Semantics out of context: nominal absolute denotations for first-order logic and computation

Murdoch J. Gabbay, Heriot-Watt University, Scotland, UK *http://www.gabbay.org.uk*

Call a semantics for a language with variables *absolute* when variables map to fixed entities in the denotation. That is, a semantics is absolute when the denotation of a variable a is a copy of itself in the denotation.

We give a trio of lattice-based, sets-based, and algebraic absolute semantics to first-order logic. Possibly open predicates are directly interpreted as lattice elements / sets / algebra elements, subject to suitable interpretations of the connectives and quantifiers. In particular, universal quantification $\forall a.\phi$ is interpreted using a new notion of *'fresh-finite'* limit $\bigwedge^{\#a}[\![\phi]\!]$ and using a novel dual to substitution.

The interest of this semantics is partly in the non-trivial and beautiful technical details, which also offer certain advantages over existing semantics—but also the fact that such semantics exist at all suggests a new way of looking at variables and the foundations of logic and computation, which may be well-suited to the demands of modern computer science.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic**]: Set Theory; F.4.m [**Mathematical Logic**]: Universal algebra and first-order logic; F.3.2 [**Semantics of Programming Languages**]: Algebraic approaches to semantics

Additional Key Words and Phrases: Nominal algebra, semantics, variables, first-order logic, mathematical foundations, sigma-algebra, amgis-algebra, fresh-finite limits, nominal lattices

Contents

1 Introduction **3**
- 1.1 What are variables? . . . 3
- 1.2 The three paradigmatic semantics . . . 4
- 1.3 Map of the paper . . . 5

2 Background on nominal techniques **6**
- 2.1 Basic definitions . . . 7
- 2.2 Examples . . . 8
  - 2.2.1 Atoms . . . 8
  - 2.2.2 Cartesian product . . . 8
  - 2.2.3 Tensor product . . . 8
  - 2.2.4 Full function space . . . 9
  - 2.2.5 Finitely supported function space . . . 9
  - 2.2.6 Full powerset . . . 9
- 2.3 The principle of equivariance and the NEW quantifier . . . 10
- 2.4 Further examples . . . 11
  - 2.4.1 Finitely supported powerset . . . 12
  - 2.4.2 Strictly finitely supported powerset . . . 12

3 Algebras over nominal sets **13**
- 3.1 Definitions: sigma-algebra and amgis-algebra . . . 13
- 3.2 Duality I: sigma to amgis . . . 15
- 3.3 Exact amgis-algebra . . . 16
- 3.4 Duality II: amgis to sigma . . . 17
  - 3.4.1 The pointwise sigma-action on subsets of an amgis-algebra . . . 17
  - 3.4.2 The σ-powerset $Pow_\sigma(\mathcal{P})$ . . . 18
- 3.5 Brief interlude: simultaneous sigma- and amgis-actions . . . 19

4 Nominal posets **20**

I am very grateful to Nachum Dershowitz and to two excellent anonymous referees.

4.1 Nominal posets and fresh-finite limits . . . 20
4.2 σ-algebra structure and fresh-finite limits . . . 22
4.3 Equality . . . 24
4.4 Definition of a FOLeq algebra . . . 24

5 Interpretation of first-order logic in a FOLeq algebra 26
5.1 Syntax and derivability of first-order logic . . . 26
5.2 Sound interpretation in a FOLeq algebra . . . 26

6 The σ-powerset as a FOLeq algebra 29
6.1 Intersection, complement, top element . . . 29
6.2 Quantification . . . 30
6.3 Equality in the σ-powerset . . . 32
6.4 Interpreting first-order logic in the σ-powerset . . . 33

7 Completeness 34
7.1 Filters and points . . . 35
7.1.1 Filters . . . 35
7.1.2 On universal quantification in filters . . . 35
7.1.3 Growing filters . . . 36
7.1.4 Growing ideals . . . 37
7.1.5 Prime filters and ultrafilters . . . 37
7.1.6 The Zorn argument . . . 38
7.2 The amgis-action on (prime) filters . . . 40
7.3 Proof of completeness . . . 41

8 Tarski models 43
8.1 Tarski-style model of first-order logic . . . 43
8.2 Lifting to a FOLeq algebra . . . 44
8.2.1 Lifting to a sigma-algebra . . . 44
8.2.2 Interlude: more on limits in nominal posets . . . 46
8.2.3 Lifting the logical structure . . . 47

9 Herbrand models (Lindenbaum-Tarski algebras) 48

10 Conclusions 49
10.1 Semantics out of context . . . 49
10.2 Recalling the trio of semantics . . . 50
10.3 Related work . . . 51
10.3.1 Nominal precedents . . . 51
10.3.2 Tarski-style valuations and Herbrand-style quotiented syntax . . . 51
10.3.3 Combinators . . . 51
10.3.4 Cylindric and polyadic algebras . . . 51
10.3.5 Boolean algebras with operators . . . 52
10.3.6 Varieties of FOLeq algebras . . . 53
10.3.7 The Schanuel topos: putting context back in, if we want it . . . 53
10.4 Future work . . . 53
10.5 Summary . . . 54

A Nominal algebraic axiomatisation of fresh-finite limits and equality 57
A.1 Bounded meet-semilattices with $\forall$ . . . 57
A.2 Bounded lattices with $\forall$ . . . 58

A.3 FOLeq algebras algebraically 59

B On the Leibnitz equality **61**

1. INTRODUCTION

We give three nominal absolute semantics to first-order logic with equality, based on lattices, sets, and algebra in a nominal universe. Thus we provide alternatives to the *de facto* standard semantics based on valuations, and in doing this we question deep-seated mathematical habits in syntax and semantics, and give evidence that the correct environment for doing logic in computer science is a mathematical foundation with *names*, modelling *variables*, building on the foundational work of Fraenkel and Mostowski.

We treat first-order logic because this is a paradigmatic formal language with variables. It is probably the simplest language with variables of any importance; and, it is of great importance, since it is a language for axiomatising set theory and arithmetic, amongst other things.

The expert and impatient reader, wanting to see just how our models really differ from valuation-based models, might like to skip right to Remark 8.2.16. This is not the most technically advanced part of the paper, but it expresses a point at which we can see one concrete way in which the nominal models are clearly *not* just a rephrasing of standard constructions. Such a reader might also browse Subsection 1.3, which maps out the underlying mathematics, and consider Example 3.4.11.

We now take a step back and discuss the background issues in more depth.

1.1. What are variables?

Variables and quantifiers (or more generally, binders) are widespread. The integral $\int_a$ is a binder which binds a, so that a in $\int_a f(a,b)$ is bound. The same phenomenon appears in logic and computation, so that e.g. we write $\forall a.P(a,b)$, $\lambda a.ab$, $let\ a{=}2\ in\ a{+}b$, and so on.

Our notation for integration has an intended meaning: integration of the function. In logic and computation there are many binders, and much design freedom in interpreting them. The choices we make early on will influence the nature of the mathematics that follows out of them.

We will compare and contrast this in detail in the Conclusions, but for here it is probably fair to say that the main methodology is to treat variables as a look-up to an external context; a quantifier interacts with this context by scanning possible values for the variable. For instance:

— $\int_a f(a,b)$ means "take a value for b to be fixed by some context and vary possible values for a, taking an integral".
— $\forall a.P(a,b)$ means "take a value for b to be fixed by some context and vary possible values for a, taking a(n infinite) logical conjunction".
— $\lambda a.f(a)$ means "input a value from the user, associate that value to a, and calculate f in that context".

In logic and computation this context of variable-to-value assignments is called a *valuation*, and the idea is attributed to Tarski [Tar44].

But here are two other semantics of variables and quantifiers:

— We can treat quantifiers as *modalities* (operators taking a formula and making a new formula), satisfying certain axiomatic properties. Variables are used to label an infinite family of modalities: $\int_a$, $\int_b$, $\int_c$ or $\forall_a$, $\forall_b$, $\forall_c$, ...
This interpretation is useful for proof—a formula may be too complicated to compute, but we can still prove it equivalent to another formula by axiomatic manipulations.
In the context of logic, this is exemplified by cylindric and polyadic algebras due to Halmos and Tarski amongst others [Hal06; HMT85].
— We can treat quantifiers as *binding sites* (distinguished points in the formula), and variables are links/wires connecting different parts of the formula, via the binding site.

This interpretation is useful in programming. For instance, a method name or function declaration binds the invocations of that method or function to the location where it is defined, and a pointer binds a location in memory to the locations in the program where it is dereferenced.

In this paper we investigate a mathematical semantics which builds on and unifies the three interpretations above. This works by reviving an old alternative to Zermelo–Fraenkel sets, known as *Fraenkel-Mostowski sets* (**FM sets**)—we will simply call this *nominal techniques*. Variables have the properties of look-up, *and* axiomatic-modality, *and* binding-sites—depending on how one looks at the nominal semantics.

We can do this because we analyse variables using nominal techniques. Detailed discussions of FM sets and their applications to computing are elsewhere [GP01; Gab11; Pit13]. What matters to us here is that in FM sets, atomic symbols are assumed to exist as *urelemente* or *atoms*.

In the language of programming we would say that atoms are a *datatype* whose job is to contain infinitely many elements that are all symmetric up to permutations, and indeed this is exactly what happens in the Murϕ system [ID96] and the FreshML programming language [SP05]. In some sense, atoms are a polar opposite to the familiar datatype of *natural numbers*, which is a datatype whose job is to contain infinitely many elements that are all totally ordered in a single fixed and canonical manner.

Atoms are *symmetric* up to permutations; symmetry is a primitive property of the mathematical universe. To put this in the context of similar axioms, the Axiom of Infinity assumes the infinitude of the natural numbers, the Axiom of Comprehension assumes that predicates can be used to select subsets of sets, and the Axiom of Replacement assumes that functions can be applied pointwise to sets. In FM sets we assume variable symbols, and their purpose is to be symmetric.

Because symmetry properties are foundationally assumed, they propagate naturally to constructions in FM sets. How this works in full generality is a field of study in its own right. What interests us here is what happens when we try to match up FM sets atoms with $\int_a$, $\forall a$, and λa. That is, can we model the behaviour of variables and binding as special cases of general FM behaviour? This question was answered positively in previous work on abstract syntax, where FM atoms were used to model α-equivalence in syntax [GP01].

But now we want to model the more complex *semantic* behaviour of variables too. We shall find that this works surprisingly well.

At a high level, we shall see that variables are moved from being a specific property of a formal language, which requires explanation on a per-language/per-quantifier basis, to being a generic property of the mathematical universe on a par with generic concepts such as ‘set’, ‘cardinality’, and ‘function’. These can be handled at a high level of abstraction and generalisation, and then instantiated to specific languages and applications.

Doing this is a technical challenge, of course. Just as important as the technical details is the *ideas* that motivate them. And, because variables and binders are so common in formal languages, and so fundamental to how they work, such new ideas about general semantics can pay worthwhile dividends. The necessary mathematics is not particularly complicated, once we understand that we are dealing with a symmetric datatype with characteristic, albeit deep, properties. The necessary nominal background is handled in Section 2, with plenty of examples and exposition. We now discuss the specific technical application.

1.2. The three paradigmatic semantics

We study the concrete example of *first-order logic* (**FOL**). This is a paradigmatic language with binders, and is of practical importance since it is a base language for set theory and arithmetic. The basic FOL connectives are $\bot$ (false), $\wedge$ (conjunction), $\neg$ (negation), and $\forall a$ (first-order quantification).

FOL has three standard semantics: *lattices*, *sets*, and *algebra* (a fourth is *topology*, which is handled in a sister paper [Gab14]). We will briefly summarise how these work:

— In lattices we take a partially ordered set and explain connectives using limits. By this semantics we take some underlying set X of *truth-values* with an *entailment* ordering $\leq$ and assume greatest

lower bounds (and some other structure). Given $x, y \in X$ we declare $x \wedge y$ to be the greatest lower bound (the limit) of x and y.

— In sets we take some underlying set and explain connectives by combining subsets of the underlying set (sets intersection, complement, and so on). That is, we take some set X and subsets $U, V \subseteq X$, and we declare $U \wedge V$ to be $U \cap V$.

— In algebra we equip an underlying set with functions satisfying equalities, which should be abstractly specified, and explain connectives in terms of those functions (de Morgan laws, commutativity, associativity, and so on). That is, we take some set X and functions $\bot$, $\wedge$, $\neg$, and $\forall$ on X which must insist certain axioms—the correct axioms for $\forall$ is a major contribution of this paper.

