Nominal semantics for predicate logic: algebras, substitution, quantifiers, and limits

Gilles Dowek and Murdoch J. Gabbay

Abstract. We define a model of predicate logic in which every term and predicate, open or closed, has an absolute denotation independently of a valuation of the variables. For each variable a, the domain of the model contains an element $[\![a]\!]$ which is the denotation of the term a (which is also a variable symbol). Similarly, the algebra interpreting predicates in the model directly interprets open predicates. Because of this models must also incorporate notions of substitution and quantification.
These notions are axiomatic, and need not be applied only to sets of syntax. We prove soundness and show how every 'ordinary' model (i.e. model based on sets and valuations) can be translated to one of our nominal models, and thus also prove completeness.

Keywords: Lattices and algebra; First-order logic; Nominal semantics

1 Introduction

What is a notion of algebraic truth suitable for interpreting predicate logic?

For *propositional* logic an answer is clear: Heyting algebra for the intuitionistic case, and Boolean algebra for the classical case.[1] This is not sufficient for interpreting predicate logic because predicates can contain variable symbols.

The standard method, due to Tarski [25], uses valuations. We assume a domain D and a valuation ς mapping variables to D, and then terms and predicates are given denotation in the context of the valuation: $[\![r]\!]_\varsigma$ and $[\![\phi]\!]_\varsigma$. Intuitively, valuations convert predicates to propositions, which reduces the problem of truth to the infinite propositional case.

Thus with the usual notion of model, names denote semantic objects, but semantic objects have no inherent notion of names.

We propose an alternative. Allow semantic objects to be aware of names. Then, we can map a variable *a to a representation of itself* in the denotation. Thus, open terms and open predicates map to open elements of a domain and algebra of truth-values, like so: $[\![r]\!]$ and $[\![\phi]\!]$. Connectives, including quantifiers, just become operations on the domain. So, the denotation of a is $[\![a]\!]$ and the denotation of $\forall a.\phi$ is some operation $\bigwedge^{\#a}$ applied to $[\![\phi]\!]$.

What it means for an element to be open is given for us by the recently-developed theory of *nominal semantics*, which was designed to provide semantics for variable symbols in inductive syntax-with-binding [20,12]. In this paper

[1] Accessible references for Heyting and Boolean algebra are easily found online. For a modern and encyclopaedic treatment, see e.g. [28].

we apply nominal ideas to semantic objects which, unlike the syntax of terms and predicates, need not be inductively defined. We use *nominal algebra* to axiomatise substitution following [17,18] and a simple theory of *$A\#$fresh limits* in partial orders, which is introduced in this paper. Using $A\#$fresh limits we show how $\top$, $\wedge$, and the quantifier $\bigwedge^{\#a}$ are all just greatest $A\#$lower fresh bounds for suitable finite sets. Specifically:

- $\top$ is the greatest lower bound for the empty set $\varnothing$.
- $x \wedge y$ is the greatest lower bound for the set $\{x, y\}$, i.e. $\bigwedge\{x, y\}$.
- $\bigwedge^{\#a}x$ is the greatest $\{a\}\#$lower bound for the set $\{x\}$, i.e. $\bigwedge^{\#\{a\}}\{x\}$.

We use this to construct a nominal algebraic semantics for first-order logic and prove it sound and complete. The completeness proof is a little unorthodox: instead of a direct proof (which is not hard) we give a more general lifting construction from Tarski-style valuation models, to our nominal models. This gives a direct translation of the familiar semantics into our new one and we obtain completeness as a corollary.

Map of the paper Section 2 presents the technical background: nominal sets and first-order logic. Section 3 introduces nominal posets and substitution algebras. This material is partly new (nominal posets and $\bigwedge^{\#A}$) and partly a restatement of [17,18] (substitution algebras; though to be fair, the presentation here is significantly simplified and updated). Section 4 introduces nominal Boolean algebras, which is a mathematically precise answer to the question with which we opened this Introduction, and proves soundness. Finally, Section 5 presents the 'lifting' operation on models and proves completeness.

2 Technical background

2.1 Background on nominal sets

Nominal sets were introduced in [20] (where they were called *equivariant Fraenkel-Mostowski sets*). The presentation below is quite self-contained; see [20] or [12] for more details.

We want names to 'exist' in the denotation. We do this by endowing sets with a symmetry group action over name permutations (Definition 2.3). We extract from this the key idea of *support* (Definition 2.4), which captures the degree of asymmetry of an element under the group action. We use this to define a *nominal set* X (Definition 2.6) as a set with a group action all of whose elements x are at most finitely asymmetric (have finite support), where $a\#x$ and $a \notin supp(x)$ mean the same thing: "x is symmetric over/does not depend on the name a". We do not *a priori* assume e.g. a substitution action on names; in our framework this is algebraically definable just from the more primitive notion of permutable names and finite asymmetry (see e.g. Figure 2).

Definition 2.1. Fix a countably infinite set of **atoms** $\mathbb{A}$. We use a **permutative convention** that $a, b, c, \ldots$ range over *distinct* atoms.

Definition 2.2. A **(finite) permutation** π is a bijection on the set $\mathbb{A}$ such that $\mathit{nontriv}(\pi) = \{a \mid \pi(a) \neq a\}$ is finite.

Write id for the **identity** permutation such that $\mathit{id}(a) = a$ for all a. Write $\pi' \circ \pi$ for composition, so that $(\pi' \circ \pi)(a) = \pi'(\pi(a))$. Write $\pi^{\text{-}1}$ for inverse, so that $\pi^{\text{-}1} \circ \pi = \mathit{id} = \pi \circ \pi^{\text{-}1}$. Write $(a\ b)$ for the **swapping** (terminology from [20]) mapping a to b, b to a, and all other c to themselves, and take $(a\ a) = \mathit{id}$.

Definition 2.3. A **set with a permutation action** X is a pair $(|\mathsf{X}|, \cdot)$ of an **underlying set** $|\mathsf{X}|$ and a **permutation action** written $\pi{\cdot}x$ which is a group action on $|\mathsf{X}|$, so that $\mathit{id}{\cdot}x = x$ and $\pi{\cdot}(\pi'{\cdot}x) = (\pi \circ \pi'){\cdot}x$ for all $x \in |\mathsf{X}|$ and permutations π and π'.

Definition 2.4. Say that $A \subseteq \mathbb{A}$ **supports** $x \in |\mathsf{X}|$ when for every π, if $\pi(a) = a$ for every $a \in A$ then $\pi{\cdot}x = x$. If some finite A supporting x exists, then call x **finitely-supported**. Then define the **support** of x by

$$\mathit{supp}(x) = \bigcap\{A \mid A \text{ finite},\ A \text{ supports } x\}.$$

Write $a\#x$ as shorthand for $a \notin \mathit{supp}(x)$ and read this as a is **fresh for** x.

Theorem 2.5. *Suppose* X *is a set with a permutation action and* $x \in |\mathsf{X}|$*. Then if* x *has finite support then* $\mathit{supp}(x)$ *supports* x *and is the unique least finite supporting set of* x*.*

Proof. See [12, Theorem 2.21]. A different but equivalent formulation of the result is in [20].

