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\,(\wedge\text{E1})}\,(\vee\text{E})$$

This critical pair is unseparated:

$$(18)\quad \dfrac{\dfrac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,(\wedge\text{I})}{\Gamma \vdash A}\,(\wedge\text{E1})$$

Rules that can occur between critical pairs are those that do not change the top-level connective of their main formula. They are:

$$(19)\quad (\vee\text{E})\ (C),\ (\exists\text{E})\ (C),\ (\pi\text{I})\ (P, \pi \cdot_s P),\ (\pi\text{diff})\ (P\{x{\mapsto}\pi \cdot t\}, P\{x{\mapsto}\pi' \cdot t\}),\ (\text{EqE}),\ (\text{New}\mathbb{A}),\ \text{and}\ (\text{Exhaust}\mathbb{A})\ .$$

[2]Nobody said it had to be *interesting*.

Proof-normalisation has two parts: **commutation rules** commute members of (19) with elimination rules (20), drawing them together. **Essential cases** then eliminate them entirely. A proof without critical pairs is **normalised**.

The possible elimination rules are:

(20) $(\wedge\mathrm{E1})$, $(\wedge\mathrm{E2})$, $(\vee\mathrm{E})$, $(\forall\mathrm{E})$, $(\exists\mathrm{E})$, $(\bot\mathrm{E})$, $(\text{И}\mathrm{E})$, and (EqE).

We do not consider all the cases!

In a proof, let a **closest critical pair** be such that the number of intervening rules separating the relevant intro- and elim-rules is no greater than that of any other critical pair.

Definition 8.1. Let the **proof-size** of a Fresh Logic derivation (we write them $\Pi, \Pi', \ldots$, we may also call them **proofs**) be the pair $(C, S) : \mathbb{N} \times \mathbb{N}$. C is the number of rules intervening between a closest critical pair, S is the depth of the derivation as a tree. We order proof-size lexicographically left to right.

Say Π is **smaller/larger** than Π' when it has lesser/greater size.

Clearly proof-size is well-founded. This section is now a long list of proof-transformations which do not increase, or strictly decrease, proof-size. Applied repeatedly in a natural way (outlined in Theorem 8.9) to a valid derivation, the procedure terminates with a normal form.

8.2. **Essential case of $\forall$.**

Theorem 8.2. *A derivation Π concluding with $(\forall\mathrm{I})$,$(\forall\mathrm{E})$ may be transformed to a smaller derivation of the same conclusion.*

Proof. Suppose we have a proof

$$\Pi = \dfrac{\dfrac{\dfrac{\Pi'}{\Gamma \vdash P}\, x \notin FV(\Gamma)}{\Gamma \vdash \forall x.\, P}\,(\forall\mathrm{I})}{\Gamma \vdash P\{x{\mapsto}t\}}\,(\forall\mathrm{E})$$

We use Lemma 8.3 to reduce it to Π'' proving $\Gamma\{x{\mapsto}t\} \equiv \Gamma \vdash P\{x{\mapsto}t\}$. □

Lemma 8.3 (Substitutivity)**.** *For s a term and $z \in \mathbb{V}$, if Π proves $\Gamma \vdash P$ then there is a no longer derivation Π' of $\Gamma\{z{\mapsto}s\} \vdash P\{z{\mapsto}s\}$.*

So Fresh Logic enjoys the admissible rule

$$\dfrac{\Gamma \vdash C}{\Gamma\{z{\mapsto}s\} \vdash C\{z{\mapsto}s\}} \tag{21}$$

and it does not increase proof-size.

Proof. By induction on proof-size. If Π is a proof, write the transformed proof $\Pi\{z{\mapsto}s\}$.

The case $(\forall\mathrm{E})$**.** Suppose $\Pi, (\forall\mathrm{E})\ (t/x)$ is a proof, where $x \notin FV(\Gamma, C)$. We α-convert so also $x \neq z$ and $x \notin FV(s)$. $P\{x{\mapsto}t\}\{z{\mapsto}s\} = P\{z{\mapsto}s\}\{x{\mapsto}t\{z{\mapsto}s\}\}$ and we see that $\Pi\{z{\mapsto}s\}, (\forall\mathrm{E})\ (t\{z{\mapsto}s\}/x)$ is the transformed proof.

The case $(\text{И}\mathrm{I})$**.** α-convert so $n \notin \mathrm{A}(s)$. It suffices to verify that the side-conditions of this rule are preserved under substituting s for z. And indeed they are.

The case (Exhaust$\mathbb{A}$)**.** Suppose $x \notin FV(s)$ and we have proofs Π_a of $\Gamma\{x{\mapsto}a\} \vdash P\{x{\mapsto}a\}$ for all $a \in \mathrm{A}(\Gamma, P)$ and for (at least) one other fresh n. By applying the transposition $(b\ n)$ to each of these for $b \in \mathrm{A}(s)$ not amongst the a, we obtain by FM equivariance applied to valid proofs, other valid proofs Π_b. These proofs also inherit the induction hypothesis. Therefore we have proofs $\Pi_b\{z{\mapsto}s\}$ for $b \in \mathrm{A}(\Gamma, P, s)$,

which includes $\mathrm{A}(\Gamma\{z{\mapsto}s\}, P\{z{\mapsto}s\})$, and also fresh n. Then $\bigwedge b.\Pi_b, (\mathrm{Exhaust}_{\mathbb{A}})$ is the transformed proof. □

8.3. **Essential case of π and discussion of** (πI) **and** (πdiff)**.** (πI) has no associated elimination rule and therefore no associated essential case. Recall from the formal definition (43) that the action $\pi \cdot_s P$ in (πI) is in any case a meta-level operation on syntax which does not change the top-level connective of P.

Clearly two consecutive uses of (πI) with π and π', say, can be concatenated to one of $\pi' \circ \pi$, and a slightly tidier notion of proof-normalisation can be obtained if we allow this and similar simplifications.

To make proof-normalisation work at all, we need the following result:

However (πI) and (πdiff) are invertible in suitable senses made formal in the following two lemmas:

Lemma 8.4. *Fresh Logic enjoys the admissible rule*

$$\frac{\Gamma, P\{x{\mapsto}\pi' \cdot_s t\} \vdash C \quad \Gamma, P\{x{\mapsto}\pi' \cdot_s t\} \vdash ds(\pi,\pi')\#t}{\Gamma, P\{x{\mapsto}\pi \cdot_s t\} \vdash C}, \tag{22}$$

proof-size does not increase.

Proof. By (πdiff). □

Lemma 8.5. *Fresh Logic enjoys the 'elimination' and 'left-intro' admissible rules*

$$\frac{\Gamma \vdash \pi \cdot_s P}{\Gamma \vdash P}, \tag{23}$$

$$\frac{\Gamma, P \vdash C}{\Gamma, \pi \cdot_s P \vdash C}, \tag{24}$$

proof-size does not increase.

Proof. The first part from (πI) for π^{-1} then (πdiff) applied individually to every moderated variable in P, observing that $ds(\pi^{-1} \circ \pi, \mathbf{Id}) = \emptyset$. The second part from (22) applied individually to every moderated variable in P using the same observation. □

8.4. **Essential case of И.** Consider a proof of the form

$$\dfrac{\dfrac{\dfrac{\Pi'}{\Gamma \vdash (P' \bullet_{y_i} (t_i)_{i=1}^k)\{n{\mapsto}a\} \quad \Gamma \vdash a\#t_i}}{\Gamma \vdash \text{И}n.\, P' \bullet (t_i)}(\text{И I}) \qquad \Gamma \vdash b\#t_i}{\Gamma \vdash (P' \bullet (t_i))\{n{\mapsto}b\}}(\text{И E}). \tag{25}$$

We know (И I) and (И E) use the same slice here, because it is the least one covering n. Side-conditions dictate that $a, b \notin \mathrm{A}(P')$. Transform this to

$$\dfrac{\dfrac{\dfrac{\Pi'}{\Gamma \vdash (P' \bullet_{y_i} (t_i)_{i=1}^k)\{n{\mapsto}a\} \equiv (P'\{n{\mapsto}a\}) \bullet (t_i)}}{\Gamma \vdash ((a\ b) \cdot_s (P'\{n{\mapsto}a\})) \bullet (t_i)}(\pi\text{I}) \qquad \Gamma \vdash a, b\#t_i}{\Gamma \vdash (P'\{n{\mapsto}b\}) \bullet (t_i) \equiv (P' \bullet (t_i))\{n{\mapsto}b\}}(\#\text{E})^*.$$

Here $(\#\text{E})^*$ denotes many applications of (#E). For each moderated variable in $\pi \cdot_s (P'\{n{\mapsto}a\}) \bullet (t_i)$ we pull $(a\ b)$ into the term (conjugating a to b on the way down) to the site of the slice with some t_i, then use our proofs that $ds((a\ b), \mathbf{Id}) = \{a, b\}\#t_i$ to eliminate the transposition.