All these semantics are supposed to match up in some suitable sense. The relevant theorems have standard names: going from lattices to sets is a *representation theorem*; going from algebras to sets is a (sets) *semantics*; and going from sets back to lattices or algebras is a pair of *completeness theorems*. We will treat these, establishing nominal lattice, nominal set, and nominal algebra treatments of FOL and indicating how to move between them.

As discussed, by the nominal approach we intepret variables directly as atoms. We call the result *absolute* because the meaning assigned to a term or predicate will not require any context or valuation; contrast this with the traditional Tarski-style valuation semantics, where the meaning of a term or predicate only exists in the context of a valuation assigning (non-nominal) denotations to the variables. We will also give a detailed comparison with Tarski-style valuation semantics in Section 8, showing how to build one of our models out of a Tarski-style model. This translation will be natural, almost obvious, and it will exhibit a Tarski-style model as a particular special case of our framework (the key idea intuitively is that Tarski-style models are complete in a lattice-theoretic sense, whereas our models are in general only *fresh-finite* complete, which is a weaker condition which is only expressible in a nominal semantics; see Remark 8.2.16).

Returning to our three inter-translatable denotations, how specifically do we address the problem of variables and binding in each of them? Detailed answers are in the body of the paper, but for the reader's convenience we give here—not a summary but—some precise pointers to where the key points in those answers will appear:

— In lattices, $\forall$ is a *fresh-finite limit* (Subsection 4.1). This is a new idea.

— In sets, $\forall$ is characterised twice: as a fresh-finite limit in the powerset considered as a lattive, and as an infinite sets intersection of substitution instances. Theorem 6.4.1 notes that these two characterisations are equivalent.

— This raises an interesting question: since ϕ has an absolute semantics in sets (so ϕ is interpreted as a set, even if ϕ has free variables), then what notion of substitution is given to those sets? Our answer uses $\mathfrak{o}$-*algebras* (Subsection 3.1). These are also a new idea.

— In algebras, $\forall$ is an equivariant function satisfying certain axioms. This idea is relatively recent, but has also been studied in previous work. The axiomatisation is reminiscent of the cylindric algebra or polyadic algebra axiomatisations, though it exists in a nominal algebra framework and has its own distinct character. We discuss this in Subsection A.3.

— Lemma 6.2.2 and Proposition 6.2.3 are important technical results.

More discussion is in the body of the paper and in the Conclusions.

1.3. Map of the paper

Section 2 introduces the necessary nominal background. This section need not necessarily be read first, because the results are abstract. However, this investment in abstraction will pay dividends later when it is applied. These applications are explained in detail in Section 2, see especially the discussion preceding Theorem 2.3.2.

In Section 3 we introduce the notion of algebra over nominal sets (i.e. a nominal set with functions on it satisfying nominal axioms). Specifically, we consider σ- and $\mathfrak{o}$-algebras. A σ-algebra abstracts

those properties of term- and predicate-syntax having to do with substitution; an $\mathfrak{o}$-algebra is a dual to this.

Section 4 introduces nominal posets (a nominal poset is a partially-ordered set, built in the universe of FM sets), but then does some new things with the idea: we study *fresh-finite limits*, how σ-algebra structure interacts with the partial order, we note that simultaneous σ-action can also easily be modelled, we specify equality in the poset, and we conclude with the notion of *FOLeq* algebra. This establishes our lattice-theoretic semantics for first-order logic.

Section 5 spells this semantics out, by declaring first-order logic syntax and defining a formal notion of interpretation. We prove soundness in Theorem 5.2.13.

Section 6 notes the important fact that every σ-powerset (Definition 3.4.6) is a FOLeq algebra—just as every powerset is a Boolean algebra in Zermelo-Fraenkel set theory. This is our sets semantics.

Section 7 proves completeness of the sets semantics for FOLeq algebras (Corollary 7.3.9). The construction visibly parallels the usual ultrafilter construction, but the details are significantly different, due to all the extra structure: the constructions are in nominal sets; points form an $\mathfrak{o}$-algebra (not just a set); the set of all points forms a σ-powerset (not just a powerset); and we treat not only quantification $\forall$ but also equality $=$ which requires careful design of the structure of points. In short, all the structure of a usual completeness proof is still there and still evident, along with extra structure relating the richer foundations.

We are used to seeing models of first-order logic using valuations, in Zermelo-Fraenkel sets; we attribute this idea to Tarski. Section 8 translates such models to our nominal framework, and we see that Tarski valuation models were a special case of FOLeq algebras all along. The main result is Proposition 8.2.15. The converse is not true: not every FOLeq algebra can be expressed as a Tarski valuation model. This is, intuitively, because Tarski models are very complete whereas FOLeq algebras satisfy a weaker property of being only fresh-finite complete in general. What makes this interesting is that in our lattice semantics, being fresh-finite complete captures exactly what is necessary to model first-order logic; so FOLeq semantics are in this sense canonical, and the extra strength of Tarski-style models is due to the relative inexpressivity of the Zermelo-Fraenkel sets foundation which, implicitly, it assumes.

Another standard semantics is syntax-quotiented-by-derivable-equivalence. We call this a *Herbrand* (or *Lindenbaum-Tarski*) semantics and we consider it in Section 9. We do this briefly since, by now, the result should be clear; enough detail is given to reconstruct full proofs if desired.

The algebraisation of FOLeq algebras—their equational axiomatisation—is not without interest or subtlety, but it is also rather easy and has in fact already been treated in previous work [GM06b; GM08c].[1] So we treat it in a brief appendix, Appendix A.

2. BACKGROUND ON NOMINAL TECHNIQUES

Intuitively, a nominal set is "a set $\mathcal{X}$ whose elements $x \in \mathcal{X}$ may 'contain' finitely many names $a, b, c \in \mathbb{A}$". We may call names *atoms*. The notion of 'contain' used here is not the obvious notion of 'is a set element of': formally, we say that x has *finite support* (Definition 2.1.8).

For instance, here are some nominal sets:

— The set of finite sets of atoms:

$$\{\, \varnothing,\ \{a\},\ \{b\},\ \{c\}, \ldots,\ \{a,b\},\ \{a,c\}, \ldots \,\}.$$

— The set of *complements* of finite sets of atoms:

$$\{\, \mathbb{A},\ \mathbb{A}\backslash\{a\},\ \mathbb{A}\backslash\{b\},\ \mathbb{A}\backslash\{c\}, \ldots,\ \mathbb{A}\backslash\{a,b\},\ \mathbb{A}\backslash\{a,c\}, \ldots \,\}.$$

Nominal sets are formally defined in Subsection 2.1, and examples are in Subsections 2.2 and 2.4.

[1] Chronologically the axioms came first, and this paper emerged from efforts to understand these axioms' semantics.

What is most important to realise is that the notion of 'being in the support of x' is not equal to the notion of 'being a set element of x'. For instance if we take $x = \mathbb{A}\backslash\{a\}$, then x contains infinitely many elements—but its support contains precisely the atom that is not an element of x, namely a.

Support measures *name-symmetry*, not name-elementhood. More on this below.

The reader not interested in nominal techniques *per se* might like to read this section only briefly in the first instance, and use it as a reference for the later sections, where these underlying ideas get applied. More detailed expositions are also in [GP01; Gab11].

In the context of the broader literature, the message of this section is as follows:

— The reader with a category-theory background can read this section as stating that we work in the category of nominal sets, or equivalently in the Schanuel topos (more on this in [MM92, Section III.9], [Joh03, A.21, page 79], or [Gab11, Theorem 9.14]).
— The reader with a sets background can read this section as stating that our constructions can be carried out in Fraenkel-Mostowski set theory (**FM sets**).
A discussion of this sets foundation, tailored to nominal techniques, can be found in [Gab11, Section 10]). FM sets add *urelemente* or *atoms* to the sets universe.
— The reader not interested in foundations can note that previous work [GP01; Gab11] has shown that just assuming names as primitive entities in Definition 2.1.1 yields a remarkable clutch of definitions and results, notably Theorem 2.1.11 and Corollary 2.1.12, and Theorems 2.3.2 and 2.3.8. Empirically these properties turn out to be incredibly useful, and they will be just what we need next in Section 3.

2.1. Basic definitions

DEFINITION 2.1.1. Fix a countably infinite set of **atoms** $\mathbb{A}$. We use a **permutative convention** that $a, b, c, \ldots$ range over *distinct* atoms.

DEFINITION 2.1.2. A **(finite) permutation** π is a bijection on atoms such that $nontriv(\pi) = \{a \mid \pi(a) \neq a\}$ is finite.

Write id for the **identity** permutation such that $\mathrm{id}(a) = a$ for all a. Write $\pi' \circ \pi$ for composition, so that $(\pi' \circ \pi)(a) = \pi'(\pi(a))$. Write π^{-1} for inverse, so that $\pi^{-1} \circ \pi = \mathrm{id} = \pi \circ \pi^{-1}$. Write $(a\ b)$ for the **swapping** (terminology from [GP01]) mapping a to b, b to a, and all other c to themselves, and take $(a\ a) = \mathrm{id}$.

NOTATION 2.1.3. If $A \subseteq \mathbb{A}$ write

$$fix(A) = \{\pi \mid \forall a{\in}A.\pi(a) = a\}.$$

DEFINITION 2.1.4. (1) A **set with a permutation action** $\mathcal{X}$ is a pair $(|\mathcal{X}|, \cdot)$ of an **underlying set** $|\mathcal{X}|$ and a **permutation action** written $\pi{\cdot}x$ which is a group action on $|\mathcal{X}|$, so that $\mathrm{id}{\cdot}x = x$ and $\pi{\cdot}(\pi'{\cdot}x) = (\pi \circ \pi'){\cdot}x$ for all $x{\in}|\mathcal{X}|$ and permutations π and π'.
(2) Say that $A \subseteq \mathbb{A}$ **supports** $x{\in}|\mathcal{X}|$ when $\forall \pi.\pi \in fix(A) \Rightarrow \pi{\cdot}x = x$. If a finite A supporting x exists, call x **finitely supported**.

DEFINITION 2.1.5. Call a set with a permutation action $\mathcal{X}$ a **nominal set** when every $x{\in}|\mathcal{X}|$ has finite support. $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$ will range over nominal sets.

DEFINITION 2.1.6. Call a function $f \in |\mathcal{X}{\Rightarrow}\mathcal{Y}|$ **equivariant** when $\pi{\cdot}(f(x)) = f(\pi{\cdot}x)$ for all permutations π and $x{\in}|\mathcal{X}|$. In this case write $f : \mathcal{X}{\Rightarrow}\mathcal{Y}$.

The category of nominal sets and equivariant functions between them is usually called the category of *nominal sets* [Gab01; GP01; Gab11; Pit13].

REMARK 2.1.7. 'Equivariant' appears in Definition 2.1.6 referring to functions between nominal sets, in Definition 2.1.8 referring to elements of nominal sets, and in Theorem 2.3.2 referring to predicates and functions in the language of ZFA set theory.

The notions of equivariance in Definition 2.1.6 and Theorem 2.3.2 are evidently related; the former existing inside the sets universe and the latter outside out. The notion of equivariance in Definition 2.1.8 looks different, but we shall see in Lemma 2.2.2 how it closely relates to the other two.

DEFINITION 2.1.8. Suppose $\mathcal{X}$ is a nominal set and $x{\in}|\mathcal{X}|$. Define the **support** of x by

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

If $supp(x) = \varnothing$ we call x **equivariant**.

NOTATION 2.1.9. — Write $a\#x$ as shorthand for $a \notin supp(x)$ and read this as a is **fresh for** x.
— Given atoms $a_1, \ldots, a_n$ and elements $x_1, \ldots, x_m$ write $a_1, \ldots, a_n\#x_1, \ldots, x_m$ as shorthand for $\{a_1, \ldots, a_n\} \cap \bigcup_{1\leq j\leq m} supp(x_j) = \varnothing$. That is: $a_i\#x_j$ for every i and j.