Definition 2.6. Suppose X is a set with a permutation action.
Call a set with a permutation action X a **nominal set** when every $x \in |\mathsf{X}|$ has finite support. X, Y, Z will range over nominal sets.

Example 2.7.

- $\mathbb{A}$ is a nominal set where $\pi{\cdot}a = \pi(a)$. It is easy to check that $\mathit{supp}(a) = \{a\}$.
- The sets of terms and predicates from Definition 2.12 below, are nominal sets if we let permutation act in the natural way on the atoms (considered as variable symbols) inside them. It is easy to check that $\mathit{supp}(r)$ or $\mathit{supp}(\phi)$ is equal to the atoms occurring in r or ϕ if we *do not* take terms up to α-equivalence, and $\mathit{supp}(r)$ or $\mathit{supp}(\phi)$ is equal to the atoms occurring *free* in r or ϕ if we *do* take r or ϕ up to α-equivalence. A detailed study of this is in [12, Section 5]; see also [4].

Later on in this paper we shall see more examples. Notably, the sets $X^{\bullet}$ from Definition 5.9 are nominal sets, and the set of *valuations* from Definition 5.3 is a set with a permutation action but not a nominal set (see Remark 5.8 and associated footnote). For more examples see [12].

Definition 2.8. Call a function $f \in |\mathsf{X}| \to |\mathsf{Y}|$ **equivariant** when $\pi{\cdot}f(x) = f(\pi{\cdot}x)$ for all permutations π and $x \in |\mathsf{X}|$. In this case write $f : \mathsf{X} \to \mathsf{Y}$.

Proposition 2.9. $\mathit{supp}(\pi{\cdot}x) = \{\pi(a) \mid a \in \mathit{supp}(x)\}$.

Proof. It is not hard to check that A supports x if and only if $\{\pi(a) \mid a \in A\}$ supports $\pi{\cdot}x$. We use Theorem 2.5.

Corollary 2.10. *1. If $\pi(a) = a$ for all $a \in supp(x)$ then $\pi{\cdot}x = x$.*
2. If $\pi(a) = \pi'(a)$ for every $a \in supp(x)$ then $\pi{\cdot}x = \pi'{\cdot}x$.
3. $a\#x$ if and only if $\exists b.b\#x \wedge (b\ a){\cdot}x = x$.

Proof. Parts 1 and 2 follow from the fact that $supp(x)$ supports x (Theorem 2.5). Part 3 follows using Proposition 2.9.

2.2 First-order logic

First-order logic is of course standard. The reader can find any number of presentations in the literature, for instance [27].

Definition 2.11. Here and for the rest of this paper fix a signature

$$\Sigma = (\mathit{TermFormers}, \mathit{PredicateFormers}, \mathit{arity})$$

of **term-formers** and **predicate-formers** and an **arity** function arity mapping $\mathit{TermFormers} \cup \mathit{PredicateFormers}$ to $\mathbb{N} = \{0, 1, 2, \dots\}$. We let f range over distinct term-formers and P range over distinct predicate-formers.

Definition 2.12. Define **terms** r and **predicates** ϕ inductively by:

$$\boxed{\begin{array}{l} r ::= a \mid \mathsf{f}(r_1, \dots, r_{\mathit{arity}(\mathsf{f})}) \\ \phi ::= \bot \mid \mathsf{P}(r_1, \dots, r_{\mathit{arity}(\mathsf{P})}) \mid \phi \wedge \phi \mid \neg\phi \mid \forall a.\phi \end{array}}$$

Define **free atoms** inductively by:

$$\begin{array}{lll} fa(a) = \{a\} & fa(\mathsf{f}(r_1, \dots, r_n)) = \bigcup_i fa(r_i) & \\ fa(\bot) = \varnothing & fa(\mathsf{P}(r_1, \dots, r_n)) = \bigcup_i fa(r_i) & \\ fa(\neg\phi) = fa(\phi) & fa(\phi_1 \wedge \phi_2) = fa(\phi_1) \cup fa(\phi_2) & fa(\forall a.\phi) = fa(\phi) \setminus \{a\} \end{array}$$

Definition 2.13. We take predicates ϕ up to α-equivalence as usual (so for instance $\forall a.\mathsf{P}(a) = \forall b.\mathsf{P}(b)$ where $\mathit{arity}(\mathsf{P}) = 1$).

We write $r[a{:=}s]$ and $\phi[a{:=}s]$ for the usual capture-avoiding substitution on terms and predicates. For instance, $\mathsf{f}(a)[a{:=}b] = \mathsf{f}(b)$ and $(\forall b.\mathsf{P}(a))[a{:=}b] = \forall b'.\mathsf{P}(b)$.

Definition 2.14. Let Φ and Ψ range over finite sets of predicates. A **sequent** is a pair $\Phi \vdash \Psi$. We define the **derivable sequents** as usual for classical first-order logic by the rules in Figure 1.

3 Partial orders and substitution

We obtain our nominal semantics for first-order logic by combining two things: a partial order, and a substitution action. We need the partial order to interpret entailment. We need the substitution action to interpret variables and substitutions. These are Subsections 3.1 and 3.2.

Of course, these two things interact via quantifiers. Thus we introduce the novel notion of $\bigwedge^{\#A} X$, an $A\#$greatest lower bound for a set X (Notation 3.3). It turns out that this generalises $\bot$, $\wedge$, and $\bigwedge^{\#a}$ into a single definition.

$$\frac{}{\Phi,\bot\vdash\Psi}\,(\bot\mathbf{L})\qquad\frac{\phi_1,\phi_2,\Phi\vdash\Psi}{\phi_1\land\phi_2,\Phi\vdash\Psi}\,(\land\mathbf{L})\qquad\frac{\Phi\vdash\psi_1,\Psi\quad\Phi\vdash\psi_2,\Psi}{\Phi\vdash\psi_1\land\psi_2,\Psi}\,(\land\mathbf{R})$$

$$\frac{\Phi\vdash\phi,\Psi}{\Phi,\neg\phi\vdash\Psi}\,(\neg\mathbf{L})\qquad\frac{\Phi,\phi\vdash\Psi}{\Phi\vdash\neg\phi,\Psi}\,(\neg\mathbf{R})$$

$$\frac{\Phi,\phi[a{:=}s]\vdash\Psi}{\Phi,\forall a.\phi\vdash\Psi}\,(\forall\mathbf{L})\qquad\frac{\Phi\vdash\psi,\Psi\quad(a\notin fa(\Phi,\Psi))}{\Phi\vdash\forall a.\psi,\Psi}\,(\forall\mathbf{R})$$

Fig. 1: Derivability in first-order classical logic

3.1 Nominal posets (nominal partially ordered set)

We can now begin to ask what happens if we combine the notion of partially ordered set with the notion of nominal set. In particular, we can think about greatest lower and least upper bounds *in the presence* of nominal freshness. This leads us to the idea of $\bigwedge^{\#A}X$ and $\bigvee^{\#A}X$; greatest lower and least upper bounds amongst elements that do not include A in their support.