The behaviour of π at binders $\forall x$ and $\exists x$ in (43) is carefully chosen so that $(a\ b)$ is never introduced to a moderation of x. (πdiff) would be useless to reduce $\pi \circ \kappa \circ \pi^{-1}$ here because it uses capture-avoiding substitution.

We could simplify the definition (43) at the expense of making (πdiff) 'contextual'; allowing possibly capturing substitution. This is reasonable but would complicate the proof of soundness because (πdiff) then invents names for bound variables. To account for this we would have to quantify over suitable extensions of the valuations associated to frames, which looks complicated.

8.5. **Commutation cases.** The other essential cases are easy, standard, or both.

Recall from §8.1 that we work by induction on the closest intro/elim pair, and to bring critical pairs together we must show that ($\vee$E), ($\exists$E), (EqE), (πI), (πdiff), (New$\mathbb{A}$), and (Exhaust$\mathbb{A}$), commute downwards past ($\vee$E), ($\wedge$E1), ($\wedge$E2), ($\Rightarrow$E), ($\forall$E), ($\exists$E), (EqE), and ($\mathit{И}$E).

A simple and standard example is that a proof-fragment of the form ($\vee$E),($\wedge$E1)

$$\dfrac{\dfrac{\Gamma \vdash P \vee Q \quad \Gamma, P \vdash A \wedge B \quad \Gamma, Q \vdash A \wedge B}{\Gamma \vdash A \wedge B}\,(\vee\mathrm{E})}{\Gamma \vdash A}\,(\wedge\mathrm{E1}) \tag{26}$$

can be transformed to one of the form ($\wedge$E1),($\vee$E):

$$\dfrac{\Gamma \vdash P \vee Q \qquad \dfrac{\Gamma, P \vdash A \wedge B}{\Gamma, P \vdash A}\,(\wedge\mathrm{E1}) \qquad \dfrac{\Gamma, Q \vdash A \wedge B}{\Gamma, Q \vdash A}\,(\wedge\mathrm{E1})}{\Gamma \vdash A}\,(\vee\mathrm{E}) \tag{27}$$

We do not give all the examples. They do require weakening:

Lemma 8.6 (Weakening)**.** *If* Π *proves* $\Gamma \vdash C$ *then it proves* $\Gamma, \Delta \vdash C$ *for any set of formulae* Δ*.*

Thus the following is an admissible rule:

$$\dfrac{\Gamma \vdash C}{\Gamma, \Delta \vdash C} \tag{28}$$

Proof. By induction on proofs. □

Lemma 8.7. (πI) *commutes downwards through all elimination rules.*

Proof. The case of ($\forall$E) uses the fact that $\pi \cdot_s (P\{x{\mapsto}t\}) \equiv (\pi \cdot_s P)\{x{\mapsto}t\}$, the case of ($\exists$E) uses (24) and (22), the case of ($\mathit{И}$E) follows since n may be renamed so $\pi \cdot \mathit{И}n.\, P \equiv \pi \cdot \mathit{И}n.\, P$. Other cases are easy. □

Lemma 8.8. (πdiff) *commutes downwards through all elimination rules.*

Proof. The case of ($\forall$E) uses Lemma 8.3, that of ($\exists$E) uses (22). Other rules are easy, except for ($\mathit{И}$E) which we consider in more detail. Under consideration is a proof of the form

(29)

$$\dfrac{\dfrac{\dfrac{\Pi'}{\Gamma \vdash P\{x{\mapsto}\pi' \cdot_s t\}} \qquad \Gamma \vdash ds(\pi, \pi')\#t}{\Gamma \vdash P\{x{\mapsto}\pi \cdot_s t\}}\,(\pi\mathrm{diff}) \qquad \Gamma \vdash a\#t_i \quad (P\{x{\mapsto}\pi \cdot_s t\})/n = P' \bullet_{y_i} (t_i)}{\Gamma \vdash \mathit{И}n.\,(P\{x{\mapsto}\pi \cdot_s t\})}\,(\mathit{И}\mathrm{E})$$

For brevity write C for $P\{x{\mapsto}\pi \cdot_s t\}$. C/n is a least slice and n (chosen fresh) satisfies $n \notin \mathrm{A}(\pi, \pi', t)$. Thus the substitutions $\pi \cdot_s t$ and $\pi' \cdot_s t$ affect parts of C only within the t_i and we can write C/n as $P' \bullet (t_i'\{x{\mapsto}\pi \cdot_s t\})$ for appropriate t_i'. This suffices to allow us to perform the commutation. □

The commutation of (EqE) with (И E) is similar.

Similar commutation results hold of the other rules; (∃E), (EqE), (New$\mathbb{A}$), and (Exhaust$\mathbb{A}$). For example, commuting (Exhaust$\mathbb{A}$) with (∀E). Suppose we have a proof of the form

$$\dfrac{\dfrac{\dfrac{\Pi_n}{\bigwedge n \in S.\ \Gamma\{z{\mapsto}n\} \ \vdash\ \forall x.\, P\{z{\mapsto}n\}}\,(\text{Exhaust}\mathbb{A})}{\Gamma \ \vdash\ \forall x.\, P}\,(\forall\text{E})}{\Gamma \ \vdash\ P\{x{\mapsto}t\}.} \tag{30}$$

Here S contains $\mathrm{A}(\Gamma, P)$ and at least one fresh atom. Let $\mathrm{A}(t) = S'$ and without loss of generality (swapping if necessary) assume the fresh atom(s) a in S are also fresh for S' (and also assume $z \notin FV(t)$). Choose one such fresh a. Then for each $a' \in S'$ the proof $(a'\ a){\cdot}\Pi_a$ is by FM equivariance a valid proof of $\Gamma\{z{\mapsto}a'\} \ \vdash\ \forall x.\, P\{z{\mapsto}a'\}$. We now construct the following valid proof:

$$\dfrac{\dfrac{\dfrac{\Pi_n}{\Gamma\{z{\mapsto}n\} \ \vdash\ \forall x.\, P\{z{\mapsto}n\}}}{\bigwedge n \in S \cup S'.\ \Gamma \ \vdash\ P\{x{\mapsto}t\}\{z{\mapsto}n\}}\,(\forall\text{E})}{\Gamma \ \vdash\ P\{x{\mapsto}t\}}\,(\text{Exhaust}\mathbb{A}) \tag{31}$$

We never need to use the following observation in this paper, so we leave it to the reader to check that (New$\mathbb{A}$) commutes down through *all* rules — except for (∀I). Thus we can 'garbage-collect' all the fresh atoms right at the end of the proof (or reading bottom-up, 'generate' them), except for those atoms which are generated fresh for variables which are then consumed by a (∀I).

8.6. **Proof normalisation.**

Theorem 8.9 (Proof normalisation)**.** *Any derivable judgement of Fresh Logic has a proof in a normal form.*

Sketch proof. We proceed in the standard way on proof-size. We bring a closest critical pair closer together with commutation rules, then eliminate it with an essential case. □

9. Completeness

9.1. (Small)**.** Fresh Logic is not complete with respect to the semantics considered in §7. This is because no axiom insists that there be *countably* many atoms (as is the case in our class of models); perhaps there are uncountably many, and all elements have countable, rather than finite, support.

In order to make Fresh Logic complete[3] we must add an axiom to the effect that 'atoms are countable'. This is not expressible in First-Order Logic [17], but in the case of Fresh Logic we can get around this, exploiting the fact that we represent atoms by a class of what are, essentially, constants.

We add one axiom:

$$\dfrac{\bigwedge L \in \text{CoFin}\mathbb{A}.\ \Gamma, L\#t \ \vdash\ P}{\Gamma \ \vdash\ P}. \tag{Small}$$

Here t is any term, $\bigwedge L \in \text{CoFin}\mathbb{A}$ indicates a(n infinite) conjunction of proofs, one for each cofinite set of atoms L, and $L\#t$ denotes the (infinite) list of hypotheses $l\#t$ for each $l \in L$.

[3]— if we care; for many purposes a reasonable-looking sound semantics is quite good enough.

(Small) changes to our notion of proof in two ways: sequents may mention infinitely many assumptions, and instances of proof-rules may have infinitely many hypotheses above the line. To adjust for this, we need to introduce a notion of freshness for *sets of formulae*, and for *derivations*.

Recall from §7.4 that given atoms a, b and a formula P we write $(a\ b)\cdot P$ for the formula obtained by swapping a and b in the syntax of P. Given a (possibly infinite) set of formulae Γ write $(a\ b)\cdot\Gamma$ for the same operation applied pointwise.

Then the infinite sequents needed to write (Small) still have **finite support** in the sense that

$$\text{И}a.\, a\#(\Gamma\cup\{P\})$$

where $a\#x$ means $\text{И}b.\,(b\ a)\cdot x = x$, and И here means 'for all but finitely many atoms'. Write

$$S(\Gamma) \quad \text{for} \quad \{a\in\mathbb{A}\mid \neg a\#\Gamma\}.$$

As a matter of notation write $S(\Gamma, P)$ for $S(\Gamma\cup\{P\})$. Say Γ **has finite support** when $S(\Gamma)$ is finite.