PROPOSITION 2.1.10. *If $A \subseteq \mathbb{A}$ is finite and supports x, and $a \in \mathbb{A}$ and $a\#x$, then $A\backslash\{a\}$ supports x.*

Proof. Suppose $\pi \in fix(A\backslash\{a\})$. We assumed $a\#x$ so choose an $A' \subseteq \mathbb{A}$ such that A' is finite, A' supports x, and $a \notin A'$. Also, choose some fresh a' (so $a' \notin A{\cup}\{a\}{\cup}A'$).

Write $\tau = (a'\ a)$. Note that $\tau{\cdot}x = x$ by Definition 2.1.4, because A' supports x and $\tau \in fix(A')$.

It is a fact that $(\tau \circ \pi \circ \tau)(a) = a$ for every $a{\in}A$, so $\tau \circ \pi \circ \tau \in fix(A)$. Also by the group action $(\tau \circ \pi \circ \tau){\cdot}x = \tau{\cdot}(\pi{\cdot}(\tau{\cdot}x))$. Since A supports x we have $\tau{\cdot}(\pi{\cdot}(\tau{\cdot}x)) = x$ by Definition 2.1.4.

We apply τ to both sides, recall that $\tau{\cdot}x = x$, and conclude that $\pi{\cdot}x = x$ as required. □

THEOREM 2.1.11. *Suppose $\mathcal{X}$ is a nominal set and $x \in |\mathcal{X}|$. Then $supp(x)$ is the unique least finite set of atoms that supports x.*

Proof. Consider a permutation $\pi \in fix(supp(x))$ and write $\{a_1, \ldots, a_n\} = nontriv(\pi)$. Choose any finite $A{\subseteq}\mathbb{A}$ that supports x, so by construction $supp(x){\subseteq}A$.

By Proposition 2.1.10 $A\backslash nontriv(\pi)$ supports x. By construction $\pi \in fix(A\backslash nontriv(\pi))$, so by Definition 2.1.4 $\pi{\cdot}x = x$ as required. □

COROLLARY 2.1.12. *(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. By routine calculations from the definitions and from Theorem 2.1.11. □

2.2. Examples

Suppose $\mathcal{X}$ and $\mathcal{Y}$ are nominal sets, and suppose $\mathcal{Z}$ is a set with a permutation action. We consider some examples of sets with a permutation action and of nominal sets. These will be useful later on in the paper.

2.2.1. Atoms. $\mathbb{A}$ is a nominal set with the *natural permutation action* $\pi{\cdot}a = \pi(a)$.

For the case of $\mathbb{A}$ only we will be lax about the difference between $\mathbb{A}$ (the set of atoms) and $(|\mathbb{A}|, \cdot)$ (the nominal set of atoms with its natural permutation action). What that means in practice is that we will write $a \in \mathbb{A}$ and never write $a{\in}|\mathbb{A}|$.[2]

2.2.2. Cartesian product. $\mathcal{X} \times \mathcal{Y}$ is a nominal set with underlying set $\{(x, y) \mid x{\in}|\mathcal{X}|, y{\in}|\mathcal{Y}|\}$ and the *pointwise* action $\pi{\cdot}(x, y) = (\pi{\cdot}x, \pi{\cdot}y)$.

2.2.3. Tensor product. $\mathcal{X}{\otimes}\mathcal{Y}$ is a nominal set with underlying set $\{(x, y) \mid x{\in}|\mathcal{X}|, y{\in}|\mathcal{Y}|,\ supp(x){\cap}supp(y) = \varnothing\}$ and the pointwise action. For the pointwise action here to be well-defined depends on π being a permutation and the fact (Proposition 2.3.4 below) that $supp(\pi{\cdot}x) = \pi{\cdot}supp(x)$.

[2] Just sometimes, pedantry has its limit.

2.2.4. Full function space. $\mathcal{X}{\to}\mathcal{Y}$ is a set with a permutation action with underlying set all functions from $|\mathcal{X}|$ to $|\mathcal{Y}|$, and the **conjugation** permutation action

$$(\pi{\cdot}f)(x) = \pi{\cdot}(f(\pi^{-1}{\cdot}x)).$$

The conjugation action can be rephrased as 'permutations distribute' (cf. Theorem 2.3.2 below):

LEMMA 2.2.1. *If* $f{\in}|\mathcal{X}{\to}\mathcal{Y}|$ *then* $\pi{\cdot}f(x) = (\pi{\cdot}f)(\pi{\cdot}x)$.

Proof. By easy calculations. □

2.2.5. Finitely supported function space. $\mathcal{X}{\Rightarrow}\mathcal{Y}$ is a nominal set with underlying set the functions from $|\mathcal{X}|$ to $|\mathcal{Y}|$ with finite support under the conjugation action, and the conjugation permutation action.

LEMMA 2.2.2. $f \in |\mathcal{X}{\Rightarrow}\mathcal{Y}|$ *is equivariant in the sense of Definition 2.1.6 if and only if it is equivariant in the sense of Definition 2.1.8.*

Proof. We sketch the proof: If $\pi{\cdot}f = f$ then for any $x{\in}\mathcal{X}$ we have by Lemma 2.2.1 that $\pi{\cdot}(f(x)) = (\pi{\cdot}f)(\pi{\cdot}x) = f(\pi{\cdot}x)$. Conversely if for any $x{\in}\mathcal{X}$ we have $\pi{\cdot}(f(x)) = f(\pi{\cdot}x)$ then by the conjugation action

$$(\pi{\cdot}f)(x) = \pi{\cdot}(f(\pi^{-1}{\cdot}x)) = f(\pi{\cdot}(\pi^{-1}{\cdot}x)) = f(x).$$ □

2.2.6. Full powerset

DEFINITION 2.2.3. Suppose $\mathcal{X}$ is a set with a permutation action. Give subsets $X \subseteq |\mathcal{X}|$ the **pointwise** permutation action

$$\pi{\cdot}X = \{\pi{\cdot}x \mid x \in X\}.$$

EXAMPLE 2.2.4. A useful instance of the pointwise action is for sets of atoms. As discussed in Subsection 2.2.1 above, if $a \in \mathbb{A}$ then $\pi{\cdot}a = \pi(a)$. Thus if $A \subseteq \mathbb{A}$ then

$$\pi{\cdot}A \quad \text{means} \quad \{\pi(a) \mid a \in A\}.$$

LEMMA 2.2.5. *Continuing the notation of Definition 2.2.3,*

$$X \textit{ is equivariant in the sense of Definition 2.1.8} \quad \Leftrightarrow \quad \forall\pi.\forall x{\in}|\mathcal{X}|.(x \in X \Leftrightarrow \pi{\cdot}x \in X).$$

Proof. If X is equivariant then by Corollary 2.1.12 $\forall\pi.\pi{\cdot}X = X$. It follows that $x \in X \Leftrightarrow x \in \pi^{-1}{\cdot}X \Leftrightarrow \pi{\cdot}x \in X$.

Conversely if $\forall\pi.\forall x{\in}|\mathcal{X}|.(x \in X \Leftrightarrow \pi{\cdot}x \in X)$ then also $x \in X \Leftrightarrow \pi^{-1}{\cdot}x \in X \Leftrightarrow x \in \pi{\cdot}X$, so that $\pi{\cdot}X = X$. This is for any π, so using Corollary 2.1.12 we have that $supp(X) = \varnothing$. □

DEFINITION 2.2.6. Define $PowSet(\mathcal{X})$ the **full powerset** of $\mathcal{X}$ to be the set with a permutation action with

— underlying set $\{X \mid X \subseteq |\mathcal{X}|\}$ (the set of all subsets of $|\mathcal{X}|$), and
— the pointwise action $\pi{\cdot}X = \{\pi{\cdot}x \mid x \in X\}$.

REMARK 2.2.7. Even if $\mathcal{X}$ is a nominal set, $PowSet(\mathcal{X})$ need not be a nominal set. To see why, take $\mathcal{X}$ to be equal to $\mathbb{A} = \{a, b, c, d, e, f, \dots\}$ and consider the set

$$comb = \{a, c, e, \dots\}$$

of 'every other atom'. This does not have finite support, though permutations still act on it pointwise. For more discussion of this point, see [Gab11, Remark 2.18].

We consider further examples in Subsection 2.4.

2.3. The principle of equivariance and the NEW quantifier

We come to Theorem 2.3.2, a result which is central to the 'look and feel' of nominal techniques. It enables a particularly efficient management of renaming and α-conversion in syntax and semantics and captures why it is so useful to use *names* in the foundations of our semantics and not some other infinite set, such as numbers.

Names are by definition symmetric (i.e. can be permuted). Taking names and permutations as *primitive* implies that permutations propagate to the things we build using them. This is the *principle of equivariance* (Theorem 2.3.2 below; see also [Gab11, Subsection 4.2] and [GP01, Lemma 4.7]).

The principle of equivariance implies that, provided we permute names uniformly in all the parameters of our definitions and theorems, we then get another valid set of definitions and theorems. This is not true of e.g. numbers because our mathematical foundation equips numbers by construction with numerical properties such as *less than or equal to* $\leq$, which can be defined from first principles with no parameters.

So if we use numbers for names then we do not care about $\leq$ because we just needed a countable set of elements, but we repeatedly have to *prove* that we did not use an asymmetric property like $\leq$. In contrast, with nominal foundations and atoms, we do not have to explicitly prove symmetry because we can just look at our mathematical foundation and note that it is naturally symmetric under permuting names; we reserve numbers for naturally *a*symmetric activities, such as counting.

This style of name management is characteristic of nominal techniques. The reader will find it used often, e.g. in Lemmas 3.1.8, 3.4.9, and 6.1.2, Propositions 3.2.6 and 4.2.6, and Definitions 3.3.2, 3.4.1, and 6.3.1.

REMARK 2.3.1. The languages of ZFA set theory and FM set theory are identical: first-order logic with equality $=$ and sets membership $\in$.

THEOREM 2.3.2. *If $\overline{x}$ is a list $x_1, \ldots, x_n$, write $\pi{\cdot}\overline{x}$ for $\pi{\cdot}x_1, \ldots, \pi{\cdot}x_n$. Suppose $\Phi(\overline{x})$ is a predicate in the language of ZFA/FM set theory, with free variables $\overline{x}$. Suppose $\Upsilon(\overline{x})$ is a function specified in the language of ZFA/FM set theory, with free variables $\overline{x}$. Then we have the following principles:*

(*1*) **Equivariance of predicates.** $\Phi(\overline{x}) \Leftrightarrow \Phi(\pi{\cdot}\overline{x})$.[3]
(*2*) **Equivariance of functions.** $\pi{\cdot}\Upsilon(\overline{x}) = \Upsilon(\pi{\cdot}\overline{x})$.
(*3*) **Conservation of support.** *If $\overline{x}$ denotes elements with finite support then $supp(\Upsilon(\overline{x})) \subseteq supp(x_1) \cup \cdots \cup supp(x_n)$.*

Proof. See Theorem 4.4, Corollary 4.6, and Theorem 4.7 from [Gab11]. □

REMARK 2.3.3. Theorem 2.3.2 is three fancy ways of observing that if a specification is symmetric in atoms, the the result must be at least as symmetric as the inputs. The benefit of using atoms (instead of e.g. numbers) to model names makes this a one-line argument.[4]

PROPOSITION 2.3.4. *$supp(\pi{\cdot}x) = \pi{\cdot}supp(x)$ (which means $\{\pi(a) \mid a \in supp(x)\}$).*

Proof. Immediate consequence of part 2 of Theorem 2.3.2.[5] □

DEFINITION 2.3.5. Write $\text{И}a.\Phi(a)$ for '$\{a \mid \neg\Phi(a)\}$ is finite'. We call this the **И quantifier**.

[3] Here $\overline{x}$ is understood to contain *all* the variables mentioned in the predicate. It is not the case that $a = a$ if and only if $a = b$—but it is the case that $a = b$ if and only if $b = a$.

[4] The reasoning in this paper could in principle be fully formalised in a sets foundation with atoms, such as Zermelo-Fraenkel set theory with atoms **ZFA**. Nominal sets can be implemented in ZFA sets such that nominal sets map to equivariant elements (elements with empty support) and the permutation action maps to 'real' permutation of atoms in the model. See [Gab11, Subsection 9.3] and [Gab11, Section 4].

[5] There is also a nice proof of this fact by direct calculations; see [Gab11, Theorem 2.19]. However, it just instantiates Theorem 2.3.2 to the particular Υ specifying support.