Definition 3.1. A **nominal poset** is a tuple $\mathcal{L}=(|\mathcal{L}|,\cdot,\leq)$ such that $(|\mathcal{L}|,\cdot)$ is a nominal set and $\leq\subseteq|\mathcal{L}|\times|\mathcal{L}|$ is an equivariant partial order.[2]

Call $\mathcal{L}$ **finitely fresh-complete** when for every finite subset $X\subseteq|\mathcal{L}|$ and every finite set of atoms $A\subseteq\mathbb{A}$ the set of A-fresh lower bounds

$$\{x'\in|\mathcal{L}|\mid A\cap supp(x')=\varnothing\ \land\ \forall x{\in}X.x'{\leq}x\}$$

has a $\leq$-greatest element $\bigwedge^{\#A}X$.

Similarly call $\mathcal{L}$ **finitely fresh-cocomplete** when for every finite X and A as above the set of A-fresh upper bounds $\{x'\in|\mathcal{L}|\mid A\cap supp(x')=\varnothing\ \land\ \forall x{\in}X.x{\leq}x'\}$ has a $\leq$-least element $\bigvee^{\#A}X$.

Lemma 3.2. *$\bigwedge^{\#A}X$ and $\bigvee^{\#A}X$ are unique if they exist.*

Proof. From the fact that for a partial order, $x\leq y$ and $y\leq x$ imply $x=y$.

Notation 3.3. Suppose $\mathcal{L}=(|\mathcal{L}|,\cdot,\leq)$ is a finitely fresh-complete and -cocomplete nominal poset. Suppose $X\subseteq|\mathcal{L}|$ and $A\subseteq\mathbb{A}$ are finite. Then:

- Call $\bigwedge^{\#A}X$ an $A\#$**greatest lower bound** (*'A-fresh greatest lower bound'*) of X. If $A=\varnothing$ then we may just call this a greatest lower bound. If $A=\{a\}$ we may just call this an $a\#$greatest lower bound and write it $\bigwedge^{\#a}X$.
- Call $\bigvee^{\#A}X$ an $A\#$**least upper bound** of X. If $A=\varnothing$ then we may just call this a least upper bound. If $A=\{a\}$ we may just call this an $a\#$least upper bound and write it $\bigvee^{\#a}X$.

Notation 3.4. Suppose $\mathcal{L}=(|\mathcal{L}|,\cdot,\leq)$ is a nominal poset.

- Write $\top$ for the greatest lower bound and $\bot$ for the least upper bound of the empty set $\varnothing$, where these exist.

[2] So $x\leq y$ if and only if $\pi{\cdot}x\leq\pi{\cdot}y$.

(**Suba**)		$a[a{\mapsto}u] = u$
(**Subid**)		$x[a{\mapsto}a] = x$
(**Sub#**)	$a\#x \Rightarrow$	$x[a{\mapsto}u] = x$
(**Subα**)	$b\#x \Rightarrow$	$x[a{\mapsto}u] = ((b\ a)\cdot x)[b{\mapsto}u]$
(**Subσ**)	$a\#v \Rightarrow$	$x[a{\mapsto}u][b{\mapsto}v] = x[b{\mapsto}v][a{\mapsto}u[b{\mapsto}v]]$

Fig. 2: Nominal algebra axioms for substitution action

- Write $x \wedge y$ for the greatest lower bound and $x \vee y$ for the least upper bound of $\{x, y\}$, where these exist.
- Write $\bigwedge^{\#a} x$ for the $a\#$greatest lower bound and $\bigvee^{\#a} x$ for the $a\#$least upper bound of $\{x\}$, where these exist.

Definition 3.5. Suppose $\mathcal{L}$ is a partial order (it does not matter whether it is nominal). Call $x' \in |\mathcal{L}|$ a **complement** of $x \in |\mathcal{L}|$ when $x \wedge x' = \bot$ and $x \vee x' = \top$. If every $x \in |\mathcal{L}|$ has a complement say that $\mathcal{L}$ **has complements**, and write the complement of x as $\neg x$.

Lemma 3.6. *Complements are unique if they exist.*

Corollary 3.7. *Suppose $\mathcal{L} = (|\mathcal{L}|, \cdot, \leq)$ is a nominal poset.*

1. *Suppose $X \subseteq |\mathcal{L}|$ is finite and $A \subseteq \mathbb{A}$, and $\bigwedge^{\#A} X$ exists. Then $supp(\bigwedge^{\#A} X) \subseteq \bigcup\{supp(x) \mid x \in X\} \setminus A$.*
2. *Suppose $x \in |\mathcal{L}|$, and suppose $\neg x$ exists. Then $supp(\neg x) = supp(x)$.*

Proof. We can use symmetry (equivariance) properties of atoms:

1. Using [12, Theorems 2.29 and 4.7] $supp(\bigwedge^{\#A} X) \subseteq \bigcup\{supp(x) \mid x \in X\} \cup A$.[3] Since by assumption $A \cap supp(\bigwedge^{\#A} X) = \varnothing$, the result follows.
2. By [12, Theorem 4.7] and since the map $x \mapsto \neg x$ is injective.

3.2 Substitution algebra

The reader may be used to seeing substitution as a concrete operation on sets. However, using nominal sets we can powerfully (and axiomatically) generalise substitution to be an abstract (nominal) algebraic structure.

Definition 3.8. A **termlike substitution algebra** over Σ is a tuple $\mathsf{U} = (|\mathsf{U}|, \cdot, atm, sub)$ where:

- $(|\mathsf{U}|, \cdot)$ is a nominal set.
- $atm : \mathbb{A} \to |\mathsf{U}|$ is an equivariant injection, usually written invisibly (so we write $atm(a)$ just as a).
- $sub : |\mathsf{U}| \times \mathbb{A} \times |\mathsf{U}| \to |\mathsf{U}|$ is an equivariant **substitution action**, written infix $v[a{\mapsto}u]$.

such that the equalities (**Suba**) to (**Subσ**) of Figure 2 hold, where x, u, and v range over elements of $|\mathsf{U}|$.

[3] This is just a fancy way of observing that since the definition of $\bigwedge$ is symmetric in atoms, the the result must be at least as symmetric as the inputs.

Definition 3.9. Suppose $\mathsf{U} = (|\mathsf{U}|, \cdot, sub, atm)$ is a termlike substitution algebra. A **substitution algebra** over U is a tuple $\mathsf{B} = (|\mathsf{B}|, \cdot, sub)$ where:

- $(|\mathsf{B}|, \cdot)$ is a nonempty nominal set.
- $sub : |\mathsf{B}| \times \mathbb{A} \times |\mathsf{U}| \to \mathsf{B}$ is an equivariant **substitution** function satisfying the axioms (**Subid**) to (**Subσ**) of Figure 2, where x ranges over elements of $|\mathsf{B}|$ and u and v range over elements of $|\mathsf{U}|$.

Example 3.10.