We extend the notion of sequent to allow $\Gamma \vdash P$ to be a sequent even if Γ is infinite *provided that*:

- Γ has finite support.
- Γ mentions finitely many variable symbols x, y, z (see below for explanation).

To account for possibly infinite sets of assumptions in sequents $\Gamma \vdash P$, modify (New$\mathbb{A}$) and (Exhaust$\mathbb{A}$) as follows:

$$\frac{\Gamma, a\#ts \vdash C}{\Gamma \vdash C} \qquad a\#\Gamma, C \tag{32}$$

$$\frac{\bigwedge n\in S.\,\Gamma\{x\mapsto n\} \vdash P\{x\mapsto n\}}{\Gamma\{x\mapsto t\} \vdash P\{x\mapsto t\}} \qquad S(\Gamma,P)\subsetneq S \tag{33}$$

It is an easy exercise to verify that if Γ is finite, the condition $S(\Gamma,P)\subsetneq S$ reduces to the condition $\mathrm{A}(\Gamma,P)\subsetneq S$, and $a\#\Gamma, C$ reduces to $a\notin\mathrm{A}(\Gamma,C)$, and in general that if Γ is finite then $S(\Gamma)$ is precisely the set of atoms mentioned in Γ which is in our notation written $\mathrm{A}(\Gamma)$.

Given a proof Π write $(a\ b)\cdot\Pi$ for the proof obtained by swapping a and b throughout the syntax of Π. Write $a\#\Pi$ when $\text{И}b.\,(b\ a)\cdot\Pi\equiv\Pi$. Then proofs (even if they mention (Small)) have **finite support** in the sense that

$$\text{И}a.\, a\#\Pi.$$

Write $S(\Pi)$ for $\{a\in\mathbb{A}\mid\neg a\#\Pi\}$.

We now observe that

$$S(\{\Gamma, L\#t, P\}\mid L \text{ cofinite}\}) = S(\Gamma, P),$$

so that $S(\Pi)$ is still finite, even if Π mentions (Small).

Note finally that, even in the presence of (Small), proofs remain **finitely deep**. That is, given any proof there is some number n such that if we traverse the proof from bottom to top, the length of our path is no greater than n.

To prove that Fresh Logic with (Small) is still normalising, it now suffices to observe that (Small) may be commuted down through all the elimination rules in (20), and to observe that 'most of the time', formally that means for cofinitely many hypotheses of (Small), if some proof-transformation may be applied to one of those hypothesis, the same proof-transformation may be applied to them all (it suffices to treat *en bloc* those such that L is disjoint from $S(\Gamma,P)$).

The proofs up till now generalise (for example Lemma 8.6, which we stated for sets of formulae in anticipation of this development) and we conduct the rest of the

paper in this more general setting, though we will not henceforth specify that we are using the generalised form of a result.

We *still* insist that Γ mention finitely many different variable symbols $x \in \mathbb{V}$. Otherwise proofs might break because Γ might mention all of $\mathbb{V}$ and we could not guarantee there exists $x \notin FV(\Gamma)$ (as required by some rules, like ($\forall$I)).

It is now easy to verify that Fresh Logic + (Small) still normalises (and in finitely many steps); since (Small) introduces no new essential cases, we only need check that (Small) commutes down through all elim-rules in (20), and that any proof-transformations occurring in its hypotheses may be treated *en bloc* for 'most' of the time.

Lemma 9.1. (Small) *is sound with respect to the semantics of* §7.

Proof. Fix some model $\mathcal{M}$ and $\alpha \in \mathcal{M}$. Suppose for all $L \in \mathrm{CoFin}\mathbb{A}$ we know if $\alpha \Vdash \Gamma, L\#t$ then $\alpha \Vdash P$. Suppose also that $\alpha \Vdash \Gamma$.

Choose L disjoint from $S(x)$ for all $x \in U_\alpha$, and disjoint also from $\mathrm{A}(t)$. It is easy to verify by induction on the syntax of t that $L\#[\![t]\!]_\alpha$ so $\alpha \Vdash L\#t$. By assumption therefore $\alpha \Vdash P$ as required. □

Notation 9.2. Say a context Γ is **consistent** when $\Gamma \nvdash \bot$.

(Small) gives us completeness via the following theorem:

Theorem 9.3. *Let* Γ *in some language of Fresh Logic* $\mathcal{L}$ *be consistent and suppose* $\Gamma \vdash \exists x : \mathtt{X}.\, P$ *is derivable. Then there exists* $n \in \mathbb{N}$ *and list of* n *atoms* $(a_i)_1^n$ *such that we may extend* $\mathcal{L}$ *with a fresh constant* $\mathtt{f} : \mathtt{A}^n \to \mathtt{X}$ *to* $\mathcal{L}'$ *and in this new language* $\Gamma, P\{x{\mapsto}\mathtt{f}(a_i)\}$ *is consistent.*

Similarly, if Γ *is consistent and* $\Gamma \nvdash \forall x.\, P$*, then there exists* n*,* $\mathtt{f}$*, and* $(a_i)_1^n$ *such that we may extend* $\mathcal{L}$ *as above, and in this new language* $\Gamma \nvdash P\{x{\mapsto}\mathtt{f}(a_i)\}$.

Proof. Consider the context $\Gamma, P\{x{\mapsto}\mathtt{f}(a_i)\}$. $\mathtt{f}$ has no axioms so if $\Gamma, P\{x{\mapsto}\mathtt{f}(a_i)\} \vdash Q$ is derivable in $\mathcal{L}'$ where Q is a sentence of $\mathcal{L}$ (i.e. does not mention $\mathtt{f}$), then the proof may be transformed into one of $\Gamma, P, L\#x \vdash Q$, for $L \stackrel{\text{def}}{=} \mathbb{A}\backslash\{a_i \mid 1 \leq i \leq n\}$.

Now suppose $\Gamma, P\{x{\mapsto}\mathtt{f}(a_i)\} \vdash \bot$ for all n. Then by the transformed proofs we know for all $L \in \mathrm{CoFin}\mathbb{A}$ that $\Gamma, P, L\#x \vdash \bot$. By (Small) we may deduce $\Gamma, P \vdash \bot$. Since $\Gamma \vdash \exists x.\, P$ we may apply ($\exists$E) to deduce $\Gamma \vdash \bot$, contradicting the assumption that Γ is consistent.

The last part is similar. □

9.2. (Exhaust$\mathbb{A}$) **and** (New$\mathbb{A}$)**.** (Exhaust$\mathbb{A}$) and (New$\mathbb{A}$) give theorems similar in spirit to Theorem 9.3, and also needed for the completeness proof:

Theorem 9.4. *Suppose* Γ *is a consistent context and suppose* $t : \mathtt{A}$ *is any term. Then there exists* $a \in \mathbb{A}$ *such that* $\Gamma, t = a$ *is consistent.*

Proof. If $\Gamma, t = a \vdash \bot$ for every a, then $\Gamma, t = x \vdash \bot$ by (Exhaust$\mathbb{A}$) ($x \notin FV(\Gamma, t)$). Thus $\Gamma \vdash \forall x.\, x \neq t$ and in particular $\Gamma \vdash t \neq t$, whence $\Gamma \vdash \bot$, which contradicts consistency of Γ. □

This ties in with our definition of frame α, which insisted that $[\![\mathtt{A}]\!]_\alpha \cong \mathbb{A}$ so that $[\![t]\!]_\alpha = a$ for some a.

Lemma 9.5. *If* Γ *is consistent then for fresh* $a\#\Gamma$ *and any list of terms* ts*,* $\Gamma, a\#ts$ *is consistent.*

Proof. By (New$\mathbb{A}$). □

9.3. **Remark on literal equality.** Recall from §7.1 that we commented on the fact that frame maps need not be injective, and equality is interpreted as literal equality rather than some relation with special properties.

We *have* to do this. When below we build frames out of so-called prime theories (thus, out of syntax) to prove completeness, we interpret terms by their equivalence class under provable equality. If instead we interpreted terms as themselves, a variable symbol x would not have finite support (necessary for a Nominal Set semantics), since $\pi \cdot x$ is not the same term as x for any $\pi \neq \mathbf{Id}$, though they may be provably equal.

9.4. **Prime theories.**

Definition 9.6. A set Φ of (possibly open) sentences is a **prime theory** when

(i) Φ is deductively closed.
(ii) Φ mentions only finitely many variable symbols $x \in \mathbb{V}$.
(iii) If $P \vee Q \in \Phi$ then $P \in \Phi$ or $Q \in \Phi$ (or both).
(iv) If $\exists x.\, P \in \Phi$ then $P\{x{\mapsto}t\} \in \Phi$ for some term.
(v) If t has sort $\mathtt{A}$ then for some $a \in \mathbb{A}$, $t = a \in \Phi$.
(vi) If $\mathit{И}n.\, P \in \Phi$ then, for $P/n = P' \bullet_{y_i} (t_i)_1^k$, there is an $a \in \mathbb{A}$ such that $a\#t_i \in \Phi$ for $1 \leq i \leq k$, and $P\{n{\mapsto}a\} \in \Phi$.