REMARK 2.3.6. We can read И as 'for all but finitely many a', 'for cofinitely many a', 'for fresh a', or 'for new a'. It captures a *generative* aspect of names, that for any x we can find plenty of atoms a such that $a \notin supp(x)$. И was designed in [GP01] to model the quantifier being used when we informally write "rename x in $\lambda x.t$ to be fresh", or "emit a fresh channel name" or "generate a fresh memory cell".

REMARK 2.3.7. И is a 'for most' quantifier [Wes89], and is a *generalised quantifier* [KW96, Section 1.2.1].

But importantly, И over nominal sets satisfies the *some/any property* that to prove a И-quantified property we test it for *one* fresh atom; we may then use it for *any* fresh atom. This is Theorem 2.3.8, which we use implicitly when later we choose a 'fresh atom' without proving that it does not matter which one we choose. We will do this often. See for instance the proofs of Lemmas 3.4.9 and 7.1.12 (where we write 'for fresh a', we are using the И-quantifier) and Definitions 3.3.2, 3.4.1, and 6.3.1 where И is made explicit.

It is important to understand that И simply means 'for all but finitely many atoms'. This can be encoded in first-order logic. What makes this quantifier so special is the $\forall/\exists$ symmetry property which arises *specifically* when И is applied to symmetric atoms with a background assumption of finite (or more generally, 'small') support—i.e. in a nominal context.

So it is not any one piece of the puzzle that makes this work, but how the pieces interact when they are fit together. This is expressed in Theorem 2.3.8:

THEOREM 2.3.8. *Suppose $\Phi(\overline{z}, a)$ is a predicate in the language of ZFA/FM set theory, with free variables $\overline{z}, a$. Suppose $\overline{z}$ denotes elements with finite support. Then the following are equivalent:*

$$\forall a.(a \in \mathbb{A} \wedge a\#\overline{z}) \Rightarrow \Phi(\overline{z}, a) \qquad И a.\Phi(\overline{z}, a) \qquad \exists a.a \in \mathbb{A} \wedge a\#\overline{z} \wedge \Phi(\overline{z}, a)$$

Proof. Where convenient we may write $\overline{z}$ as $z_1, \ldots, z_n$.

— Suppose $\Phi(\overline{z}, a)$ holds for every atom $a \in \mathbb{A} \setminus \bigcup_{1 \leq i \leq n} supp(z_i)$.
By assumption $\overline{z}$ denotes elements with finite support, and it is a fact that a finite union of finite sets is finite, so $\mathbb{A} \setminus \bigcup_{1 \leq i \leq n} supp(z_i)$ is cofinite.
It follows that $И a.\Phi(\overline{z}, a)$ holds.

— Suppose $A \subseteq \mathbb{A}$ is cofinite and $\Phi(\overline{z}, a)$ for every $a \in A$. As in the previous point, there exists some $a \in A$ such that $a\#z_i$ for every $1 \leq i \leq n$.
It follows that $\exists a \in \mathbb{A}.\big(a\#\overline{z} \wedge \Phi(\overline{z}, a)\big)$.

— Now suppose $\Phi(\overline{z}, a)$ holds for some $a \in \mathbb{A} \setminus \bigcup_{1 \leq i \leq n} supp(z_i)$.
By part 1 of Theorem 2.3.2 $\Phi((a'\ a) \cdot \overline{z}, a')$ holds for any $a' \in \mathbb{A}$. Choosing $a'\#\overline{z}$ we have by part 1 of Corollary 2.1.12 that $(a'\ a) \cdot z_i = z_i$ for every $1 \leq i \leq n$.
Thus $\forall a \in \mathbb{A}.\big(a\#\overline{z} \Rightarrow \Phi(\overline{z}, a)\big)$ holds. □

REMARK 2.3.9. It is impossible to overstate the importance and convenience of the И-quantifier and Theorem 2.3.8, which appears in the literature for instance as Theorem 6.5 from [Gab11] or Proposition 4.10 from [GP01].

Consider Definition 6.3.1 and the proof of Proposition 6.3.8. In that proof, we use α-equivalence to assume a is fresh for u and v. By of the some/any property we can immediately apply Definition 6.3.1.

If we had used a $\forall$ or $\exists$ quantifier in Definition 6.3.1 then we would have had to worry whether the fresh a in the proof of Proposition 6.3.8 was the same fresh atom as that used in Definition 6.3.1. 'Obviously' this is just a hassle; 'obviously' this choice does not matter. Theorem 2.3.8 and the И-quantifier capture precisely and succinctly what that word 'obvious' means.

2.4. Further examples

We now consider the finitely supported powerset and the strictly finitely supported powerset. These examples are more technically challenging and will be key to the later constructions.

2.4.1. Finitely supported powerset. $NomPow(\mathcal{X})$ (the **nominal powerset**) is a nominal set, with

— underlying set those $X{\in}|PowSet(\mathcal{X})|$ that are finitely supported, and
— with the **pointwise** action $\pi{\cdot}X = \{\pi{\cdot}x \mid x \in X\}$ inherited from Definition 2.2.3.

As the name suggests, the nominal powerset is the powerset object in the category of nominal sets [Gab11, Lemma 9.10].

A common source of confusion is to suppose that if A supports $X{\in}|NomPow(\mathcal{X})|$ then A must support every $x \in X$. This is incorrect:

LEMMA 2.4.1. *It is not true in general that if* $X{\in}|NomPow(\mathcal{X})|$ *and* $x \in X$ *then* $supp(x) \subseteq supp(X)$.

Proof. It suffices to provide a counterexample. Take $\mathcal{X} = \mathbb{A}$ (the nominal set of atoms with the natural permutation action, from Subsection 2.2.1) and $X = \mathbb{A} \subseteq |\mathbb{A}|$ (the underlying set of the nominal set of all atoms, i.e. the set of all atoms!).

It is easy to check that $supp(X) = \varnothing$ and $a \in X$ and $supp(a) = \{a\} \not\subseteq \varnothing$. □

Lemma 2.4.1 will lead us to the notion of the *strictly finitely supported* powerset in a moment. For completeness we take just a moment to mention Lemma 2.4.2, which describes a weaker property than that of Lemma 2.4.1 which *is* valid in general; see [GM07, Lemma 5.2] and [GM09, Corollary 4.30] for applications, and [Gab13, Lemma 7.6.2] for the more general context.

LEMMA 2.4.2. *If* $X{\in}|NomPow(\mathcal{X})|$ *and* $a\#X$ *then there exists an* $x \in X$ *with* $a\#x$.

Proof. Choose $x' \in X$ and fresh b (so $b\#x', X$). By Corollary 2.1.12 $(b\ a){\cdot}X = X$, so $(b\ a){\cdot}x \in X$. By Proposition 2.3.4 $a\#(b\ a){\cdot}x$. □

2.4.2. Strictly finitely supported powerset. Suppose $\mathcal{X}$ is a nominal set.

DEFINITION 2.4.3. Call $X \subseteq |\mathcal{X}|$ **strictly supported** by $A \subseteq \mathbb{A}$ when

$$\forall x{\in}X.supp(x) \subseteq A.$$

If there exists some finite A which strictly supports X, then call X **strictly finitely supported** (see [Gab11, Theorem 2.29]).

Write $Strict(\mathcal{X})$ for the set of strictly finitely supported $X \subseteq |\mathcal{X}|$. That is:

$$Strict(\mathcal{X}) = \{X \subseteq |\mathcal{X}| \mid \exists A{\subseteq}\mathbb{A}.A \text{ finite} \wedge X \text{ strictly supported by } A\}$$

LEMMA 2.4.4. *If* $X \in Strict(\mathcal{X})$ *then:*

(1) $\bigcup\{supp(x) \mid x{\in}X\}$ *is finite.*
(2) $\bigcup\{supp(x) \mid x{\in}X\} = supp(X)$.
(3) *If* $X \subseteq |\mathcal{X}|$ *is strictly finitely supported then it is finitely supported.*
(4) $x \in X$ *implies* $supp(x) \subseteq supp(X)$ *(contrast this with Lemma 2.4.1).*
(5) $Strict(\mathcal{X})$ *with the pointwise permutation action is a nominal set.*

Proof. The first part is immediate since by assumption there is some finite $A{\subseteq}\mathbb{A}$ that bounds $supp(x)$ for all $x \in X$. The second part follows by an easy calculation using part 3 of Corollary 2.1.12; full details are in [Gab11, Theorem 2.29], of which Lemma 2.4.4 is a special case. The other parts follow by definitions from the first and second parts. □

EXAMPLE 2.4.5. (1) $\varnothing \subseteq |\mathbb{A}|$ is finitely and strictly finitely supported by $\varnothing$.
(2) $\{a\}$ is finitely supported by $\{a\}$ and also strictly finitely supported by $\{a\}$.
(3) $\mathbb{A} \subseteq |\mathbb{A}|$ is finitely supported by $\varnothing$ but not strictly finitely supported.
(4) $\mathbb{A}\backslash\{a\}$ is finitely supported by $\{a\}$ but not strictly finitely supported.

$(\sigma\mathbf{a})$		$a[a{\mapsto}x] = x$
$(\sigma\mathbf{id})$		$x[a{\mapsto}a] = x$
$(\sigma\#)$	$a\#x \Rightarrow$	$x[a{\mapsto}u] = x$
$(\sigma\alpha)$	$b\#x \Rightarrow$	$x[a{\mapsto}u] = ((b\ a){\cdot}x)[b{\mapsto}u]$
$(\sigma\sigma)$	$a\#v \Rightarrow$	$x[a{\mapsto}u][b{\mapsto}v] = x[b{\mapsto}v][a{\mapsto}u[b{\mapsto}v]]$
$(\mathfrak{o}\sigma)$	$a\#v \Rightarrow$	$p[v{\leftarrow}b][u{\leftarrow}a] = p[u[b{\mapsto}v]{\leftarrow}a][v{\leftarrow}b]$

Fig. 1: Nominal algebra axioms for σ and $\mathfrak{o}$

3. ALGEBRAS OVER NOMINAL SETS

3.1. Definitions: sigma-algebra and amgis-algebra

Definitions 3.1.1, 3.1.5, and 3.1.6 assemble three key technical structures (see also Definitions 3.2.4 and 3.4.6).

Definition 3.1.1. A **termlike σ-algebra** is a tuple $\mathcal{U} = (|\mathcal{U}|, \cdot, \mathsf{sub}_{\mathcal{U}}, \mathsf{atm}_{\mathcal{U}})$ of:

— a nominal set $(|\mathcal{U}|, \cdot)$ which we may write just as $\mathcal{U}$; and
— an equivariant σ**-action** $\mathsf{sub}_{\mathcal{U}} : (\mathcal{U} \times \mathbb{A} \times \mathcal{U}) {\Rightarrow} \mathcal{U}$, written infix $v[a{\mapsto}u]$; and
— an equivariant injection $\mathsf{atm}_{\mathcal{U}} : \mathbb{A} {\Rightarrow} \mathcal{U}$,

such that the equalities $(\sigma\mathbf{a})$, $(\sigma\mathbf{id})$, $(\sigma\#)$, $(\sigma\alpha)$, and $(\sigma\sigma)$ of Figure 1 hold, where x, u, and v range over elements of $|\mathcal{U}|$. We usually write $\mathsf{atm}_{\mathcal{U}}$ invisibly (so we write $\mathsf{atm}_{\mathcal{U}}(a)$ just as a), and we may omit subscripts (so we may write $\mathsf{sub}_{\mathcal{U}}$ as sub if we are confident that $\mathcal{U}$ is clear and understood).

Remark 3.1.2. We unpack what equivariance from Definition 2.1.6 means for the σ-action from Definition 3.1.5: for every $x{\in}|\mathcal{X}|$, atom a, and $u{\in}|\mathcal{U}|$, and for every permutation π, we have that

$$\pi{\cdot}(x[a{\mapsto}u]) = (\pi{\cdot}x)[\pi(a){\mapsto}\pi{\cdot}u].$$

Similarly for the equivariant $\mathfrak{o}$-action in Definition 3.1.6 below.

Example 3.1.3. First-order terms form a termlike σ-algebra:

— Variables are atoms,
— the permutation action is pointwise, and
— the σ-action is 'real' substitution.

Consider for instance a first-order term language for arithmetic: then $a * b$ is a term, $\pi{\cdot}(a * b)$ is $\pi(a) * \pi(b)$, and $(a * b)[a{\mapsto}2][b{\mapsto}3]$ is $2 * 3$. Such a syntax is defined in this paper in Definition 5.1.1 as $r ::= a \mid \mathsf{f}(r_1, \ldots, r_{ar(\mathsf{f})})$.