1. The set of atoms $\mathbb{A}$ is a termlike substitution algebra where $atm(a) = a$ and $a[a{\mapsto}x] = x$ and $b[a{\mapsto}x] = b$.
2. Terms from Definition 2.12 are a termlike substitution algebra with $atm(a) = a$ and $r[a{\mapsto}s]$ equal to $r[a{:=}s]$.
3. Predicates from Definition 2.12 are not a termlike substitution algebra, because there are no predicate variables or substitution for predicates. Predicates are however a substitution algebra over terms.

In Corollary 5.13 we will prove that $|\mathcal{N}|^{\bullet}$ from Definition 5.11 is a termlike substitution algebra, and $X^{\bullet}$ from Definition 5.11 is a substitution algebra. For different and non-trivial classes of substitution algebras see [11,13].

4 The model

4.1 Nominal Boolean algebra

Definition 4.1 is the culminating definition of this paper; we define a notion of Boolean algebraic structure suitable for giving an 'absolute' interpretation of first-order logic. This means that variables, substitution, and quantification must be part of the model just as much as conjunction and negation are part of ordinary Boolean algebras. With our definitions so far, this is now not difficult.

Definition 4.1 is not quite enough to interpret first-order logic on its own. For that, we need the notion of model which we develop next in Subsection 4.2.

Definition 4.1. Suppose U is a termlike substitution algebra. A **nominal Boolean algebra** over U is a tuple $\mathsf{B} = (|\mathsf{B}|, \cdot, \leq, sub)$ such that:

- $(|\mathsf{B}|, \cdot, \leq)$ is a finitely fresh-complete nominal poset with complements.
- $(|\mathsf{B}|, \cdot, sub)$ is a substitution algebra over U.
- The partial order is *compatible* with the substitution action:

(**Compat** $\bigwedge$) $\quad (\bigwedge^{\#A} X)[a{\mapsto}u] = \bigwedge^{\#A}\{x[a{\mapsto}u] \mid x{\in}X\} \quad$ if $A{\cap}supp(u){=}\varnothing,\ a{\notin}A$

(**Compat**$\neg$) $\quad (\neg x)[a{\mapsto}u] = \neg(x[a{\mapsto}u])$.

A more accurate name for 'nominal Boolean algebra' might be 'nominal $\sigma\bigwedge^{\#A}\neg$ partial order'. But that is a bit of a mouthful.[4] It is quite easy to derive some basic properties of nominal Boolean algebras, which are useful for Theorem 4.13.

[4] A possibly nice classification is to define a notion of *bounded nominal lattice* which is equipped with $\wedge$, $\vee$, $\bot$, $\top$—and also with $\bigwedge^{\#a}x$. Then our nominal Boolean algebra is just a nominal lattice with complements and a compatible substitution action.

Lemma 4.2. *If $b\#u$ then $(\bigwedge^{\#b}x)[a\mapsto u] = \bigwedge^{\#b}(x[a\mapsto u])$.*

Proof. This is just (**Compat** $\bigwedge$) where $A = \{b\}$. Recall that by our permutative convention $a \notin \{b\}$.

Lemma 4.3. *$x \leq y$ if and only if $x \wedge y = x$.*

Proof. Both hold if and only if x is a greatest lower bound for $\{x, y\}$.

Lemma 4.4. *1. $(x \wedge y)[a\mapsto u] = x[a\mapsto u] \wedge (y[a\mapsto u])$.*
2. $\bot[a\mapsto u] = \bot$.

Proof. From (**Compat** $\bigwedge$) and (**Compat**$\neg$) respectively.

Corollary 4.5. *If $x \leq y$ then $x[a\mapsto u] \leq y[a\mapsto u]$. As a further corollary, if $x \leq y$ and $a\#x$ then $x \leq y[a\mapsto u]$ for every $u \in |\mathsf{U}|$.*

Proof. This follows from Lemma 4.3 and part 1 of Lemma 4.4. The further corollary follows using (**Sub**$\#$) from Figure 2.

Lemma 4.6. *Suppose $x \in |\mathsf{B}|$ and $u \in |\mathsf{U}|$ and $a \in \mathbb{A}$. Then $\bigwedge^{\#a}x \leq x[a\mapsto u]$.*

Proof. By Notation 3.4 $\bigwedge^{\#a}x$ is the $a\#$greatest lower bound for $\{x\}$. This means that $\bigwedge^{\#a}x \leq x$ and $a\#\bigwedge^{\#a}x$. We use Corollary 4.5.

Lemma 4.7. *Suppose $x, y{\in}|\mathsf{B}|$ and $a{\in}\mathbb{A}$. Then $x \leq y$ and $a\#x$ imply $x \leq \bigwedge^{\#a}y$.*

Proof. This is the $a\#$greatest lower bound property of $\bigwedge^{\#a}y$ with respect to $\{y\}$.

Corollary 4.8. *$\bigwedge^{\#a}x$ is the greatest lower bound of $\{x[a\mapsto u] \mid u \in |\mathsf{U}|\}$ in B.*

Proof. By Lemma 4.6 $\bigwedge^{\#a}x$ is a lower bound. Now suppose $z \leq x[a\mapsto u]$ for every $u \in |\mathsf{U}|$. In particular then, $z \leq x[a\mapsto a] \overset{\textbf{(Subid)}}{=} x$. By Lemma 4.7 $z \leq \bigwedge^{\#a}x$.

4.2 Soundness

Definition 4.9. A **model** $\mathcal{M} = (\mathsf{U}, \mathsf{B}, \text{-}^{\mathcal{M}})$ consists of:

- A termlike substitution algebra U (Definition 3.8).
- A nominal Boolean algebra B over U (Definition 4.1).
- An assignment $\text{-}^{\mathcal{M}}$ to each term-former f of an equivariant function $\mathsf{f}^{\mathcal{M}} : |\mathsf{U}|^{arity(\mathsf{f})} \to |\mathsf{U}|$ and to each predicate-former P of an equivariant function $\mathsf{P}^{\mathcal{M}} : |\mathsf{U}|^{arity(\mathsf{P})} \to |\mathsf{B}|$ such that

(**Commf**) $\quad \mathsf{f}^{\mathcal{M}}(u_1, \ldots, u_{arity(\mathsf{f})})[a\mapsto u] = \mathsf{f}^{\mathcal{M}}(u_1[a\mapsto u], \ldots, u_{arity(\mathsf{f})}[a\mapsto u])$
(**CommP**) $\quad \mathsf{P}^{\mathcal{M}}(u_1, \ldots, u_{arity(\mathsf{f})})[a\mapsto u] = \mathsf{P}^{\mathcal{M}}(u_1[a\mapsto u], \ldots, u_{arity(\mathsf{f})}[a\mapsto u])$.

For the rest of this section fix a model $\mathcal{M} = (\mathsf{U}, \mathsf{B}, \text{-}^{\mathcal{M}})$.