Lemma 9.7. *If* $\Gamma \nvdash P$ *then there is a prime theory* $\Phi \supseteq \Gamma$ *such that* $P \notin \Phi$.

Proof. Write U_Φ for the variables mentioned in Γ and P. We build an ascending chain of theories $\Gamma_0{=}\Gamma \subseteq \Gamma_1 \subseteq \ldots$. Choose some ordering on constants, predicate symbols, variables, and atoms, with free variables in U_Φ. Use that to enumerate terms and formulae of Fresh Logic with free variables in U_Φ. At each step, we use this enumeration to extend Γ_k to Γ_{k+1} according to the following recipe:

(i) $\Gamma_0 = \Gamma$.
(ii) $\Gamma_{4k'} \mapsto \Gamma_{4k'+1}$: take the least $P \vee Q$ such that $P \vee Q \in \Gamma_{4k'}$ and $P, Q \notin \Gamma_{4k'}$. Add the least of the two to give $\Gamma_{4k'+1}$, let $\Gamma_{4k'+1} = \Gamma_{3k}$ otherwise.
(iii) $\Gamma_{4k'+1} \mapsto \Gamma_{4k'+2}$: take the least $\exists x.\, P$ such that for no t is $P\{x{\mapsto}t\} \in \Gamma$, if one such exists. If one such exists, adjoin $P\{x{\mapsto}\mathtt{f}(a_i)\}$ for some fresh constant symbol $\mathtt{f}$ and atoms a_i preserving consistency if $\Gamma_{4k'+1}$ is consistent. This is possible by Theorem 9.3.
(iv) $\Gamma_{4k'+2} \mapsto \Gamma_{4k'+3}$: take the least $t : \mathtt{A}$ such that for no $a \in \mathbb{A}$ is $t = a \in \Gamma$, if one such exists. If one such exists, adjoin $t = a$ for an a, preserving consistency. This is possible by Theorem 9.4.
(v) $\Gamma_{4k'+3} \mapsto \Gamma_{4(k'+1)}$: take the least $\mathit{И}n.\, P$ such that $\Gamma_{4k'+3}$ does not satisfying the closure condition (which we do not rewrite) for it, if one such exists. If one such exists there is an atom a such that $\Gamma_{4k'+3}, a\#FV(t_i)$ is consistent by Lemma 9.5. This new context entails $P\{n{\mapsto}a\}$ by ($\mathit{И}$E) and we let $\Gamma_{4k'+4}$ be $\Gamma_{4k'+3}, a\#t_i, P\{n{\mapsto}a\}$, or $\Gamma_{4k'+3}$, as appropriate.

Take $\Phi = \bigcup_i \Gamma_i$, which we can verify is a prime theory by construction (deductive closure is 'for free' because if $\Phi \vdash P$ then $\Phi \vdash P \vee P$ so $P \in \Phi$). □

Lemma 9.8. *In a prime theory* Φ*;*

(i) *If* $t : \mathtt{A}$ *then* $\Phi \vdash t = a$ *for some* $a \in \mathbb{A}$.
(ii) *If* $x : \mathtt{X} \in U_\Phi$ *then* $\Phi \vdash x = \mathtt{f}(as)$ *for some function symbol* $\mathtt{f}$ *and atoms* as.
(iii) *For* x *as above,* $\Phi \vdash a\#x$ *for cofinitely many* a.

Lemma 9.9. *A prime theory* Φ *in a language* $\mathcal{L}$ *generates a frame, which we also write* Φ*, as follows:*

(i) $[\![\mathtt{X}]\!]$ *is the set of equivalence classes of* $t : \mathtt{X}$ *such that* $FV(t) \subseteq U_\Phi$*, up to provable equality* $t = t' \in \Phi$*. Write such a class* $[\![t]\!]$*. The permutation action is given by the action on the representatives;* $\pi \cdot [\![t]\!] \stackrel{\text{def}}{=} [\![\pi \cdot_s t]\!]$.
(ii) $[\![\mathtt{c}]\!]$ *is the function mapping* $\langle [\![t_1]\!], \ldots \rangle$ *to* $[\![c(t_1, \ldots)]\!]$.
(iii) *The canonical isomorphism* $[\![\mathtt{A}]\!] \cong \mathbb{A}$ *is given by* $[\![t]\!]$ *maps to that a such that* $\Phi \vdash t = a$.
(iv) $[\![p]\!]$ *is the set of* $\langle [\![t_1]\!], \ldots \rangle$ *such that* $p(t_1, \ldots) \in \Phi$.
(v) $[\![\top]\!] = \{*\}$ *and* $[\![\bot]\!] = \emptyset$.

Accessibility $\Phi \sqsubseteq \Phi'$ *holds in a natural way when* Φ *is a prime theory for* $\mathcal{L}$ *and* Φ' *a prime theory for a superlanguage* $\mathcal{L}'$ *of* $\mathcal{L}$*, and* $U_\Phi \subseteq U_{\Phi'}$*, and* $\Phi \subseteq \Phi'$.

Proof. We verify every clause of Definition 4.1. □

We shall be imprecise and write $t \in \Phi$ to mean t is a sentence in the language $\mathcal{L}$ of Φ, such that $FV(t) \subseteq U_\Phi$.

Notation 9.10. Take some language $\mathcal{L}$ of Fresh Logic. Prime theories Φ in languages $\mathcal{L}' \geq \mathcal{L}$, and their maps, form a Kripke model of $\mathcal{L}$ in the sense defined above. Henceforth we shall write $\Phi \Vdash P$, and this will mean model-theoretic entailment with respect to this model.

Lemma 9.11. $\Phi \Vdash P$ *if and only if* $P \in \Phi$.

Proof. If $P \in \Phi$ then $\Phi \vdash P$ so by soundness Theorem 7.7, $\Phi \Vdash P$.

(i) Suppose $P \equiv p(ts)$. $\Phi \Vdash P$ when $p(ts) \in \Phi$ by definition.
(ii) Suppose $P \equiv P_1 \vee P_2$. Then either P_1 or P_2 is in Φ, and we apply the induction hypothesis. The case of $\wedge$ is similar but even simpler.
(iii) Suppose $P \equiv P_1 \Rightarrow P_2$. $\Phi \Vdash P$ when

$$\begin{array}{c} \forall \Phi' \sqsubseteq \Phi.\, \Phi' \Vdash P_1 \implies \Phi' \Vdash P_2 \\ P_1 \in \Phi' \qquad\qquad P_2 \in \Phi'. \end{array} \tag{34}$$

(The second line rewrites parts of the first using the inductive hypothesis.)

If $P_1 \Rightarrow P_2 \notin \Phi$ then $\Phi, P_1 \nvdash P_2$ and Lemma 9.7 says there is $\Phi' \sqsubseteq \Phi, P_1$ such that $P_2 \notin \Phi'$. Therefore $\Phi \nVdash P_1 \Rightarrow P_2$.

(iv) Suppose $P \equiv \forall x : \mathtt{X}.\, P_1$. $\Phi \Vdash P$ when

$$\begin{array}{c} \forall \Phi' \sqsubseteq \Phi. \forall t : \mathtt{X} \in \Phi'.\, \Phi' \Vdash P_1\{x \mapsto t\} \\ P_1\{x \mapsto t\} \in \Phi'. \end{array} \tag{35}$$

If $\forall x.\, P_1 \notin \Phi$ then $\Phi \nvdash \forall x.\, P_1$ and by Theorem 9.3 for some constant $\mathtt{f}$ in a superlanguage $\mathcal{L}' \sqsubseteq \mathcal{L}$ and atoms as, $\Phi' \nvdash P\{x \mapsto \mathtt{f}(as)\}$.

(v) Suppose $P \equiv \mathit{N}n.\, P_1$ where n is chosen fresh. $\Phi \Vdash P$ when $\Phi \Vdash P_1$. By induction hypothesis $\Phi \vdash P_1$ and by properties of the deduction system and prime theories, $\Phi \vdash \mathit{N}n.\, P_1$.
(vi) Suppose $P \equiv t_1 = t_2 : \mathtt{X}$. $\Phi \Vdash t_1 = t_2$ when $[\![t_1]\!] = [\![t_2]\!] \in [\![\mathtt{X}]\!]$, which is the case precisely when $\Phi \vdash t_1 = t_2$.
(vii) Suppose $P \equiv t_1 : \mathtt{A} \# t_2 : \mathtt{X}$. Suppose $\Phi \Vdash t_1 \# t_2$. $\Phi \vdash t_1 = a$ for some $a \in \mathbb{A}$ so $\Phi \Vdash a \# t_2$ and $a \# [\![t_2]\!]$.

Then for cofinitely many b, $(b\ a) \cdot [\![t_2]\!] = [\![t_2]\!]$, thus for cofinitely many b, $\Phi \vdash (b\ a) \cdot_s t_2 = t_2$. Choose a b such that we can apply ($\mathit{N}\mathtt{I}$) to deduce $\Phi \vdash \mathit{N}b.\, (b\ a) \cdot_s t_2 = t_2$. Using (**D6**) from Figure 4 we conclude $\Phi \vdash a \# t_2$ and so $\Phi \vdash t_1 \# t_2$.