Similarly, untyped λ-terms quotiented by α-equivalence form a termlike σ-algebra, where the σ-action is capture-avoiding substitution. So $[(\lambda a.b)]_\alpha[b{\mapsto}a] = [\lambda a'.a]_\alpha$.

Predicates of first-order logic quotiented by α-equivalence are not a termlike σ-algebra under capture-avoiding substitution action for terms because predicates do not belong to the same class as terms u. However, predicates do form a (not-necessarily-termlike) σ-algebra; see Definition 3.1.5.

Remark 3.1.4. The 'σ' in σ-action stands for 'substitution'. No connection is suggested with the notion of sigma-algebra from measure theory.

Definition 3.1.5. Suppose $\mathcal{U} = (|\mathcal{U}|, \cdot, \mathsf{sub}, \mathsf{atm})$ is a termlike σ-algebra. A σ**-algebra** over $\mathcal{U}$ is a tuple $\mathcal{X} = (|\mathcal{X}|, \cdot, \mathcal{U}, \mathsf{sub})$ of:

— A nominal set $(|\mathcal{X}|, \cdot)$ which we may write just as $\mathcal{X}$; and
— an equivariant σ**-action** $\mathsf{sub}_{\mathcal{X}} : (\mathcal{X} \times \mathbb{A} \times \mathcal{U}) {\Rightarrow} \mathcal{X}$, written infix $x[a{\mapsto}u]$;

such that the equalities $(\sigma\mathbf{id})$, $(\sigma\#)$, $(\sigma\alpha)$, and $(\sigma\sigma)$ of Figure 1 hold,[6] where x ranges over elements of $|\mathcal{X}|$ and u and v range over elements of $|\mathcal{U}|$. As for termlike σ-algebras, we may omit the subscript $\mathcal{X}$.

DEFINITION 3.1.6. Suppose $\mathcal{U} = (|\mathcal{U}|, \cdot, \mathsf{sub}, \mathsf{atm})$ is a termlike σ-algebra.

An **$\mathfrak{o}$-algebra** (spoken: **amgis**-algebra) over $\mathcal{U}$ is a tuple $\mathcal{P} = (|\mathcal{P}|, \cdot, \mathcal{U}, \mathsf{amgis}_{\mathcal{P}})$ of:

— a set with a permutation action $(|\mathcal{P}|, \cdot)$ which we may write just as $\mathcal{P}$; and
— an equivariant **amgis**-action $\mathsf{amgis}_{\mathcal{P}} : (\mathcal{P} \times \mathcal{U} \times \mathbb{A}) \Rightarrow \mathcal{P}$, written infix $p[u \leftarrow\!\shortmid a]$,

such that the equality $(\mathfrak{o}\sigma)$ of Figure 1 holds, where p ranges over elements of $|\mathcal{P}|$ and u and v range over elements of $|\mathcal{U}|$. We may omit the subscript $\mathcal{P}$.

REMARK 3.1.7. $[u \leftarrow\!\shortmid a]$ looks like $[a \shortmid\!\rightarrow u]$ written backwards, and a casual glance at $(\mathfrak{o}\sigma)$ suggests that it is just $(\sigma\sigma)$ written backwards. This is not quite true: we have $u[b \shortmid\!\rightarrow v]$ on the right in $(\mathfrak{o}\sigma)$ and not '$u[v \leftarrow\!\shortmid b]$' (which would make no sense, since $\mathcal{U}$ has no amgis-action).

Discussion of the origin of the axioms of $\mathfrak{o}$-algebras is in Subsections 3.2 and 3.4; see also Proposition 3.2.6 and Remark 3.2.7.

We conclude this subsection with some technical lemmas which will be useful later.

Lemma 3.1.8 is a technical corollary of $(\sigma\alpha)$, and is used in Lemma 5.2.6:

LEMMA 3.1.8. *If $a\#u$ then $a\#x[a \shortmid\!\rightarrow u]$. As a corollary,*

$$supp(x[a \shortmid\!\rightarrow u]) \subseteq (supp(x) \backslash \{a\}) \cup supp(u).$$

Proof. Choose fresh b (so $b\#x, u$). By $(\sigma\alpha)$ $x[a \shortmid\!\rightarrow u] = ((b\ a)\cdot x)[b \shortmid\!\rightarrow u]$. Also by part 1 of Corollary 2.1.12 $(b\ a)\cdot u = u$ and by Theorem 2.3.2 $(b\ a)\cdot(x[a \shortmid\!\rightarrow u]) = ((b\ a)\cdot x)[b \shortmid\!\rightarrow (b\ a)\cdot u]$. We put this all together and we deduce that $(b\ a)\cdot(x[a \shortmid\!\rightarrow u]) = x[a \shortmid\!\rightarrow u]$. It follows by part 3 of Corollary 2.1.12 that $a \notin supp(x[a \shortmid\!\rightarrow u])$.

The corollary follows since by Theorem 2.3.2 $supp(x[a \shortmid\!\rightarrow u]) \subseteq supp(x) \cup \{a\} \cup supp(u)$. □

REMARK 3.1.9. The reader should know Lemma 3.1.8 for concrete syntax, which is a σ-algebra.

For instance, if ϕ is a predicate of first-order logic (up to α-equivalence) and r is a term, then *support* coincides with *free atoms/variables* fa(-) and $fa(\phi[a \shortmid\!\rightarrow r]) \subseteq (fa(\phi) \backslash \{a\}) \cup fa(r)$.

But, Lemma 3.1.8 is an abstract property of models of nominal algebra axioms. Syntax is one concrete instance of the abstract class (as natural numbers are a concrete instance of rings).

Lemma 3.1.10 goes back to [GM06a; GM08a], where it was taken as an axiom or the original nominal algebraic treatment of substitution. In the presence of $(\sigma\alpha)$ it is equivalent to $(\sigma\mathbf{id})$; it is useful in the proofs of Propositions 4.2.6 and 6.3.5:

LEMMA 3.1.10. *If $b\#x$ then $x[a \shortmid\!\rightarrow b] = (b\ a)\cdot x$.*

Proof. By $(\sigma\mathbf{id})$ $(b\ a)\cdot x = ((b\ a)\cdot x)[b \shortmid\!\rightarrow b]$. By $(\sigma\alpha)$ $((b\ a)\cdot x)[b \shortmid\!\rightarrow b] = x[a \shortmid\!\rightarrow b]$. □

Lemma 3.1.11 will be useful to prove Proposition 4.2.6 and 4.2.8. See also the related Lemma 8.2.8:

LEMMA 3.1.11. *If $U \subseteq |\mathcal{U}|$ is finitely supported and $a\#U$ then $a\#\{x[a \shortmid\!\rightarrow u] \mid u \in U\}$. As corollaries, the following freshnesses all hold:*

$$a\#\{x[a \shortmid\!\rightarrow u] \mid u \in |\mathcal{U}|\}$$
$$a\#\{x[a \shortmid\!\rightarrow n] \mid n \in \mathbb{A}\}$$

Proof. We use part 3 of Corollary 2.1.12. Choose fresh b (so $b\#x, U$). Note by Corollary 2.1.12 (since $a, b\#U$) that $(b\ a)\cdot U = U$ so that by Lemma 2.2.5 $\forall u \in |\mathcal{U}|. u \in U \Leftrightarrow (b\ a)\cdot u \in U$. Then we reason

[6] That is, the σ axioms except $(\sigma\mathbf{a})$, since we do not assume a function $\mathtt{atm}_{\mathcal{X}}$.

as follows:

$$\begin{array}{rll}(b\ a){\cdot}\{x[a{\mapsto}u] \mid u{\in}U\} & = \{(b\ a){\cdot}(x[a{\mapsto}u]) \mid u{\in}U\} & \text{Pointwise action}\\ & = \{((b\ a){\cdot}x)[b{\mapsto}(b\ a){\cdot}u] \mid u{\in}U\} & \text{Theorem 2.3.2}\\ & = \{((b\ a){\cdot}x)[b{\mapsto}u] \mid u{\in}U\} & (b\ a){\cdot}U = U\\ & = \{x[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\} & (\sigma\alpha),\ b\#x\end{array}$$

The proof for $a\#\{x[a{\mapsto}n] \mid n{\in}\mathbb{A}\}$ is similar. □

3.2. Duality I: sigma to amgis

In Subsection 3.2 we explore how to move from a σ-algebra to an $\mathfrak{o}$-algebra; we explore the other direction in Subsection 3.4.

Given a σ-algebra we generate an $\mathfrak{o}$-algebra out of its subsets. This is Proposition 3.2.6.

Definition 3.2.1. Suppose $\mathcal{X} = (|\mathcal{X}|, \cdot, \mathcal{U}, \mathsf{sub}_{\mathcal{X}})$ is a σ-algebra over a termlike σ-algebra $\mathcal{U}$.

Give subsets $p \subseteq |\mathcal{X}|$ **pointwise** actions as follows:

$$\begin{array}{rl}\pi{\cdot}p = \{\pi{\cdot}x \mid x \in p\}\\ p[u{\leftarrow}a] = \{x{\in}|\mathcal{X}| \mid x[a{\mapsto}u] \in p\} & u{\in}|\mathcal{U}|\end{array}$$

Proposition 3.2.2. *Suppose $\mathcal{X}$ is a σ-algebra over a termlike σ-algebra $\mathcal{U}$. Suppose $p \subseteq |\mathcal{X}|$. Then:*

— $x \in p[u{\leftarrow}a]$ *if and only if* $x[a{\mapsto}u] \in p$.
— $x \in \pi{\cdot}p$ *if and only if* $\pi^{-1}{\cdot}x \in p$.

Proof. By easy calculations on the pointwise actions in Definition 3.2.1. □

Remark 3.2.3. We take a moment to suggest intuitively why Definition 3.2.1 is natural.

The pointwise permutation action is the natural action on subsets, also mentioned in Subsection 2.4.1. This comes from the Fraenkel-Mostowski foundations.

The pointwise $\mathfrak{o}$-action is its natural generalisation from a group to a monoid (which need not necessarily have inverses). But why *amgis*? Why does $[a{\mapsto}u]$ get turned round to $[u{\leftarrow}a]$?

When we take the powerset of a set V, V is in *negative position* (i.e. $PowSet(V)$ is equivalent to a function-space $V{\to}2$ and here V is to the left of the arrow). This implies that any modal operations which we assume on elements of V, need to be 'flipped'. Thus, σ-algebras turn into $\mathfrak{o}$-algebras and vice versa. This is why an underlying set with a σ-action gives rise to a 'flipped'—a *dual*—$\mathfrak{o}$-action on the subsets.

Definition 3.2.4. Suppose $\mathcal{X}$ is a σ-algebra over a termlike σ-algebra $\mathcal{U}$.

Define the **$\mathfrak{o}$-powerset** algebra $Pow_{\mathfrak{o}}(\mathcal{X})$ by setting

— $|Pow_{\mathfrak{o}}(\mathcal{X})|$ to be the set of all subsets $p \subseteq |\mathcal{X}|$ (Definition 2.2.6) with
— permutation action $\pi{\cdot}p$ and
— amgis-action $p[u{\leftarrow}a]$ from Definition 3.2.1.

Remark 3.2.5. Note that $p \subseteq |\mathcal{X}|$ need not have finite support in Definition 3.2.4.

Our notion of $\mathfrak{o}$-algebra (Definition 3.1.6) admits p without finite support. We will need this: the $\mathfrak{o}$-algebras we construct in Definition 7.2.2 need not have finite support; see the discussion in Remark 7.1.2. The action happens in Theorem 7.1.20 where we use Zorn's Lemma to make infinitely many choices.