Definition 4.10. Define an **interpretation** function $\llbracket \text{-} \rrbracket^{\mathcal{M}}$ mapping terms and predicates (Definition 2.12) to elements of $|\mathsf{U}|$ and $|\mathsf{B}|$ respectively, as follows:

$$\begin{array}{rclcrcl}
\llbracket a \rrbracket^{\mathcal{M}} &=& atm(a) & \qquad & \llbracket \mathsf{f}(r_1,\ldots,r_n) \rrbracket^{\mathcal{M}} &=& \mathsf{f}^{\mathcal{M}}(\llbracket r_1 \rrbracket^{\mathcal{M}},\ldots,\llbracket r_n \rrbracket^{\mathcal{M}}) \\
\llbracket \bot \rrbracket^{\mathcal{M}} &=& \bot & & \llbracket \mathsf{P}(r_1,\ldots,r_n) \rrbracket^{\mathcal{M}} &=& \mathsf{P}^{\mathcal{M}}(\llbracket r_1 \rrbracket^{\mathcal{M}},\ldots,\llbracket r_n \rrbracket^{\mathcal{M}}) \\
\llbracket \phi' \wedge \phi \rrbracket^{\mathcal{M}} &=& \llbracket \phi' \rrbracket^{\mathcal{M}} \wedge \llbracket \phi \rrbracket^{\mathcal{M}} & & \llbracket \forall a.\phi \rrbracket^{\mathcal{M}} &=& \bigwedge^{\#a} \llbracket \phi \rrbracket^{\mathcal{M}} \\
\llbracket \neg \phi \rrbracket^{\mathcal{M}} &=& \neg \llbracket \phi \rrbracket^{\mathcal{M}} & & & &
\end{array}$$

Definition 4.11. Write $\mathcal{M} \vDash \phi$ when $\llbracket \phi \rrbracket^{\mathcal{M}} = \top$ and call ϕ **valid** in $\mathcal{M}$.

Proposition 4.12. *$\llbracket r[a{:=}s] \rrbracket^{\mathcal{M}} = \llbracket r \rrbracket^{\mathcal{M}}[a{\mapsto}\llbracket s \rrbracket^{\mathcal{M}}]$ and $\llbracket \phi[a{:=}s] \rrbracket^{\mathcal{M}} = \llbracket \phi \rrbracket^{\mathcal{M}}[a{\mapsto}\llbracket s \rrbracket^{\mathcal{M}}]$.*

Proof. By routine inductions on r and ϕ:

- *The case of a.* $\llbracket a[a{:=}s] \rrbracket^{\mathcal{M}} = \llbracket s \rrbracket^{\mathcal{M}}$ and by (**Subid**) $\llbracket a \rrbracket^{\mathcal{M}}[a{\mapsto}\llbracket s \rrbracket^{\mathcal{M}}] = \llbracket s \rrbracket^{\mathcal{M}}$.
- *The case of $\mathsf{f}(r_1,\ldots,r_n)$.* Using (**Commf**).
- *The case of $\phi' \wedge \phi$.* Using part 1 of Lemma 4.4.
- *The case of $\forall a.\phi$.* Using Lemma 4.2.

Theorem 4.13. *If $\Phi \vdash \Psi$ then $\bigwedge_{\phi \in \Phi} \llbracket \phi \rrbracket^{\mathcal{M}} \leq \bigvee_{\psi \in \Psi} \llbracket \psi \rrbracket^{\mathcal{M}}$.*

Proof.
- Soundness of ($\wedge$**L**) and ($\wedge$**R**) follows from $\llbracket \phi_1 \rrbracket^{\mathcal{M}} \wedge \llbracket \phi_2 \rrbracket^{\mathcal{M}}$ being a greatest $\leq$-lower bound of $\llbracket \phi_1 \rrbracket^{\mathcal{M}}$ and $\llbracket \phi_2 \rrbracket^{\mathcal{M}}$.
- Soundness of ($\neg$**L**) and ($\neg$**R**) follows from the fact of complements that $\neg x \leq y$ if and only if $\neg y \leq x$.
- Soundness of ($\bot$**L**) follows from $\bot$ being the complement of the greatest lower bound of $\varnothing$.
- Soundness of ($\forall$**L**) follows using Lemma 4.6 and Proposition 4.12, and soundness of ($\forall$**R**) follows from Lemma 4.7.

5 Lifting ordinary models to nominal models

We have to recall the (standard) definition of model for first-oder logic. This is Subsection 5.1. Then in Subsection 5.2 we show how to lift this to a nominal model and obtain completeness as an immediate corollary (Theorem 5.18 and Corollary 5.19).

5.1 Ordinary model of first-order logic

We briefly sketch the ordinary model of first-order classical logic, with valuations and without atoms. This model is not intended to be sophisticated; we will just need that one exists.

Notation 5.1. To avoid the confusion between sets and nominal sets, we may write *ordinary set* for the former.

Definition 5.2. An **ordinary model** $\mathcal{N} = (|\mathcal{N}|, \text{-}^{\mathcal{N}})$ is a tuple such that:

- $|\mathcal{N}|$ is some non-empty (ordinary) **underlying set**.

– $\text{-}^{\mathcal{N}}$ assigns to each term-former f a function $\mathsf{f}^{\mathcal{N}} : |\mathcal{N}|^{arity(\mathsf{f})} \to |\mathcal{N}|$ and to each predicate-former P a function $\mathsf{P}^{\mathcal{N}} : |\mathcal{N}|^{arity(\mathsf{P})} \to \{\bot, \top\}$.

Definition 5.3. Write $\mathbb{A}\Rightarrow|\mathcal{N}|$ for the set of functions from atoms to $|\mathcal{N}|$. Let ς range over elements of $\mathbb{A}\Rightarrow|\mathcal{N}|$ and call these **valuations** (to $\mathcal{N}$).

Also if $x \in |\mathcal{N}|$ then write $\varsigma[a{:=}x]$ for the function such that $(\varsigma[a{:=}x])(a) = x$ and $(\varsigma[a{:=}x])(b) = \varsigma(b)$.

Definition 5.4. $\{\bot, \top\}$ is a complete Boolean algebra. We use standard definitions, such as $\wedge$, $\neg$, $\bigwedge$, and $\bigvee$ without further comment

Definition 5.5. Define an **interpretation** function $[\![\text{-}]\!]^{\mathcal{N}}_{\varsigma}$ mapping terms and predicates (Definition 2.12) to elements of $|\mathcal{N}|$ and $\{\bot, \top\}$ (considered as a Boolean algebra) respectively, as follows:

$$\begin{array}{rlrl} [\![a]\!]^{\mathcal{N}}_{\varsigma} &= \varsigma(a) & [\![\mathsf{f}(r_1,\dots,r_n)]\!]^{\mathcal{N}}_{\varsigma} &= \mathsf{f}^{\mathcal{N}}([\![r_1]\!]^{\mathcal{N}}_{\varsigma},\dots,[\![r_n]\!]^{\mathcal{N}}_{\varsigma}) \\ [\![\bot]\!]^{\mathcal{N}}_{\varsigma} &= \bot & [\![\mathsf{P}(r_1,\dots,r_n)]\!]^{\mathcal{N}}_{\varsigma} &= \mathsf{P}^{\mathcal{N}}([\![r_1]\!]^{\mathcal{N}}_{\varsigma},\dots,[\![r_n]\!]^{\mathcal{N}}_{\varsigma}) \\ [\![\phi'\boldsymbol{\wedge}\phi]\!]^{\mathcal{N}}_{\varsigma} &= [\![\phi']\!]^{\mathcal{N}}_{\varsigma} \wedge [\![\phi]\!]^{\mathcal{N}}_{\varsigma} & [\![\boldsymbol{\forall} a.\phi]\!]^{\mathcal{N}}_{\varsigma} &= \bigwedge_{x\in|\mathcal{N}|}[\![\phi]\!]^{\mathcal{N}}_{\varsigma[a:=x]} \\ [\![\boldsymbol{\neg}\phi]\!]^{\mathcal{N}}_{\varsigma} &= \neg[\![\phi]\!]^{\mathcal{N}}_{\varsigma} & & \end{array}$$