□

Extending notation from Notation 7.6, write $\Gamma \Vdash P$ when for all models $\mathcal{M}$, $\Gamma \Vdash_{\mathcal{M}} P$.

Theorem 9.12 (Completeness)**.** $\Gamma \vdash P \iff \Gamma \Vdash P$.

Proof. Suppose $\Gamma \vdash P$. Then $\Gamma \Vdash P$ by soundness Theorem 7.7. Conversely if $\Gamma \nvdash P$, there is a prime theory $\Phi \supseteq \Gamma$ such that $P \notin \Phi$ by Lemma 9.7 and by Lemma 9.11 $\Phi \nVdash P$. □

10. Alternative formulations of Fresh Logic

We made several design decisions in Fresh Logic, we now discuss how we might do things differently, and if that does not work, why.

10.1. **No swappings or #.** We might envisage a calculus without swappings and #. The И-intro and elim rules would be controlled by a top-level quantifier context consisting of a list of ∀ and И quantifiers whose ordering encodes freshness information in a way we have already seen in the theory of Fresh Logic in (D4) in §6, and [8].

For example, in such a logic we might *axiomatise* # as $\forall x.\, И a.\, (\vdash a\#x)$.

In (∀E), x is instantiated for t. For Lemma 8.3 to work we must insist as a side-condition of (∀E) that we only substitute a t which satisfies the freshness conditions encoded in the position of x in the quantifier context, which we can only do with conditions on the positions of where $y \in FV(t)$ appear in the quantifier context. The details of getting this to work seem to become extremely complex.

The problem is in Lemma 8.3. In the modified logic sketched above $\{x{\mapsto}t\}$ is only admissible when t satisfies the freshness constraints on x encoded in the quantifier context. It is difficult to build a framework to formally deduce when this is so, and would be even more difficult in the presence of term-formers binding atoms, e.g. atoms-abstraction $[a]x$ [12, 26], which for simplicity we omitted from Fresh Logic. See the comparison with Generic Judgements in the Conclusion.

10.2. **Variables $x : \mathbb{A}$ in swappings.** As it stands Fresh Logic only allows swappings by atoms $(a\ b)$—not swappings by variables (of type $\mathtt{A}$), which would be $(x\ y)$. Note that we can always determine $a = b$ or $a \neq b$ by whether $a \equiv b$, so we can always simplify an expression of the form $(a\ b) \cdot_s c$. This is why Fresh Logic has no moderated atoms. This is not so of $x = y$ or $x \neq y$, so we must also consider nested swappings such as $(((x'\ y') \cdot_s x)\ (u\ v) \cdot_s y)$.

Having constructed the normalisation and completeness proofs, it appears Fresh Logic would sustain this extension. The difference which would be probably most significant is that (πdiff) changes completely and splits into three (rules implementing) identities

$$(36)\quad (x\ x) \cdot u = u \qquad (x\ y) \circ (x\ y) \cdot u = u$$
$$(x\ y) \circ (x'\ y') \cdot u = (x'\ y') \cdot (((x'\ y') \cdot x)\ ((x'\ y') \cdot y)) \cdot u$$

which characterise the group generated by the swappings $(x\ y)$ [15, Beispiel 19.7], [26, p.10].

10.3. FM **swapping in this document.** This paper is an analysis of a particular syntax system Fresh Logic. It has two kinds of variable symbol; $x \in \mathbb{V}$ and $a \in \mathbb{A}$. Both can be bound, by $\forall, \exists$ and И respectively.

We encounter the usual problems with capture-avoidance and renaming. Sometimes we are 'traditional' about it and say "α-rename to assume…". Sometimes it is more convenient to be rigorous and talk about swapping. This is particularly with (Exhaust$\mathbb{A}$) and because atoms a appear in both Fresh Logic syntax *and* its semantics, there are points in this document where swappings and freshnesses are rather beautifully applied to judgements and functions with both semantic and syntactic parts, such as '$\alpha \Vdash P$' or '$[\![t]\!]_\alpha$' (Theorem 7.7, Lemma 8.3).

$$(\text{И I}) \quad \frac{\Gamma \vdash P \quad \Gamma \vdash a\#t_i \ (i=1,\dots,k)}{\Gamma \vdash \text{И}a.\,P} \qquad P/a = P' \bullet_{y_i} (t_i)_1^k$$

$$(\text{И E}) \quad \frac{\Gamma \vdash \text{И}a.\,P \quad \Gamma \vdash a\#t_i \ (i=1,\dots,k)}{\Gamma \vdash P} \qquad P/a = P' \bullet_{y_i} (t_i)_1^k$$

$$(\forall \text{I}) \quad \frac{\Gamma \vdash P}{\Gamma \vdash \forall x.\,P} \qquad x \notin FV(\Gamma)$$

$$(\forall \text{E}) \quad \frac{\Gamma \vdash \forall x.\,P}{\Gamma \vdash P\{x\mapsto t\}} \qquad x \notin FV(\Gamma, C),\ P\{x\mapsto t\} \text{ defined}$$

$$(\exists \text{I}) \quad \frac{\Gamma \vdash P\{x\mapsto t\}}{\Gamma \vdash \exists x.\,P} \qquad x \notin FV(\Gamma)$$

$$(\exists \text{E}) \quad \frac{\Gamma \vdash \exists x.\,P \quad \Gamma, P \vdash C}{\Gamma \vdash C} \qquad x \notin FV(\Gamma, C),\ P\{x\mapsto t\} \text{ defined}$$

FIGURE 5. Rules of Fresh Logic with И non-binding

In fact 'secretly' we have used an FM treatment of variables and binding, $\mathbb{V}$ and $\mathbb{A}$, throughout, simply because we understand it well, it works, and it is completely rigorous. We still present things traditionally; we write $x \notin FV(\Gamma)$ in ($\forall$I) instead of 'x fresh'.

10.4. **Иa does not bind a.** In the construction of Fresh Logic syntax, И binds the abstracted atom, so that for example И$n.\,n\#x$ and И$n'.\,n'\#x$ are equal formulae just like $\forall x.\,a\#x$ and $\forall x'.\,a\#x'$.

The essential case of И uses swapping to rename a to b if И was introduced using a and eliminated using b. However, let us backtrack: atoms $a, b, c, n, n' \dots$ are constants. How can you bind a constant?

The answer adopted in the bulk of this paper is: atoms a, b, c, n are not constants but variables of a particular kind which denote distinct elements of $\mathbb{A}$. For brevity write their denotation also $a, b, c, n \dots \in \mathbb{A}$ and call them atoms. Atoms behave like constants in that $a \neq b$ if $a \not\equiv b$, but they can still be bound.

In this subsection we consider an alternative. Consider a Fresh Logic in whose syntax И does *not* bind, so И$a.\,a\#x$ and И$b.\,b\#x$ are distinct formulae. There is now a notion of '**syntactic free**' and '**syntactic bound**' atom of a formula.

Substitution $\{x\mapsto t\}$ is problematic because syntactic capture is now possible, e.g. in (И$a.\,a\#x)\{x\mapsto a\}$. We do not introduce syntactic α-renaming of syntactic bound atoms, instead we declare substitution undefined in this case. Similarly permutation (43) changes in the clause of И, so $\pi \cdot_s \text{И}a.\,P$ is equal to И$a.\,\pi \cdot_s P$ if $a \notin \mathrm{A}(\pi)$ and is undefined otherwise.

The technical benefit is that the И intro- and elim-rules, and their proof-theory, are simplified. We give the parts of Fresh Logic which must change in Figure 5.

The essential case of И is now just as follows:

$$(37) \quad \dfrac{\dfrac{\dfrac{\Pi'}{\Gamma \vdash (P' \bullet_{y_i} (t_i)_{i=1}^k)}\ \Gamma \vdash a\#t_i}{\Gamma \vdash \text{И}a.\,P' \bullet (t_i)}(\text{И I}) \qquad \Gamma \vdash a\#t_i}{\Gamma \vdash (P' \bullet (t_i))}(\text{И E})\ .$$

We just transform this to Π'.

This rule is compellingly simple but the $\forall$ and $\exists$ rules change, violating the Slogan of the Introduction—though not completely because for P containing no $И$ quantifiers $P\{x{\mapsto}t\}$ is always defined and as usual.

An equivalent formulation of the same condition was suggested by Pitts, we devote this paragraph to it. Recall from §7.4 the discussion of meta-level swapping. Then we can define $\alpha \Vdash И a.\, P$ when for fresh n, $(n\ a) \cdot \alpha \Vdash P$. The two formulations are equivalent: $(n\ a)\cdot\alpha \Vdash P$ if and only if $\alpha \Vdash P\{a{\mapsto}n\}$. The proof follows since for fresh n, $(n\ a) \cdot P$ equals $P\{a{\mapsto}n\}$ because n is fresh and does not occur in P. We use $\{a{\mapsto}n\}$ here because it is more concrete for the reader coming from outside FM theory.