Proposition 3.2.6. *If $\mathcal{X}$ is a σ-algebra over a termlike σ-algebra $\mathcal{U}$ then $Pow_{\mathfrak{o}}(\mathcal{X})$ (Definition 3.2.4) is an $\mathfrak{o}$-algebra over $\mathcal{U}$.*

Proof. By Theorem 2.3.2 the operations are equivariant. We verify rule $(\mathfrak{o}\sigma)$ from Figure 1:

— *Property* $(\mathfrak{o}\sigma)$. We use $(\sigma\sigma)$. Suppose $a\#v$. Then:

$$\begin{array}{rll} x \in p[v{\leftarrow}b][u{\leftarrow}a] & \Leftrightarrow x[a{\mapsto}u][b{\mapsto}v] \in p & \text{Proposition 3.2.2} \\ & \Leftrightarrow x[b{\mapsto}v][a{\mapsto}u[b{\mapsto}v]] \in p & (\sigma\sigma),\ a\#v \\ & \Leftrightarrow x \in p[u[b{\mapsto}v]{\leftarrow}a][v{\leftarrow}b] & \text{Proposition 3.2.2} \end{array}$$ □

REMARK 3.2.7. It is interesting to note a non-result of $Pow_{\mathfrak{o}}(\mathcal{X})$. Consider $(\sigma\#)$; let us try to dualise it as we dualised $(\sigma\sigma)$ to $(\mathfrak{o}\sigma)$ in Proposition 3.2.6.

$$(\mathfrak{o}\#) \qquad a\#p \Rightarrow p[u{\leftarrow}a] = p \qquad \text{Bad axiom}$$

Suppose $a\#p$. Then

$$\begin{array}{rll} x \in p[u{\leftarrow}a] & \Leftrightarrow x[a{\mapsto}u] \in p & \text{Proposition 3.2.2} \\ & \Leftrightarrow ???? & \end{array}$$

and now we are stuck: $a\#p$ does not imply that $a\#x$, so we cannot use $(\sigma\#)$ (e.g. if $p = \mathbb{A}$ then $b \in \mathbb{A}$ and $b \in supp(b)$; more on this in [Gab11, Subsection 9.5]).

REMARK 3.2.8. It has been suggested that if σ-algebra algebraises substitution on syntax, then $\mathfrak{o}$-algebra should algebraise pattern-matching on syntax. This is not quite right; pattern-matching does not satisfy $(\mathfrak{o}\sigma)$. Suppose a syntax of terms with pairing (r_1, r_2) and one unary term-former f. Consider the term $(\mathsf{f}(c), \mathsf{f}(b))$ and interpret $[a{\mapsto}r]$ as substitution and $[r{\leftarrow}a]$ as pattern-matching. Then

$$(\mathsf{f}(c),\mathsf{f}(b))[c{\leftarrow}b][\mathsf{f}(b){\leftarrow}a] = (\mathsf{f}(b),\mathsf{f}(b))[\mathsf{f}(b){\leftarrow}a] = (a,a) \quad \text{and}$$
$$(\mathsf{f}(c),\mathsf{f}(b))[\mathsf{f}(b)[b{\mapsto}c]{\leftarrow}a][c{\leftarrow}b] = (\mathsf{f}(c),\mathsf{f}(b))[\mathsf{f}(c){\leftarrow}a][c{\leftarrow}b] = (a,\mathsf{f}(b))[c{\leftarrow}b] = (a,\mathsf{f}(b))$$

and $(a,a) \neq (a,\mathsf{f}(b))$.

The natural examples of $\mathfrak{o}$-algebras are the $\mathfrak{o}$-powersets as constructed above in Definition 3.2.4.

3.3. Exact amgis-algebra

We now strengthen Proposition 3.2.6 to Proposition 3.3.3, which states that $Pow_{\mathfrak{o}}(\mathcal{X})$ is an *exact* $\mathfrak{o}$-algebra (Definition 3.3.2).

Exactness will be useful later to interpret equality in Pow_{σ}; see Subsection 6.3 and in particular Lemma 6.3.4.

To understand exactness, it is interesting to consider an easy property of σ-algebras:

LEMMA 3.3.1. *If $a\#x, y$ then $x[a{\mapsto}u] = y[a{\mapsto}u]$ implies $x = y$.*

Proof. Immediate from $(\sigma\#)$. □

We cannot hope to replicate the proof above directly for $\mathfrak{o}$-algebras using an axiom $(\mathfrak{o}\#)$ because, as noted in Remark 3.2.7, we do *not* want such an axiom.

But $\mathfrak{o}$ is a kind of dual to σ, and *exactness* is a kind of dual to Lemma 3.3.1. Recall the definition of $\mathfrak{o}$-algebra from Definition 3.1.6 and the И-quantifier from Definition 2.3.5.

DEFINITION 3.3.2. Call an $\mathfrak{o}$-algebra $\mathcal{P}$ **exact** when if $p, q{\in}|\mathcal{P}|$ and $u{\in}|\mathcal{U}|$ then

$$И c.p[u{\leftarrow}c] = q[u{\leftarrow}c] \quad \text{implies} \quad p = q.$$

In words, $\mathcal{P}$ is exact when for every p, q, and u, if $p[u{\leftarrow}c]$ and $q[u{\leftarrow}c]$ are equal for *most* (meaning 'for all but finitely meany') c then p and q are equal.

PROPOSITION 3.3.3. *Suppose $\mathcal{X}$ is a σ-algebra over a termlike σ-algebra $\mathcal{U}$. Then $Pow_{\mathfrak{o}}(\mathcal{X})$ is an exact $\mathfrak{o}$-algebra over $\mathcal{U}$.*

Proof. By Proposition 3.2.6 $Pow_{\mathfrak{o}}(\mathcal{X})$ is an $\mathfrak{o}$-algebra over $\mathcal{U}$. It remains to prove exactness.

Suppose $p, q{\in}|Pow_{\mathfrak{o}}(\mathcal{X})|$ and $u{\in}|\mathcal{U}|$, and suppose $И c.p[u{\leftarrow}c] = q[u{\leftarrow}c]$. We now check that for all $x{\in}|\mathcal{X}|$ it is the case that $x \in p$ if and only if $x \in q$. We reason as follows:

$$\begin{array}{lll} x \in p \Leftrightarrow \mathit{И}c.x[c{\mapsto}u] \in p & & (\sigma\#),\ c\#x \\ \Leftrightarrow \mathit{И}c.x \in p[u{\leftarrow}c] & & \text{Proposition 3.2.2} \\ \Leftrightarrow \mathit{И}c.x \in q[u{\leftarrow}c] & & \text{Assumption} \\ \Leftrightarrow \mathit{И}c.x[c{\mapsto}u] \in q & & \text{Proposition 3.2.2} \\ \Leftrightarrow x \in q & & (\sigma\#),\ c\#x \end{array}$$ □

Example 3.3.4. The set of atoms $\mathbb{A}$ is a termlike σ-algebra over itself (so $a[a{\mapsto}c] = c$ and $a[b{\mapsto}c] = a$). By Proposition 3.2.6 the full powerset of $\mathbb{A}$ is an $\mathfrak{o}$-algebra over $\mathbb{A}$, and by Proposition 3.3.3 this $\mathfrak{o}$-algebra is exact.

So suppose $p, q \subseteq \mathbb{A}$ and $\mathit{И}c.p[a{\leftarrow}c] = q[a{\leftarrow}c]$. This means that for all but finitely many atoms c it is the case that for every $x{\in}\mathbb{A}$, $x[c{\mapsto}a] \in p$ if and only if $x[c{\mapsto}a] \in q$. In particular for every x there exists some $c \neq x$ such that $x \in p$ if and only if $x \in q$. Thus, p and q are equal.

Note that exactness is not an algebraic property (it has the form *if …then* rather than the form *LHS = RHS*). So although the class of all $\mathfrak{o}$-algebras is algebraic, the class of *exact* $\mathfrak{o}$-algebras is not.[7] This will not be a problem.

3.4. Duality II: amgis to sigma

In Subsection 3.2 we showed how to build an $\mathfrak{o}$-algebra out of a σ-algebra. Dually, we can build a σ-algebra out of an $\mathfrak{o}$-algebra; this is a little harder because, while we are free to define $\mathfrak{o}$-algebra to suit ourselves, the notion of σ-algebra is already quite fixed by the behaviour of substitution, which it is intended to model. The set of all finitely supported subsets is too large, so Definition 3.4.6 cuts this down with additional conditions 1 and 2.

3.4.1. The pointwise sigma-action on subsets of an amgis-algebra

Definition 3.4.1. Suppose $\mathcal{P} = (|\mathcal{P}|, \cdot, \mathcal{U}, \mathsf{amgis}_{\mathcal{P}})$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$. Give subsets $X \subseteq |\mathcal{P}|$ **pointwise** actions as follows:

$$\begin{array}{ll} \pi{\cdot}X = \{\pi{\cdot}x \mid x \in X\} & \\ X[a{\mapsto}u] = \{p{\in}|\mathcal{P}| \mid \mathit{И}c.p[u{\leftarrow}c] \in (c\ a){\cdot}X\} & u{\in}|\mathcal{U}| \end{array}$$

Proposition 3.4.2. *Suppose $\mathcal{P}$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$ and suppose $X \subseteq |\mathcal{P}|$.*[8] *Suppose $p{\in}|\mathcal{P}|$ and $u{\in}|\mathcal{U}|$ and $a\#u$. Then:*

(1) $p \in X[a{\mapsto}u]$ if and only if $\mathit{И}c.p[u{\leftarrow}c] \in (c\ a){\cdot}X$.[9]
(2) $p \in \pi{\cdot}X$ if and only if $\pi^{-1}{\cdot}p \in X$.

Proof. (1) Direct from Definition 3.4.1.
(2) Direct from Theorem 2.3.2. □

Remark 3.4.3. Definitions 3.2.1 and 3.4.1 are not perfectly symmetric; Definition 3.4.1 contains a $\mathit{И}$-quantifier. It is there specifically to make Lemma 3.4.4 true; in other words we guarantee axiom $(\sigma\alpha)$ from Figure 1 by construction:[10]

Lemma 3.4.4 (α-equivalence). *Suppose $\mathcal{P}$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$ and suppose $X \subseteq |\mathcal{P}|$. Then if $b\#X$ then $X[a{\mapsto}u] = ((b\ a){\cdot}X)[b{\mapsto}u]$.*

Proof. By part 1 of Proposition 3.4.2 $p \in X[a{\mapsto}u]$ if and only $\mathit{И}c.p[u{\leftarrow}c] \in (c\ a){\cdot}X$, and $p \in ((b\ a){\cdot}X)[b{\mapsto}u]$ if and only if $\mathit{И}c.p[u{\leftarrow}c] \in (c\ b){\cdot}((b\ a){\cdot}X)$. By Corollary 2.1.12 $(c\ a){\cdot}X = (c\ b){\cdot}((b\ a){\cdot}X)$ since $b\#X$. The result follows. □

[7] To be precise, the class of all $\mathfrak{o}$-algebras is a *variety* in the nominal algebraic sense of [Gab09a], whereas the class of exact $\mathfrak{o}$-algebras is a *quasivariety*.
[8] We care most about the case that $X \in |NomPow(\mathcal{P})|$—$X$ is finitely supported—but this result does not depend on that.
[9] Recall that p need not have finite support here; see Remark 3.2.5.
[10] The proof of Lemma 6.3.6 will be in one line precisely because of this design for Definition 3.4.1.

Lemma 3.4.5 is useful, amongst other things, in Lemma 3.4.9. On syntax it is known as the *substitution lemma*, but here it is about an action on sets X, and the proof is different:

LEMMA 3.4.5. *Suppose $\mathcal{P}$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$ and suppose $X \subseteq |\mathcal{P}|$. Suppose $u, v \in |\mathcal{U}|$. Then*

$$a\#v \quad \textit{implies} \quad X[a{\mapsto}u][b{\mapsto}v] = X[b{\mapsto}v][a{\mapsto}u[b{\mapsto}v]].$$

Proof. We reason as follows, where we write $\pi = (a'\ a) \circ (b'\ b)$:

$$\begin{array}{lll}
p \in X[a{\mapsto}u][b{\mapsto}v] & \Leftrightarrow \mathit{И}a', b'.p[v{\leftarrow\!\shortmid}b'][(b'\ b){\cdot}u{\leftarrow\!\shortmid}a'] \in \pi{\cdot}X & \text{Proposition 3.4.2} \\
& \Leftrightarrow \mathit{И}a', b'.p[((b'\ b){\cdot}u)[b'{\mapsto}v]{\leftarrow\!\shortmid}a'][v{\leftarrow\!\shortmid}b'] \in \pi{\cdot}X & (\mathfrak{o}\sigma)\ a'\#v \\
& \Leftrightarrow \mathit{И}a', b'.p[u[b{\mapsto}v]{\leftarrow\!\shortmid}a'][v{\leftarrow\!\shortmid}b'] \in \pi{\cdot}X & (\sigma\alpha)\ b'\#u \\
& \Leftrightarrow \mathit{И}a'.p[u[b{\mapsto}v]{\leftarrow\!\shortmid}a'] \in ((a'\ a){\cdot}X)[b{\mapsto}v] & \text{Proposition 3.4.2} \\
& \Leftrightarrow \mathit{И}a'.p[u[b{\mapsto}v]{\leftarrow\!\shortmid}a'] \in ((a'\ a){\cdot}X)[b{\mapsto}(a'\ a){\cdot}v] & \text{Corollary 2.1.12 } a', a\#v \\
& \Leftrightarrow \mathit{И}a'.p[u[b{\mapsto}v]{\leftarrow\!\shortmid}a'] \in (a'\ a){\cdot}(X[b{\mapsto}v]) & \text{Theorem 2.3.2} \\
& \Leftrightarrow p \in X[b{\mapsto}v][a{\mapsto}u[b{\mapsto}v]] & \text{Proposition 3.4.2}
\end{array}$$

□

3.4.2. The σ-powerset $Pow_\sigma(\mathcal{P})$. Recall from Subsection 2.4.1 the *finitely supported powerset* $NomPow(\mathcal{X})$ of a nominal set $\mathcal{X}$.