Theorem 5.6 expresses the usual soundness and completeness result for first-order logic; for details and proofs see e.g. [27, Subsection 1.5]:

Theorem 5.6. *$\Phi \vdash \Psi$ is derivable if and only if for every ordinary model $\mathcal{N}$ and every valuation ς to $\mathcal{N}$ it is the case that $\bigwedge_{\phi\in\Phi}[\![\phi]\!]^{\mathcal{N}}_{\varsigma} = \top$ implies $\bigvee_{\psi\in\Psi}[\![\psi]\!]^{\mathcal{N}}_{\varsigma} = \top$.*

5.2 Constructing a nominal model from an ordinary model

We now show how to 'lift' a model over ordinary sets to a nominal model (Proposition 5.16). We then deduce completeness for nominal Boolean algebras (Corollary 5.19).

Definition 5.7. Give valuations $\varsigma \in \mathbb{A}\Rightarrow|\mathcal{N}|$ a permutation action by

$$(\pi\cdot\varsigma)(a) = \varsigma(\pi^{\text{-}1}(a)).$$

Remark 5.8. $\mathbb{A}\Rightarrow|\mathcal{N}|$ forms a set with a permutation action (Definition 2.3). This is not in general a nominal set because it has elements without finite support[5] for our purposes that will not be a problem.

In fact, the action from Definition 5.7 is a special case of the standard *conjugation action* $(\pi\cdot\varsigma)(a) = \pi\cdot\varsigma(\pi^{\text{-}1}(a))$ where we use the trivial action $\pi\cdot x = x$ for every $x \in |\mathcal{N}|$. This is standard; for a specifically 'nominal' discussion see [14, Definition 2.4.2].

Definition 5.9. Given an ordinary set X write $X^{\bullet} = (|X^{\bullet}|, \cdot)$ for

[5] The finitely-supported ς are such that there exists a finite $A \subseteq \mathbb{A}$ such that for all $a, b \notin A$, $\varsigma(a) = \varsigma(b)$; it is not worth our while to impose this restriction, though it would do no harm to do so.

- $|X^\bullet|$ is the set of functions f from $\mathbb{A}\Rightarrow|\mathcal{N}|$ to X such that there exists a finite set A such that if $\varsigma(a)=\varsigma'(a)$ for every $a\in A$ then $f(\varsigma)=f(\varsigma')$, and
- the permutation action $\cdot$ is defined by
$$(\pi{\cdot}f)(\varsigma)=f(\pi^{\text{-}1}{\cdot}\varsigma).$$

We will be most interested in $|\mathcal{N}|^\bullet$ and $\{\bot,\top\}^\bullet$.

Lemma 5.10. *For any ordinary set X, $X^\bullet$ from Definition 5.9 determines a nominal set (Definition 2.6).*

Proof. It is routine to verify that the permutation action is indeed a group action. It remains to check finite support.

Suppose $\pi(a)=a$ for every $a\in A$ (notation from Definition 5.9). By Definition 5.9 $(\pi{\cdot}f)(\varsigma)=f(\pi^{\text{-}1}{\cdot}\varsigma)$. By Definition 5.7 $(\pi^{\text{-}1}{\cdot}\varsigma)(a)=\varsigma(\pi(a))$. Now by assumption $\varsigma(a)=\varsigma(\pi(a))$ for every $a\in A$. Therefore, $(\pi^{\text{-}1}{\cdot}\varsigma)(a)=\varsigma(a)$ for every $a\in A$, and so $f(\varsigma)=f(\pi^{\text{-}1}{\cdot}\varsigma)$, and so $(\pi{\cdot}f)(\varsigma)=f(\varsigma)$. Thus, f has finite support (and is supported by A).

Definition 5.11. Suppose $\mathcal{N}$ is an ordinary model (Definition 5.2). Specify a tuple $(||\mathcal{N}|^\bullet|,\cdot,\mathit{atm},\mathit{sub})$ by:

- atm is defined by $\mathit{atm}(a)(\varsigma)=\varsigma(a)$.
- sub is defined by $f[a\mapsto g](\varsigma)=f(\varsigma[a{:=}g(\varsigma)])$.

We may write $|\mathcal{N}|^\bullet$ for $(||\mathcal{N}|^\bullet|,\cdot,\mathit{atm},\mathit{sub})$.[6]

Furthermore if X is any ordinary set then (abuse notation again and) write $X^\bullet=(|X^\bullet|,\cdot,\mathit{sub})$ where $f[a\mapsto g](\varsigma)=f(\varsigma[a{:=}g(\varsigma)])$ for $f\in|X^\bullet|$ and $g\in||\mathcal{N}|^\bullet|$.

Lemma 5.12. *Suppose X is an ordinary set and $f\in|X^\bullet|$. Then if $\varsigma(a)=\varsigma'(a)$ for every $a\in\mathit{supp}(f)$ then $f(\varsigma)=f(\varsigma')$.*

Proof. Suppose $\varsigma(a)=\varsigma'(a)$ for every $a\in\mathit{supp}(f)$. It suffices to show that if $a\in A\setminus\mathit{supp}(f)$ and $x\in|\mathcal{N}|$ then $f(\varsigma[a{:=}x])=f(\varsigma)$.

Choose fresh $b:\alpha$ (so $b\notin A$). By part 1 of Corollary 2.10 $(b\ a){\cdot}f=f$, since $a,b\notin\mathit{supp}(f)$. We reason as follows:
$$f(\varsigma[a{:=}x])\overset{b\notin A}{=}f(\varsigma[a{:=}x][b{:=}\varsigma(a)])\overset{(b\,a)\cdot f=f}{=}f(\varsigma[b{:=}x])\overset{b\notin A}{=}f(\varsigma)$$

Corollary 5.13. *$\mathcal{N}^\bullet$ from Definition 5.11 is indeed a termlike substitution algebra, and $X^\bullet$ is indeed a substitution algebra over $\mathcal{N}^\bullet$.*

Proof. We examine Definitions 3.8 and 3.9 and see that we need to check equivariance and the axioms (**Suba**) (for the termlike case) and (**Subid**) to (**Sub**σ) from Figure 2. This is routine; we use Lemma 5.12.