The theory still stands mostly unchanged with non-binding $И$, but the proofs and definitions do shift a little, and in quite interesting ways. We now discuss how.

The clause for $И$ in Definition 7.2 changes because a is not bound in the formula and so cannot just be chosen fresh, like n was. The solution is simple: $\alpha \Vdash P$ when for fresh n, $\alpha \Vdash P\{a{\mapsto}n\}$.

In Lemma 7.5 we must insist that $\pi \cdot P$ be well-defined, which need not be the case as discussed above. The clause for $И$ becomes: Suppose $P \equiv И a.\, P'$ and $\alpha \Vdash И a.\, P'$. Choose n fresh. Then $\alpha \Vdash P'\{a{\mapsto}n\}$ so by induction hypothesis $\alpha \Vdash \pi \cdot_s (P\{a{\mapsto}n\})$. Using the assumptions that $a, n \notin \mathrm{A}(\pi)$, we deduce $\alpha \Vdash \pi \cdot_s И a.\, P$.

We have already discussed the essential case of $И$ in this formulation.

The clause for $И$ in Definition 9.6 is unchanged. The one in Lemma 9.11 becomes: suppose $P \equiv И a.\, P_1$. Choose n fresh. $\Phi \Vdash P$ when $\Phi \Vdash P_1\{a{\mapsto}n\}$. By induction hypothesis $\Phi \vdash P_1\{a{\mapsto}n\}$ and by properties of the deduction system and prime theories, $\Phi \vdash И a.\, P_1$.

So we see that the only difference using a non-binding $Иa$ is that we have to choose n fresh and explicitly substitute a for n, whenever we unpack the quantifier. The interaction with substitution also engenders side-conditions on $\forall$-elimination and $\exists$-introduction rules.

11. Conclusions

We have presented 'Fresh Logic', a first-order logic enriched with $И$ (for 'fresh'), $\#$ (for 'fresh for'), and $(a\ b)$ (for 'rename').

We provide natural-deduction and sequent systems such that the FM-free core is unmodified First-Order Logic. The logic is complete with respect to a natural FM semantics, and (all?) the well-known tautologies of FM theory can be derived in it. To our knowledge this is only the second proof-theory of logics for handling names and binding, in the sense of proof normalisation or cut-elimination. The first was the Tiu-Miller logic of Generic Judgements, discussed below.

The most immediate contribution of this paper is to show that $И$ is not incompatible with proof theory. Looking beneath this result, the essential case for $И$ in proof normalisation teaches us about its relation with $\#$ and swapping, whose presence in the logic is necessary for that essential case to work. The other major construction, a sound and complete semantics, relates the logical structure of Fresh Logic to a good notion of validity and leads us to two interesting deduction rules (Exhaust$\mathbb{A}$) and (Small).

There are some beautiful interactions between FM atoms in syntax, proofs, and semantics. Some of them (see also §7.4) are:

(i) Equivariance of validity of deduction trees Π in Lemma 8.3.
(ii) Swapping applied to models α in Theorem 7.7.
(iii) In (Exhaust$\mathbb{A}$) we need only check the atoms appearing in the conclusion and one fresh atom, rather than atoms occurring in the support of the

denotations of the variable symbols in the conclusion (of course not, since denotation is distinct from derivability, but for comparison Generic Judgements build something corresponding to support into derivability via the local and global signatures).

11.1. **Related work.**

Work based on FM semantics. Pitts' **Nominal Logic** [26] and Fresh Logic are both formal systems based on first-order logic and FM-sets semantics. Nominal Logic is classical, Fresh Logic is intuitionistic, but that difference is inessential.

Nominal Logic was created to show that first-order logic can usefully describe FM sets, which previous work had shown can model names and binding in abstract syntax. It is interested in expressivity and soundness, especially with respect to modelling object-level syntax. It is not interested in other things, such a proof-theory; thus for example it uses a Hilbert-style axiom system.

This paper takes the expressivity for granted; we do not, for example, present a Fresh Logic axiomatisation of an abstract machine for evaluating λ-calculus terms, or a π-calculus transition system. We know we could: previous work has done these things outside of a proof-theoretic framework and that work used the same semantics as Fresh Logic, for which Fresh Logic is sound and (suitably strengthened) complete. We are also confident these things could be cleanly expressed, because the syntax of Fresh Logic is not so different from that of existing FM-based systems.

Another Hilbert-style axiomatisation of FM-sets is in [10]. This similar to Nominal Logic, but as a formal development in Isabelle of the full theory of FM-sets as described in [12, 9].

Urban, Pitts, and Gabbay investigate **Nominal Unification** [28]. We quote that work in technical aspects, in particular the notions of moderated variable $\pi \cdot x$ and difference set $ds(\pi, \pi')$.

Work based on Higher-Order Abstract Syntax. The basic principles of Higher-Order Abstract Syntax (HOAS) are presented in [24]. This is most concerned with expressivity, and corresponds in the context of HOAS to [12] in the context of FM.

One example of an axiomatic system based on HOAS (there are others) is the Theory of Contexts [13, 19]. This is a consistent [3] set of axioms such as can be stated in COQ and then programmed with. This corresponds to [10] or [25]; proof theory is not the issue. There is a semantics based on presheaves [19]. Fresh Logic semantics is FM-sets, which can also be viewed as presheaves [12]. However, the two presheaf semantics are not directly comparable. A detailed analysis of how and why is not possible here, and is probably slightly premature since the technologies are still maturing.

The HOAS-based work most closely corresponding to Fresh Logic is by Tiu and Miller [20]. They develop a sequent system (with cut-elimination) **Generic Judgements**. They use it to reasoning on names and binding, so the application domain is similar to that of Fresh Logic. Generic Judgements have a quantifier ∇ which is similar to N in some respects (see self-duality below) and is used in much the same way. Generic judgements lack the complication (?) of anything like $\#$ and $(a\ b)$, though they do enrich sequents with annotations which are essentially meta-level ∇-quantifiers. This makes the intro-rules for ∇ particularly simple; the structure is built into the sequents.

∇ picks a 'generic' name. A generic name in Generic Judgements is one such that we 'promise' not to use any special properties it may have. A connection to FM is that a fresh name in FM is generic *amongst fresh names* by the some/any property. It may be that this is the only commonality.

It is interesting to note that Fresh Logic and Generic Judgements approach the same problem from opposite directions. In Generic Judgements the meta-level annotations keep track of which generic elements are in use, whereas in Fresh Logic object-level freshness keeps track of which atoms are *not* in use.

Of course, in this paper Fresh Logic is interested in proof-theory, soundness, and completeness. We do, therefore, see a lot of proof-theory and models. We have not considered the logic-programming aspects, like Generic Judgements (which would be interesting future work).

Our inspiration to pursue Fresh Logic dates back to a conversation with Miller and Tiu in INRIA in October 2002. Our first attempts to formulate it tried to just use $\mathsf{И}$, imitating the structure of sequents of Generic Judgements; a judgement consisted of a context and a conclusion, wrapped in a top-level list of nested $\mathsf{И}$ and $\forall$ quantifiers, for example $\forall x.\, \mathsf{И}a.\, (\vdash\ a\#x)$.

We found that in the context of FM sets this was inexpressive and furthermore it gave bad proof theory. The underlying reason seems to be in the comment we made above, that HOAS-based techniques keep track of the names in a term (local and global signatures in Generic Judgements), while FM-based techniques keep explicit track of the names not in a term (freshness #). Once we added freshness # and swappings, all the problems vanished; we think this is an interesting observation, that $\mathsf{И}$ *needs* freshness and swapping to proof-theoretically work.

More concretely, there were serious problems with Lemma 8.3 because, in the system sketched above, a substitution of t for x is only admissible when t satisfies all the freshness constraints of x. In the system as it stands those freshness constraints are explicit assumptions of the for $a\#x$. If they no longer hold after substitution, e.g. if $t = a$, then the judgement as a whole is trivially valid since $a\#a$ entails anything.

There are other approaches to syntax and binding and all have papers exploring how well an object-level system (a π- or λ-calculus, usually) can be expressed and programmed on. To our knowledge (and we may be wrong) Fresh Logic and the logic of Generic Judgements are the only work on the *proof-theory* of names and binding, in the sense of normalisation and cut-elimination.

One possible exception is work by Menni [18]. This analyses a class of $\mathsf{И}$-like quantifiers in the context of categorical logic.

Work derived from logics for structured data and trees. Caires and Cardelli introduced **Spatial Logics** for reasoning about π-calculus processes [5]. These have two sorts, atoms and processes, and include modalities and constructors specialised to this domain. At its core is a logic with $\mathsf{И}$ similar to Fresh Logic. This makes sense since the novel feature of their logics was precisely an FM-style $\mathsf{И}$.