DEFINITION 3.4.6. Suppose $\mathcal{P}$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$. Define the **σ-powerset** algebra $Pow_\sigma(\mathcal{P})$ by setting

— $|Pow_\sigma(\mathcal{P})|$ to be those $X \in |NomPow(\mathcal{P})|$ (finitely supported subsets of $|\mathcal{P}|$; see Subsection 2.4.1) with
— the actions $\pi{\cdot}X$ and $X[a{\mapsto}u]$ from Definition 3.4.1,

satisfying conditions 1 and 2 below, where $u \in |\mathcal{U}|$ and $p \in |\mathcal{P}|$:

(1) $\forall u.\mathit{И}a.\forall p.(p[u{\leftarrow\!\shortmid}a] \in X \Leftrightarrow p \in X)$.
(2) $\forall a.\mathit{И}b.\forall p.(p[b{\leftarrow\!\shortmid}a] \in X \Leftrightarrow (b\ a){\cdot}p \in X)$.

Lemma 3.4.7 rephrases conditions 1 and 2 of Definition 3.4.6, in a simpler language, albeit one which requires the σ-action on subsets of an $\mathfrak{o}$-algebra from Definition 3.4.1:

LEMMA 3.4.7. *Continuing the notation of Definition 3.4.6, if $X \in |Pow_\sigma(\mathcal{P})|$ then*

(1) If $a\#X$ then $X[a{\mapsto}u] = X$.
(2) If $b\#X$ then $X[a{\mapsto}b] = (b\ a){\cdot}X$.

Proof. (1) Suppose $a\#X$. By part 1 of Lemma 3.4.2 $p \in X[a{\mapsto}u]$ if and only if $\mathit{И}c.p[u{\leftarrow\!\shortmid}c] \in (c\ a){\cdot}X$. By Corollary 2.1.12 $(c\ a){\cdot}X = X$ and by condition 1 of Definition 3.4.6 $p[u{\leftarrow\!\shortmid}c] \in X$ if and only if $p \in X$, so this is if and only if $\mathit{И}c.(p \in X)$, that is $p \in X$.
(2) We combine Proposition 3.4.2 with condition 2 of Definition 3.4.6, since $a\#b$. □

COROLLARY 3.4.8. *Suppose $X \in |Pow_\sigma(\mathcal{P})|$. Then $X[a{\mapsto}a] = X$.*

Proof. Suppose $b\#X$. By Lemma 3.4.4 $X[a{\mapsto}a] = ((b\ a){\cdot}X)[b{\mapsto}a]$. Note that by Proposition 2.3.4 $a\#(b\ a){\cdot}X$. By part 2 of Lemma 3.4.7 $((b\ a){\cdot}X)[b{\mapsto}a] = (b\ a){\cdot}((b\ a){\cdot}X) = X$. □

LEMMA 3.4.9. *If $X \in |Pow_\sigma(\mathcal{P})|$ and $u \in |\mathcal{U}|$ then also $X[a{\mapsto}u] \in |Pow_\sigma(\mathcal{P})|$.*
As a corollary, in Definition 3.4.6, $|Pow_\sigma(\mathcal{P})|$ is closed under the σ-action from Definition 3.4.1.

Proof. By construction $X[a{\mapsto}u] \subseteq |\mathcal{P}|$, so we now check the properties listed in Definition 3.4.6.
By assumption in Definition 3.4.6, X is finitely supported. Finite support of $X[a{\mapsto}u]$ is from Theorem 2.3.2.
We check the conditions of Definition 3.4.6 for $X[a{\mapsto}u]$:

(1) *For fresh b (so $b\#u, X$),* $X[a{\mapsto}u][b{\mapsto}v] = X[a{\mapsto}u]$.
We use Lemma 3.4.4 to assume without loss of generality that $a\#u$. It suffices to reason as follows:

$$\begin{array}{rll} X[a{\mapsto}u][b{\mapsto}v] & = X[b{\mapsto}v][a{\mapsto}u[b{\mapsto}v]] & \text{Lemma 3.4.5, } a\#v \\ & = X[b{\mapsto}v][a{\mapsto}u] & (\sigma\#),\ b\#u \\ & = X[a{\mapsto}u] & \text{Part 1 of Lemma 3.4.7, } b\#X \end{array}$$

(2) *For fresh b' (so $b'\#u, v, X$)* $X[a{\mapsto}u][b{\mapsto}b'] = (b'\ b){\cdot}(X[a{\mapsto}u])$.
It suffices to reason as follows:

$$\begin{array}{rll} X[a{\mapsto}u][b{\mapsto}b'] & = X[b{\mapsto}b'][a{\mapsto}u[b{\mapsto}b']] & \text{Lemma 3.4.5, } a\#b' \\ & = ((b'\ b){\cdot}X)[a{\mapsto}(b'\ b){\cdot}u] & \text{Lemma 3.4.7, } b'\#u, X \\ & = (b'\ b){\cdot}(X[a{\mapsto}u]) & \text{Part 2 of Theorem 2.3.2} \end{array}$$

□

PROPOSITION 3.4.10. *If $\mathcal{P}$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$ then $Pow_\sigma(\mathcal{P})$ (Definition 3.4.6) is a σ-algebra over $\mathcal{U}$.*

Proof. By Lemma 3.4.9 the σ-action does indeed map to $|Pow_\sigma(\mathcal{P})|$. By Theorem 2.3.2 so does the permutation action. It remains to check validity of the axioms from Definition 3.1.5.

— Axiom $(\sigma\mathbf{id})$ is Corollary 3.4.8.
— Axiom $(\sigma\#)$ is part 1 of Lemma 3.4.7.
— Axiom $(\sigma\alpha)$ is Lemma 3.4.4.
— Axiom $(\sigma\sigma)$ is Lemma 3.4.5. □

EXAMPLE 3.4.11. Consider some set of terms considered as a termlike σ-algebra over themselves (Example 3.1.3; so $s[a{\mapsto}t]$ is s with t substituted for a). Write this $\mathfrak{TRM}$.

By Propositions 3.4.10 and 3.2.6 $Pow_\sigma(Pow_\mathfrak{o}(\mathfrak{TRM}))$ is a σ-algebra over $\mathfrak{TRM}$.[11] It has Boolean structure given by sets intersection and complement, and by design it has a σ-algebra structure.

So perhaps $Pow_\sigma(Pow_\mathfrak{o}(\mathfrak{TRM}))$ is a model of first-order logic with equality; it is a σ-algebra so we might model quantification an infinite sets intersection. How to interpret equality is less obvious …but we might be lucky.

In the rest of this paper we make this formal, prove it, and put the result in the context of the other models: in nominal posets, using maximally consistent sets (for the completeness result), Tarski-style valuation models, and Herbrand models. If the reader holds on to the idea that

> *to a first approximation this paper is about abstracting, axiomatising, and analysing the behaviour of $Pow_\sigma(Pow_\mathfrak{o}(\mathfrak{TRM}))$,*

then they should not go too far wrong.

3.5. Brief interlude: simultaneous sigma- and amgis-actions

In Subsection 3.1 we only defined a σ-action $x[a{\mapsto}u]$ for a single atom at a time. From Subsection 4.3 and onwards it will be useful to consider a simultaneous σ-action. In fact the axioms of Figure 1 give us the power of a simultaneous action, for σ (but not $\mathfrak{o}$, that is not $p[u{\leftarrow\!\shortmid}a]$, as we shall observe).

Suppose $\mathcal{X}$ is a σ-algebra over a termlike σ-algebra $\mathcal{U}$.

DEFINITION 3.5.1. Suppose $u_1, \ldots, u_n{\in}|\mathcal{U}|$ and suppose $a_1, \ldots, a_n$ are n distinct atoms not in $\bigcup_i supp(u_i)$. Then for $x{\in}|\mathcal{X}|$ define

$$x[a_1{\mapsto}u_1, \ldots, a_n{\mapsto}u_n] = x[a_1{\mapsto}u_1] \ldots [a_n{\mapsto}u_n].$$

We need to show that Definition 3.5.1 does not depend on the order in which we take the u_i. This follows using $(\sigma\sigma)$ and $(\mathfrak{o}\sigma)$, and $(\sigma\#)$, because we assumed that each a_i is not in $\bigcup_i supp(u_i)$.

Now we extend this to the case where the a_i are not necessarily fresh for the u_i:

[11] $Pow_\mathfrak{o}$ is the $\mathfrak{o}$-powerset and is from Definition 3.2.4. Pow_σ is the *σ-powerset* and is from 3.4.6.

DEFINITION 3.5.2. Suppose $u_1,\ldots,u_n{\in}|\mathcal{U}|$ and suppose $a_1,\ldots,a_n$ are any n distinct atoms. Then for $x{\in}|\mathcal{X}|$ define

$$x[a_1{\mapsto}u_1,\ldots,a_n{\mapsto}u_n] = (((a'_1\ a_1)\circ\cdots\circ(a'_n\ a_n))\cdot x)[a'_1{\mapsto}u_1]\ldots[a'_n{\mapsto}u_n].$$

where we choose $a'_1,\ldots,a'_n$ fresh (so not in $\{a_1,\ldots,a_n\}\cup \mathit{supp}(x)\cup\bigcup_i \mathit{supp}(u_i)$).