Definition 5.14. Suppose $g',g\in|\{\bot,\top\}^\bullet|$. Write $g'\leq g$ when $\forall\varsigma.g'(\varsigma)\leq g(\varsigma)$.

[6] So we abuse notation by overloading with the nominal set, and $||\mathcal{N}|^\bullet|$ is: the underlying set of the substitution algebra obtained from the underlying set of the ordinary model $\mathcal{N}$. What could be simpler?

Lemma 5.15. $(|\{\bot,\top\}^\bullet|,\cdot,\leq)$ *is a finitely fresh-complete lattice with complements.*

Proof. It is easy to check that $\neg g$ defined by $(\neg g)(\varsigma) = \neg(g(\varsigma))$ is a complement to g.

Now suppose $X \subseteq |\{\bot,\top\}^\bullet|$ and $A \subseteq \mathbb{A}$ are finite, and suppose $A \cap \mathit{supp}(u) = \varnothing$ and $a \in \mathbb{A}\setminus A$. Define

$$(\textstyle\bigwedge^{\#A} X)(\varsigma) = \bigwedge\{g(\varsigma[a{:=}x_a]_{a\in A}) \mid g \in X,\ \forall a{\in}A.x_a \in |\mathcal{N}|\}.$$

It is routine to verify that this is an $A\#$greatest lower bound for X.

Proposition 5.16. $(|\{\bot,\top\}^\bullet|,\cdot,\leq,\mathit{sub})$ *is a nominal Boolean algebra over* $|\mathcal{N}|^\bullet$.

Proof. We unpack Definition 4.1.

- $(|\{\bot,\top\}^\bullet|,\cdot)$ is a nominal set by Lemma 5.10.
- $(|\{\bot,\top\}^\bullet|,\cdot,\leq)$ is by Lemma 5.15 a finitely fresh-complete lattice with complements.
- (**Compat**$\bigwedge$) and (**Compat**$\neg$) follow by easy calculations, since Definition 5.14 is pointwise on valuations.

Definition 5.17. The interpretation of term-formers and predicate-formers is simple:

$$\begin{aligned}\mathsf{f}^{\mathcal{I}}(f_1,\dots,f_{\mathit{arity}(\mathsf{f})})(\varsigma) &= \mathsf{f}^{\mathcal{N}}(f_1(\varsigma),\dots,f_{\mathit{arity}(\mathsf{f})}(\varsigma))\\ \mathsf{P}^{\mathcal{I}}(f_1,\dots,f_{\mathit{arity}(\mathsf{f})})(\varsigma) &= \mathsf{P}^{\mathcal{N}}(f_1(\varsigma),\dots,f_{\mathit{arity}(\mathsf{f})}(\varsigma))\end{aligned}$$

Theorem 5.18. $(|\mathcal{N}|^\bullet,\{\bot,\top\}^\bullet,\text{-}^{\mathcal{I}})$ *determines a model in the sense of Def. 4.9.*

Proof. $|\mathcal{N}|^\bullet$ is a termlike substitution algebra by Corollary 5.13. $\{\bot,\top\}^\bullet$ is a nominal Boolean algebra over $|\mathcal{N}|^\bullet$ by Proposition 5.16. The commutation conditions (**Commf**) and (**CommP**) are by easy calculations.

Corollary 5.19. *If $\Phi \nvdash \Psi$, then there exists a termlike substitution algebra* U *and a nominal Boolean algebra* M *over* U *such that* $\bigwedge_{\phi\in\Phi}[\![\phi]\!]^{\mathcal{M}} \nleq \bigvee_{\psi\in\Psi}[\![\psi]\!]^{\mathcal{M}}$.

As a corollary, if Φ is inconsistent (that is, if $\Phi \vdash \bot$) then Φ has a model.

Proof. By completeness of first-order logic there exists some ordinary model $\mathcal{N}$ (Definition 5.2) and valuation ς to $\mathcal{N}$ such that $\bigwedge_{\phi\in\Phi}[\![\phi]\!]^{\mathcal{N}}_{\varsigma} = \top$ and $\bigvee_{\psi\in\Psi}[\![\psi]\!]^{\mathcal{N}}_{\varsigma} = \bot$. It follows from Definition 5.14 that $\bigwedge_{\phi\in\Phi}[\![\phi]\!]^{\mathcal{M}} \nleq \bigvee_{\psi\in\Psi}[\![\psi]\!]^{\mathcal{M}}$.

It is also possible give a direct proof of completeness of first-order logic with respect to the models of Definition 4.9, using a model based on syntax quotiented by derivable equality. The advantage of the proof of completeness used here is that it also shows that completeness does *not* hold just because the only models available are syntax-based. Topological models [13] and set-theoretic models [11] also exist (those papers predate the consideration of lattices and $A\#$fresh limits of this paper).

6 Conclusions

The usual Tarski semantics, based on valuations, has names in the syntax but not in the semantics. Names can be used to name semantic objects but these objects do not themselves contain names. We have challenged the prejudice that only syntax contains names, and proposed a framework where names are (also) in denotation.

This is related to a broad debate in philosophy about to what extent names are real. One investigation in that literature deserves special mention: Kit Fine's notion of *arbitrary object* [8], which corresponds to our notion of 'names in denotation' and in particular to our notion of nominal substitution set, though the technical details of Fine's constructions are quite different from ours.[7]

This paper is part of a broader research context. For instance in [13] we built a representation theorem for first-order logic (in retrospect this also provides a class of 'topological' models for the axioms in this paper, distinct from the 'lifted' models of Section 5). In [11] we show that, remarkably, a large subclass of the cumulative hierarchy model of Fraenkel-Mostowski sets forms a substitution algebra.

In this paper we apply the technology to the theory of lattices, in particular identifying A#fresh limits as the single unifying notion of greatest lower bound needed to model both conjunction and universal quantification. This is another step towards a 'nominal' view of logic and computation whereby names, and associated structures, are treated as integral to denotation.

Related and future work A nominal style semantics related to the semantics of this paper has been given for the λ-calculus [19]. Nominal lattices have not been considered to date. The author and collaborators considered a notion of *Banona* in [16] (Boolean algebra with $\reflectbox{N}$) which is tangentially related to this paper. Domain theory in nominal sets is considered in [26] and a notion of *FM category* is considered in [3,2]. The notion of A#fresh limit seems to be new though, as is the specifically lattice-theoretic treatment.

A relative of nominal lattices is cylindric set algebras [22,21], a semantics expressed in terms of relations between elements of the domain of the model, where quantification is interpreted using projection and cylindrification. Instead of designing semantics for first-order logic based on relations, projection, and cylindrification, we have taken the novel approach of giving a direct interpretation to variables in the model. In fact, the 'ordinary models' we build in Subsection 5.1 correspond to locally finite dimensional regular cylindric set algebras [22]. There is no non-nominal precedent to the nominal Boolean algebras from Definition 4.1.

There are two natural avenues for future work:

[7] Fine did not have permutations or finite support independently of and preceding the notion of substitution.