One significant technical difference is that Fresh Logic has atoms a and variables x of sort $\mathtt{A}$ where Spatial Logic only allows variables. On the other hand Fresh Logic only allows swappings by atoms $(a\ b)$, where Spatial Logic (having no atoms) allows swappings by variables $(x\ y)$, and also (since equality of variables need not be decidable for a given context) nested swappings $((x'\ y') \cdot_s x\ (u\ v) \cdot_s y)$. We discussed extending Fresh Logic with variable swappings à la Spatial Logics in §10.2.

Spatial Logic also has $\mathsf{И}$ bind variables, thus $\mathsf{И}x$ rather than $\mathsf{И}a$. We do not necessarily understand the difference between these two choices, but perhaps there is none in the sense that in either case the datatype describing the choice of fresh atom (concretely, as in $\mathsf{И}n$, or the atom to which to evaluate the x bound in $\mathsf{И}x$) is $\mathbb{A} \otimes -$.

11.2. **Future work: Proof-terms, higher orders.** An obvious next step is to add proof-terms to Fresh Logic. There is no evident difficulty to this. It will be interesting to see what λ-calculus we get.

It would be interesting to consider higher-order versions of Fresh Logic, and especially to see how this changes (simplifies?) the notion of slice, which we needed to express the И intro- and elim-rules (ИI) and (ИE).

A higher-order Fresh Logic could form the basis of a И-version of the Isabelle/Pure meta-language, suitable for modelling syntax. Isabelle/Pure [23] is a higher-order logic with $\Rightarrow$, $\forall$, and $=$ *only*, consistent with a philosophy of 'weak meta-language, strong object-language'. The issues involved in implementing an FM object-logic in Isabelle/Pure, and why adding И would be useful though it conflicts with that philosophy, are discussed in [10].

11.3. **Self-duality.** Fresh Logic's И quantifier has proof-theoretic interest in its own right. Consider $\forall$ and $\exists$. These are dual in the sense that $\neg\forall x.\, P \Leftrightarrow \exists x.\, \neg P$. Call a quantifier ∇ **self-dual** when $\neg\nabla x.\, P \Leftrightarrow \nabla x.\, \neg P$, or more briefly $\neg\nabla \equiv \nabla\neg$. This makes the quantifier's rules symmetric.

It is an old observation that И is self-dual [12, (30)]. The FM atoms some/any property is logically composed of self-duality conjoined with (in the same shorthand) $\forall \Rightarrow$ И $\Rightarrow \exists$.

We named ∇ after Tiu and Miller (see above). To our knowledge self-duality is a new logical phenomenon. The Tiu-Miller ∇ and Gabbay-Pitts И quantifier are the only two so far discovered. (Thanks to this paper, И is now respectable and has a proof-theory!) What is the theory of an arbitrary self-dual quantifier ∇?

Menni [18] inspired by FM considers И-quantifiers in a purely category-theoretic context, taking self-duality as the quantifier's core feature. This is one answer.

11.4. **Vanilla.** Fresh Logic is a meta-language for reasoning on fresh names whose core is unmodified First-Order Logic. We have investigated the effects of this extension of First-Order Logic with 'vanilla' FM on proof-theory and semantics. Perhaps this could serve as a basis for modifications in application-specific directions, presumably with object-level names and binding, just as we do in general with First-Order Logic.

Caires-Cardelli Spatial Logics are a first instance of this, retrospectively. We can hope this paper will be of use to them and others in the future.

References

1. John Bell and Moshé Machover, *A course in mathematical logic*, North-Holland, 1977.
2. N. Brunner, *75 years of independence proofs by fraenkel-mostowski permutation models*, 1996.
3. Anna Bucalo, Martin Hofmann, Furio Honsell, Marino Miculan, and Ivan Scagnetto, *Consistency of the theory of contexts*, (2001), Submitted.
4. Luís Caires and Luca Cardelli, *A spatial logic for concurrency (part II)*, CONCUR'2002 Proceedings, Lecture Notes in Computer Science, no. 2421, 2002.
5. ———, *A spatial logic for concurrency (part II)*, Extended version, with proofs, 2003.
6. FreshML, *FreshML homepage*, , `http://www.freshml.org`.
7. Harvey Friedman, *Equality between functionals*, Logic Colloquium 1972-73 (Rohit Parikh, ed.), Lecture Notes in Mathematics, vol. 453, Springer Verlag, 1975, pp. 22–37.
8. Murdoch. J. Gabbay, *The π-calculus in FM*, Thirty-five years of Automath (Fairouz Kamareddine, ed.), Kluwer, 2003.
9. Murdoch J. Gabbay, *A theory of inductive definitions with alpha-equivalence*, Ph.D. thesis, Cambridge, UK, 2000.
10. ———, *Automating fraenkel-mostowski syntax*, TPHOLs, 15th International Conference on Theorem Proving in Higher Order Logics, August 2002.
11. Murdoch J. Gabbay and A. M. Pitts, *A new approach to abstract syntax involving binders*, 14th Annual Symposium on Logic in Computer Science, IEEE Computer Society Press, Washington, 1999, pp. 214–224.

12. Murdoch J. Gabbay and A. M. Pitts, *A new approach to abstract syntax with variable binding*, Formal Aspects of Computing **13** (2001), 341–363.
13. Furio Honsell, Marino Miculan, and Ivan Scagnetto, *An axiomatic approach to metareasoning on systems in HOAS*, Proceedings of ICALP'O1, vol. 2076, LNCS, 2001, pp. 963–978.
14. ———, *Theory of contexts for first-order and higher-order abstract syntax*, ENTCS, vol. 62, Elsevier/Forum, 2001.
15. B. Huppert, *Endliche gruppen i*, Grundlehren der mathematischen Wissenschaften **134** (1967), 363–397.
16. J. L. Krivine, *Lambda-calculus, types and models*, Ellis Horwood, 1993.
17. Daniel Leivant, *Higher order logic*, Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 2, 1994, pp. 229–322.
18. Matias Menni, *About Ν-quantifiers*, Applied categorical structures (2003), Submitted.
19. Marino Miculan and Ivan Scagnetto, *A framework for typed HOAS and semantics*, Proceedings of PPDP 2003, ACM, 2003, Uppsala.
20. Dale Miller and Alwen Tiu, *A proof theory for generic judgments: An extended abstract*, Accepted for LICS03. To appear, July 2003.
21. John C. Mitchell, *Foundations of Programming Languages*, MIT Press, 1996.
22. John C. Mitchell and Eugenio Moggi, *Kripke-style models for typed lambda calculus*, Proceedings, Symposium on Logic in Computer Science, The Computer Society of the IEEE, 1996, pp. 303–314.
23. Lawrence C. Paulson, *The foundation of a generic theorem prover*, Journal of Automated Reasoning **5** (1989), no. 3, 363–397.
24. Frank Pfenning and Conal Elliot, *Higher-order abstract syntax*, SIGPLAN Conference on Programming Language Design and Implementation, 1988, pp. 199–208.
25. A. M. Pitts, *Nominal logic: A first order theory of names and binding*, Theoretical Aspects of Computer Software, 4th International Symposium, TACS 2001, Sendai, Japan, October 29-31, 2001, Proceedings (N. Kobayashi and B. C. Pierce, eds.), Lecture Notes in Computer Science, vol. 2215, Springer-Verlag, Berlin, 2001, pp. 219–242.
26. ———, *Nominal logic, a first order theory of names and binding*, Information and Computation **186** (2003), 165–193.
27. R. Statman, *Equality between functionals, revisited*, Harvey Friedman's Research on the Foundations of Mathematics (Harrington, Morley, Scedrov, and Simpson, eds.), Lecture Notes in Mathematics, North-Holland, 1985, pp. 331– 338.
28. C. Urban, A. M. Pitts, and Murdoch J. Gabbay, *Nominal unification*, Computer Science Logic and 8th Kurt Gödel Colloquium (CSL'03 & KGC), Vienna, Austria. Proccedings (M. Baaz, ed.), Lecture Notes in Computer Science, vol. 2803, 2003, pp. 513–527.
29. Dirk van Dalen, *Intuitionistic Logic*. In Vol. III, *Handbook of philosophical logic*, Dov Gabbay and Franz Günthner eds., Synthese Library, no. 166, D.Reidel Publishing company, 1986.

Appendix A. Formal Inductive Definitions

$FV(P)$ and $FV(t)$ are the variables occurring (free) in P and t respectively:

$$\begin{array}{llll} FV(c(ts)) & =FV(ts) & FV(\pi\cdot x) & =\{x\} \\ FV(p(ts)) & =FV(ts) & FV(P\wedge P') & =FV(P)\cup FV(P') \\ \ldots & & FV(\forall x.\,P) & =FV(P)\setminus\{x\} \\ FV(\text{Ν}a.\,P) & =FV(P). & & \end{array} \tag{38}$$

Here ts is a list of terms $(t_1,\ldots,t_k)$. $FV(ts)$ is the set union $\bigcup_i FV(t_i)$. We shall use this kind of shorthand a lot.

$\mathrm{A}(P)$, $\mathrm{A}(t)$, and $\mathrm{A}(\pi)$ are the atoms occurring (free) in P, t, and π respectively:

$$\begin{array}{llll} \mathrm{A}((a\ b)\circ\pi) & =\{a,b\}\cup\mathrm{A}(\pi) & \mathrm{A}(\mathbf{Id}) & =\emptyset \\ \mathrm{A}(c(ts)) & =\mathrm{A}(ts) & \mathrm{A}(\pi\cdot x) & =\mathrm{A}(\pi) \\ \mathrm{A}(p(ts)) & =\mathrm{A}(ts) & \mathrm{A}(P\wedge P') & =\mathrm{A}(P)\cup\mathrm{A}(P') \\ \ldots & & \mathrm{A}(\forall x.\,P) & =\mathrm{A}(P) \\ \mathrm{A}(\text{Ν}a.\,P) & =\mathrm{A}(P)\setminus\{a\}. & & \end{array} \tag{39}$$

Term-for-variable substitution $t\{x{\mapsto}s\}$, where s must have the sort of x:

$$\begin{array}{llll} a\{x{\mapsto}s\} & = a & (\pi\cdot x)\{y{\mapsto}s\} & = \pi\cdot x \quad x\not\equiv y \\ (\pi\cdot x)\{x{\mapsto}s\} & = \pi\cdot_s s & \mathsf{c}(ts)\{x{\mapsto}s\} & = \mathsf{c}(ts\{a{\mapsto}s\}), \end{array} \tag{40}$$

Here $\pi\cdot_s s$ is defined in (42) below. Term-for-variable substitution on predicates:

$$\begin{array}{llll} p(ts)\{x{\mapsto}t\} & = p(ts\{x{\mapsto}t\}) & (P\wedge Q)\{x{\mapsto}t\} & = P\{x{\mapsto}t\}\wedge Q\{x{\mapsto}t\} \\ & \ldots & (\forall x'.\,P)\{x{\mapsto}t\} & = \forall x'.\,(P\{x{\mapsto}t\}) \\ (\exists x'.\,P)\{x{\mapsto}t\} & = \exists x'.\,(P\{x{\mapsto}t\}) & (\mathsf{И}n.\,P)\{x{\mapsto}t\} & = \mathsf{И}n.\,(P\{x{\mapsto}t\}). \end{array} \tag{41}$$

In the clauses for $\forall$ and $\exists$ we assume $x'\not\in FV(t)$. In the clause for $\mathsf{И}$ we assume $n\not\in \mathrm{A}(t)$.

The action of permutations π on terms t is given by:

$$\begin{array}{llll} \pi\cdot_s a & = \pi(a) & \pi\cdot_s(\kappa\cdot x) & = \pi\circ\kappa\cdot x \\ \pi\cdot_s \mathsf{c}(ts) & = \mathsf{c}(\pi\cdot_s ts). & & \end{array} \tag{42}$$

Here $\pi\cdot_s ts$ is the element-wise application to the list of terms ts. (42) reflects the semantic assumption made in §4.1 that $[\![\mathsf{c}]\!]$ be equivariant.

The action of permutations π on predicates is given by:

$$\begin{array}{llll} \pi\cdot_s p(ts) & = p(\pi\cdot_s ts) & \pi\cdot_s(P\wedge Q) & = \pi\cdot_s P\wedge\pi\cdot_s Q \\ & \ldots & \pi\cdot_s\forall x.\,P & = \forall x.[\![\pi\cdot_s P\{x{\mapsto}\pi^{-1}\cdot x\}]\!]_x \\ \pi\cdot_s\exists x.\,P & = \exists x.[\![\pi\cdot_s P\{x{\mapsto}\pi^{-1}\cdot x\}]\!]_x & \pi\cdot_s\mathsf{И}n.\,P & = \mathsf{И}n.\,\pi\cdot_s P. \end{array} \tag{43}$$

In this last clause we take n fresh, so $n\not\in\mathrm{A}(\pi)$. The (rather ad-hoc) notation $[\![\cdot]\!]_x$ denotes the following operation: for every moderation $\pi\circ\kappa\circ\pi^{-1}\cdot x$ on x we replace it by $\kappa^\pi\cdot x$, where κ^π denotes the permutation obtained by applying π as a permutation to the elements of κ (conjugating κ as syntax with π as semantics). Since $ds(\pi\circ\kappa\circ\pi^{-1},\kappa^\pi)=\emptyset$ and in view of (πdiff), this is reasonable. It is needed in §8.4.

Appendix B. Judgements (sequent style)

... are presented in Figure 6, Figure 7, and Figure 8.

M.J.Gabbay, jamie@dcs.kcl.ac.uk, Department of Computer Science, King's College, London

(Axiom) $$\frac{}{\Gamma, P \vdash P}$$

(∧RI) $$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}$$

(∧LI) $$\frac{\Gamma, P, Q \vdash C}{\Gamma, P \wedge Q \vdash C}$$

(∨RI1) $$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q}$$

(∨RI2) $$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q}$$

(∨LI) $$\frac{\Gamma, P \vdash C \quad \Gamma, Q \vdash C}{\Gamma, P \vee Q \vdash C}$$

(⇒RI) $$\frac{\Gamma, P \vdash Q}{\Gamma \vdash P \Rightarrow Q}$$

(⇒LI) $$\frac{\Gamma \vdash P \quad \Gamma, Q \vdash C}{\Gamma, P \Rightarrow Q \vdash C}$$

(Cut) $$\frac{\Gamma \vdash P \quad \Gamma, P \vdash Q}{\Gamma \vdash Q}$$

(⊥LI) $$\frac{}{\Gamma, \bot \vdash C}$$

(⊤RI) $$\frac{}{\Gamma \vdash \top}$$

(∀RI) $$\frac{\Gamma \vdash P}{\Gamma \vdash \forall x.\, P} \quad x \notin FV(\Gamma)$$

(∀LI) $$\frac{\Gamma, P\{x \mapsto t\} \vdash C}{\Gamma, \forall x.\, P \vdash C} \quad x \notin FV(\Gamma, C)$$

(∃RI) $$\frac{\Gamma \vdash P\{x \mapsto t\}}{\Gamma \vdash \exists x.\, P} \quad x \notin FV(\Gamma)$$

(∃LI) $$\frac{\Gamma, P \vdash C}{\Gamma, \exists x.\, P \vdash C} \quad x \notin FV(\Gamma, C)$$

(Contraction) $$\frac{\Gamma, P, P \vdash C}{\Gamma, P \vdash C}$$

FIGURE 6. Core (standard) sequent rules of Fresh Logic, sequent style

$$(\pi\mathrm{I})\quad \frac{\Gamma \vdash P}{\Gamma \vdash \pi \cdot_s P}$$

$$(\pi\mathrm{diff})\quad \frac{\Gamma \vdash P\{x{\mapsto}\pi' \cdot_s t\}}{\Gamma, ds(\pi,\pi')\#t \vdash P\{x{\mapsto}\pi \cdot_s t\}} \qquad x \notin FV(t)$$

$$(\text{И}\mathrm{RI})\quad \frac{\Gamma \vdash P\{n{\mapsto}a\}}{\Gamma, a\#t_i \vdash \text{И}n.\,P} \qquad P/n = P' \bullet_{y_i} (t_i)_1^k \quad a \notin \mathrm{A}(P')$$

$$(\text{И}\mathrm{LI})\quad \frac{\Gamma, P\{n{\mapsto}a\} \vdash C}{\Gamma, \text{И}n.\,P, a\#t_i \vdash C} \qquad P/n = P' \bullet_{y_i} (t_i)_1^k \quad a \notin \mathrm{A}(P')$$

FIGURE 7. π and И sequent rules of Fresh Logic

$$(\mathrm{EqRI})\quad \frac{}{\Gamma \vdash t = t}$$

$$(\mathrm{EqLI1})\quad \frac{\Gamma\{x{\mapsto}t'\} \vdash P\{x{\mapsto}t'\}}{\Gamma\{x{\mapsto}t\}, t' = t \vdash P\{x{\mapsto}t\}} \qquad x \notin FV(t,t')$$

$$(\#\mathrm{RI})\quad \frac{}{\Gamma \vdash a\#b}$$

$$(\#\mathrm{LI})\quad \frac{}{\Gamma, a\#a \vdash C}$$

$$(\#\mathsf{c})\quad \frac{\Gamma \vdash a\#ts}{\Gamma \vdash a\#\mathsf{c}(ts)}$$

$$(\mathrm{New}\mathbb{A})\quad \frac{\Gamma, a\#ts \vdash C}{\Gamma \vdash C} \qquad a \notin \mathrm{A}(\Gamma, C)$$

$$(\mathrm{Exhaust}\mathbb{A})\quad \frac{\bigwedge n \in S.\,\Gamma\{x{\mapsto}n\} \vdash P\{x{\mapsto}n\}}{\Gamma\{x{\mapsto}t\} \vdash P\{x{\mapsto}t\}} \quad \mathrm{A}(\Gamma,P) \subsetneq S$$

FIGURE 8. $=$, $\#$, and $\mathbb{A}$ sequent rules of Fresh Logic