Add binders to the term-language. Part of the challenge of this is to develop a suitable new syntax extending first-order logic with term-formers that can bind. Once we have this, it is easy to interpret binders, because our (nominal) denotation supports atoms-abstraction as primitive.[8]

In particular, we see this being applied computationally in formalising mathematics, where binders are everywhere, as an improvement over first-order logic (where term-formers cannot bind) or higher-order logic (which may already be unnecessarily powerful).

This would be related to the *permissive-nominal logic* of [5,6] except that here, we would want to take substitution as primitive. This is related to work by Fiore and his students [10,9] which tries to develop (purely in category theory) a unified account of names, binding, and substitution; a non-exhaustive but accessible introduction to these ideas is in [24]. Slightly further afield, Beeson's *lambda-logic* [1] enriches first-order logic with a term-language that is specifically the untyped λ-calculus; the denotational assumptions used are completely different, but the spirit of allowing binding term-formers in a first-order logic is similar. Of course this also goes back to the first author's *Binding Logic* [7].

Generalise to categories. Another obvious next step is to generalise the constructions in this paper from lattices to categories, and so obtain an enriched category theory. By this view, we note that lattices are just a special case of categories, and our $A\#$fresh limits are just limits in a subcategory of elements of the lattice for which A is fresh. Much can be done here. For instance, we can try to identify the correct common generalisation of this new view of quantifiers and the standard one based on left- and right-adjoints to projection [23, Section 5.5].

Slightly less ambitiously, we can simply consider a general notion of *nominal lattice*, which enriches ordinary lattices with $\bigwedge^{\#a} x$ (a greatest $a\#$lower bound for x) but without necessarily assuming a substitution action.[9] Clearly, there is a rich theory of nominal preorders which remains to be explored.

References

1. Michael Beeson. Lambda logic. In *Second International Joint Conference on Automated Reasoning (IJCAR 2004)*, volume 3097 of *Lecture Notes in Computer Science*, pages 460–474. Springer, 2004.
2. Ranald Clouston. *Equational logic for names and binding.* PhD thesis, University of Cambridge, UK, 2009.
3. Ranald Clouston. Nominal Lawvere theories. In *Proceedings of the 18th International Workshop on Logic, Language, and Information (WoLLIC)*, volume 6642 of *Lecture Notes in Computer Science*. Springer, 2011.
4. Roy L. Crole. α-equivalence equalities. *Theoretical Computer Science*, 2012. In press.
5. Gilles Dowek and Murdoch J. Gabbay. Permissive Nominal Logic. In *Proceedings of the 12th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPDP 2010)*, pages 165–176. ACM Press, 2010.

[8] Atoms-abstraction $[a]x$ is the native (non-functional) notion of abstraction in nominal techniques. It was introduced in [20].

[9] The $\text{И}a.x$ of [16] is not an instance of this because it commutes with negation, but the (dual of) $\nu a.x$ from [15] is an instance of this. There are two distinct 'restriction' operations here.

6. Gilles Dowek and Murdoch J. Gabbay. Permissive Nominal Logic (journal version). *Transactions on Computational Logic*, 2012. In press.
7. Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Binding logic: Proofs and models. In *Proceedings of the 9th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2002)*, pages 130–144. Springer, 2002.
8. Kit Fine. *Reasoning with Arbitrary Objects*. Blackwell, 1985.
9. Marcelo Fiore and Chung-Kil Hur. Term equational systems and logics. *Electronic Notes in Theoretical Computer Science*, 218:171–192, 2008.
10. Marcelo P. Fiore, Gordon D. Plotkin, and Daniele Turi. Abstract syntax and variable binding. In *Proceedings of the 14th IEEE Symposium on Logic in Computer Science (LICS 1999)*, pages 193–202. IEEE Computer Society Press, 1999.
11. Murdoch J. Gabbay. A study of substitution, using nominal techniques and Fraenkel-Mostowski sets. *Theoretical Computer Science*, 410(12-13):1159–1189, March 2009.
12. Murdoch J. Gabbay. Foundations of nominal techniques: logic and semantics of variables in abstract syntax. *Bulletin of Symbolic Logic*, 17(2):161–229, 2011.
13. Murdoch J. Gabbay. Stone duality for First-Order Logic: a nominal approach. In *Howard Barringer Festschrift*. December 2011.
14. Murdoch J. Gabbay. Nominal terms and nominal logics: from foundations to meta-mathematics. In *Handbook of Philosophical Logic*, volume 17. Kluwer, 2012.
15. Murdoch J. Gabbay and Vincenzo Ciancia. Freshness and name-restriction in sets of traces with names. In *Foundations of software science and computation structures, 14th International Conference (FOSSACS 2011)*, volume 6604 of *Lecture Notes in Computer Science*, pages 365–380. Springer, 2011.
16. Murdoch J. Gabbay, Tadeusz Litak, and Daniela Petrişan. Stone duality for nominal Boolean algebras with NEW. In *Proceedings of the 4th international conference on algebra and coalgebra in computer science (CALCO 2011)*, volume 6859 of *Lecture Notes in Computer Science*, pages 192–207. Springer, 2011.
17. Murdoch J. Gabbay and Aad Mathijssen. Capture-avoiding Substitution as a Nominal Algebra. In *ICTAC 2006: Theoretical Aspects of Computing*, volume 4281 of *Lecture Notes in Computer Science*, pages 198–212, November 2006.
18. Murdoch J. Gabbay and Aad Mathijssen. Capture-Avoiding Substitution as a Nominal Algebra. *Formal Aspects of Computing*, 20(4-5):451–479, June 2008.
19. Murdoch J. Gabbay and Dominic Mulligan. Nominal Henkin Semantics: simply-typed lambda-calculus models in nominal sets. In *Proceedings of the 6th International Workshop on Logical Frameworks and Meta-Languages (LFMTP 2011)*, volume 71 of *EPTCS*, pages 58–75, September 2011.
20. Murdoch J. Gabbay and Andrew M. Pitts. A New Approach to Abstract Syntax with Variable Binding. *Formal Aspects of Computing*, 13(3–5):341–363, July 2001.
21. Leon Henkin, J. Donald Monk, and Alfred Tarski. *Cylindric Algebras*. North Holland, 1971 and 1985. Parts I and II.
22. Donald Monk. An introduction to cylindric set algebras. *Logic journal of the IGPL*, 8(4):451–492, 2000.
23. Andrew M. Pitts. Categorical logic. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 5. Algebraic and Logical Structures*, chapter 2. Oxford University Press, 2000.
24. John Power. Abstract syntax: Substitution and binders: Invited address. *Electronic Notes in Theoretical Computer Science*, 173:3–16, 2007.
25. Alfred Tarski. The semantic conception of truth and the foundations of semantics. *Philosophy and Phenomenological Research*, 4, 1944.
26. David C. Turner. *Nominal Domain Theory for Concurrency*. PhD thesis, University of Cambridge, 2009.
27. Dirk van Dalen. *Logic and Structure*. Universitext. Springer, 1994. Third, augmented edition.
28. Steve Vickers. *Topology via Logic*, volume 5 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

Acknowledgements. The second author acknowledges the support of the Leverhulme trust.