LEMMA 3.5.3. *The choice of fresh a'_i and the order of the u_i in Definition 3.5.2, do not matter.*

Proof. By routine calculations using $(\sigma\alpha)$, $(\sigma\sigma)$, and $(\sigma\#)$. □

It is natural to try to duplicate Definition 3.5.2 and Lemma 3.5.3 for $\mathfrak{o}$-algebras. Suppose $\mathcal{P}$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$. Lemma 3.5.4 is a partial dual to Lemma 3.5.3 and has a simple proof:

LEMMA 3.5.4. *Suppose $u,v{\in}|\mathcal{U}|$ and $p{\in}|\mathcal{P}|$. Then*

$$a\#v, b\#u \Rightarrow p[u{\leftarrow\!\shortmid}a][v{\leftarrow\!\shortmid}b] = p[v{\leftarrow\!\shortmid}b][u{\leftarrow\!\shortmid}a].$$

Proof. We reason as follows:

$$\begin{array}{lll} p[v{\leftarrow\!\shortmid}b][u{\leftarrow\!\shortmid}a] = p[u[b{\mapsto}v]{\leftarrow\!\shortmid}a][v{\leftarrow\!\shortmid}b] & & (\mathfrak{o}\sigma),\ a\#v \\ \quad = p[u{\leftarrow\!\shortmid}a][v{\leftarrow\!\shortmid}b] & & (\sigma\#),\ b\#u \end{array}$$ □

Lemma 3.5.4 asserts that the atoms provided are 'sufficiently fresh', then the order of the $\mathfrak{o}$-action does not matter. However, we cannot freshen atoms as we can for σ-action using $(\sigma\alpha)$ from Figure 1—that is, there is no rule $(\mathfrak{o}\alpha)$ of the form ~~$b\#p \Rightarrow p[u{\leftarrow\!\shortmid}a] = (b\ a){\cdot}(p[u{\leftarrow\!\shortmid}b])$~~. Lemma 3.5.5 is the closest we can get to such a result. We mention it without proof:

LEMMA 3.5.5. *Suppose $\mathcal{P}$ is a finitely-supported $\mathfrak{o}$-algebra (so every $p{\in}|\mathcal{P}|$ has finite support) and $\mathcal{X}$ is a σ-algebra over a termlike σ-algebra $\mathcal{U}$ and suppose $u{\in}|\mathcal{U}|$. Then:*

(1) *If $X{\in}|\mathit{Pow}_\sigma(\mathcal{P})|$ and $b\#X,p,u$ and $p{\in}|\mathcal{P}|$ and $a\#p,u$ then $p[u{\leftarrow\!\shortmid}a]\ \in\ X$ if and only if $(b\ a){\cdot}(p[u{\leftarrow\!\shortmid}b])\in X$.*
(2) *If $x{\in}|\mathcal{X}|$ and $b\#x$ and $p{\in}|\mathit{Pow}_\mathfrak{o}(\mathcal{X})|$ then $x\in p[u{\leftarrow\!\shortmid}a]$ if and only if $x\in(b\ a){\cdot}(p[u{\leftarrow\!\shortmid}b])$.*

4. NOMINAL POSETS

4.1. Nominal posets and fresh-finite limits

DEFINITION 4.1.1. A **nominal poset** is a tuple $\mathcal{L}=(|\mathcal{L}|,\cdot,\leq)$ where

— $(|\mathcal{L}|,\cdot)$ is a nominal set, and
— The relation $\leq\ \subseteq |\mathcal{L}|{\times}|\mathcal{L}|$ is an equivariant partial order.[12]

Call $\mathcal{L}$ **finitely fresh-complete** or say it has **fresh-finite limits** 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**

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

has a $\leq$-greatest element $\bigwedge^{\#A}X$, which we may call the A**#limit** (A-fresh limit) or A**#greatest lower bound** of X.

Similarly call $\mathcal{L}$ **finitely fresh-cocomplete** or say it has **fresh-finite colimits** when for every finite X and A as above the set of A-fresh upper bounds

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

has a $\leq$-least element $\bigvee^{\#A}X$, which we may call its A**#colimit** or A**#least upper bound** of X.

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

EXAMPLE 4.1.2. Predicates of first-order logic quotiented by derivable logical equivalence and partially ordered by logical entailment, form a nominal poset; that is $[\phi]$ is the derivable equivalence class of ϕ and $[\phi] \leq [\psi]$ when $\phi \vdash \psi$. The permutation action is pointwise on variable symbols.

Then it is a fact that $[\phi \wedge \psi]$ is a limit for $\{[\phi], [\psi]\}$ and $[\forall a.\phi]$ is an $a\#$limit for $\{[\phi]\}$. We sketch this particular example in a little more detail in Section 9.

NOTATION 4.1.3. Suppose $\mathcal{L} = (|\mathcal{L}|, \cdot, \leq)$ is nominal poset. Suppose $X \subseteq |\mathcal{L}|$ and $A \subseteq \mathbb{A}$ are finite and suppose $x \in |\mathcal{L}|$.

— Call $\bigwedge^{\#\varnothing} X$ a **limit** or **greatest lower bound** of X.
— Call $\bigwedge^{\#\{a\}}\{x\}$ the $a\#$limit or $a\#$greatest lower bound of x and write it $\bigwedge^{\#a} x$. Unpacking Definition 4.1.1,

$$\textstyle\bigwedge^{\#a} x \quad \text{is the greatest element of} \quad \{x' \in |\mathcal{L}| \mid x' \leq x \ \wedge \ a\#x'\}.$$

— Call $\bigvee^{\#\varnothing} X$ a **colimit** or **least upper bound** of X.
— Call $\bigvee^{\#\{a\}}\{x\}$ the $a\#$colimit or $a\#$least upper bound of x and write it $\bigvee^{\#a} x$.
— Write $\top$ for the greatest lower bound and $\bot$ for the least upper bound of the empty set $\varnothing$.
— Write $x \wedge y$ for the greatest lower bound and $x \vee y$ for the least upper bound of $\{x, y\}$.

REMARK 4.1.4. So *$A\#$(co)limits* generalise (co)limits; if we take $A = \varnothing$ then we get a limit (greatest lower bound) just as we are used to.

There is a convenient factoring of 'finitely fresh-complete' into three constituent parts:

PROPOSITION 4.1.5. *Suppose $\mathcal{L}$ is a nominal poset. Then $\mathcal{L}$ is finitely fresh-complete if and only if it has limits of the following three forms:*

— *$\top$, a greatest element (limit for the empty set $\varnothing$).*
— *$x \wedge y$, a limit for $\{x, y\}$.*
— *$\bigwedge^{\#a} x$, an $a\#$limit for $\{x\}$.*

Similarly finitely fresh-cocomplete is equivalent to having $\bot$, $x \vee y$, and $\bigvee^{\#a} x$.

Proof. The interesting part is the right-to-left implication where X is non-empty. Given finite non-empty $\{x_1, \ldots, x_n\} \subseteq |\mathcal{L}|$ and $\{a_1, \ldots, a_n\} \subseteq \mathbb{A}$, it is not hard to verify that $\bigwedge^{\#a_1} \ldots \bigwedge^{\#a_n} (\ldots (x_1 \wedge x_2) \ldots \wedge x_n)$ is an $A\#$limit for X. □

DEFINITION 4.1.6. Suppose $\mathcal{L}$ is a partial order (it need not be nominal, though we will care most about the case that it is). 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 $\mathcal{L}$ **is complemented**, and write the complement of x as $\neg x$.

LEMMA 4.1.7. *Fresh-finite (co)limits, and complements, are unique if they exist, and $\neg\neg x = x$.*

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

COROLLARY 4.1.8. *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 $supp(\bigwedge^{\#A} X) \subseteq \bigcup\{supp(x) \mid x \in X\} \backslash A$.
(2) Suppose $x \in |\mathcal{L}|$, and suppose $\neg x$ exists. Then $supp(\neg x) = supp(x)$.

Proof. (1) By part 3 of Theorem 2.3.2 $supp(\bigwedge^{\#A} X) \subseteq \bigcup\{supp(x) \mid x \in X\} \cup A$. Since by assumption $A \cap supp(\bigwedge^{\#A} X) = \varnothing$, the result follows.
(2) By part 3 of Theorem 2.3.2 observing that the map $x \mapsto \neg x$ is its own inverse. □

Lemma 4.1.9 is α-equivalence for fresh-finite limits:

LEMMA 4.1.9. *Suppose $\mathcal{L}$ is a nominal poset and $x \in |\mathcal{L}|$.*
If $b\#x$ then $\bigwedge^{\#a} x = \bigwedge^{\#b} (b\ a) \cdot x$ and $\bigvee^{\#a} x = \bigvee^{\#b} (b\ a) \cdot x$, where $\bigwedge^{\#a} x$ and $\bigvee^{\#a} x$ exist.

Proof. By Corollary 4.1.8 $a, b\#\bigwedge^{\#a}x$, so by part 1 of Corollary 2.1.12 $\bigwedge^{\#a}x = (b\ a)\cdot\bigwedge^{\#a}x$. By part 2 of Theorem 2.3.2 $(b\ a)\cdot\bigwedge^{\#a}x = \bigwedge^{\#b}(b\ a)\cdot x$. The case of $\bigvee^{\#a}x$ is similar. □

We will concentrate on limits from now on; the case of colimits is dual and in the presence of negation can be obtained directly from limits. A sequel to this subsection—which we postpoone until we need it later—is in Subsection 8.2.2.

4.2. σ-algebra structure and fresh-finite limits

Definition 4.2.1. Suppose a finitely fresh-complete and finitely fresh-cocomplete nominal poset $\mathcal{L} = (|\mathcal{L}|, \cdot, \leq)$ is also a σ-algebra $(|\mathcal{L}|, \cdot, \mathcal{U}, \mathsf{sub}_{\mathcal{U}})$ over a termlike σ-algebra $\mathcal{U}$. Call the σ-algebra structure **monotone** when for every $x, y{\in}|\mathcal{L}|$ and $u{\in}|\mathcal{U}|$

$$x \leq y \quad \text{implies} \quad x[a{\mapsto}u] \leq y[a{\mapsto}u]$$

and **compatible** when for every finite $X \subseteq |\mathcal{L}|$ and $A \subseteq \mathbb{A}$ and $u{\in}|\mathcal{U}|$

$$\begin{array}{rl}(\bigwedge^{\#A}X)[a{\mapsto}u] &= \bigwedge^{\#A}\{x[a{\mapsto}u] \mid x \in X\} \ \text{ provided } A \cap (\mathit{supp}(u) \cup \{a\}) = \varnothing\\ (\bigvee^{\#A}X)[a{\mapsto}u] &= \bigvee^{\#A}\{x[a{\mapsto}u] \mid x \in X\} \ \text{ provided } A \cap (\mathit{supp}(u) \cup \{a\}) = \varnothing\\ (\neg x)[a{\mapsto}u] &= \neg(x[a{\mapsto}u]).\end{array}$$

Remark 4.2.2. As standard, negation converts greatest lower bounds to least upper bounds. Thus we obtain $\bigvee^{\#A}X$ as $\neg\bigwedge^{\#A}\{\neg x \mid x \in X\}$. So the second condition above follows from the first and third.

Lemma 4.2.3. *Given a finitely fresh-complete nominal poset with a compatible σ-action:*

(1) $(x \wedge y)[a{\mapsto}u] = x[a{\mapsto}u] \wedge (y[a{\mapsto}u])$ *and similarly for* $\vee$.
(2) If $x \leq y$ *then* $x[a{\mapsto}u] \leq y[a{\mapsto}u]$ *(so a compatible σ-action is monotone).*
(3) If $b\#a, u$ *then* $(\bigwedge^{\#b}x)[a{\mapsto}u] = \bigwedge^{\#b}(x[a{\mapsto}u])$ *and similarly for* $\bigvee^{\#b}$.
(4) $\top[a{\mapsto}u] = \top$ *and* $\bot[a{\mapsto}u] = \bot$.

Proof. Parts 1, 3, and 4 are special cases of compatibility.[13] Part 2 follows from part 1 using the fact that $x \leq y$ if and only if $x \wedge y = x$. □

Lemma 4.2.4. *In a finitely fresh-complete nominal poset:*

— *If* $z \leq x$ *and* $a\#z$ *then* $z \leq \bigwedge^{\#a}x$.
— *If the σ-action is monotone then if* $z \leq x$ *and* $a\#z$ *then* $z \leq x[a{\mapsto}u]$ *for every u, so that z is a lower bound for the (in general infinite) set* $\{x[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\}$.[14]

Proof. The first part is direct from the definition of fresh-finite limit; $\bigwedge^{\#a}x$ is by definition a least upper bound for all $z \leq x$ such that $a\#z$.

For the second part, since the σ-action is monotone $z[a{\mapsto}u] \leq x[a{\mapsto}u]$ for every $u{\in}|\mathcal{U}|$. By $(\sigma\#)$ also $z = z[a{\mapsto}u]$. It follows that $z \leq x[a{\mapsto}u]$ for every $u{\in}|\mathcal{U}|$. □

Remark 4.2.5. Clearly, we intend the $a\#$limit $\bigwedge^{\#a}x$ to model the universal quantifier $\forall a.\phi$, just as the finite limit $x \wedge y$ models logical conjunction $\phi \wedge \psi$.

An odd thing about the quantifier rules is that $(\forall\mathbf{L})$ expresses $\forall a.\phi$ as a greatest lower bound of an infinite set $\{\phi[a{\mapsto}r] \mid \text{all } r\}$ whereas $(\forall\mathbf{R})$ expresses $\forall a.\phi$ as something else and more finite (these standard derivation rules feature in this paper in Figure 2). Lemma 4.2.4 explains this dual nature of $\forall$ as an equality between

— the limit of a fairly large set of elements $\{x[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\}$—nice for an elimination rule, so we can eliminate for many u—and

[13] Part 4 has an alternative one–line proof from Theorem 2.3.2 and $(\sigma\#)$; we use this argument in Lemma 5.2.11 for the case of $\bot$.
[14] The set $\{x[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\}$ could be finite: either $a\#x$ so that by $(\sigma\#)$ $x[a{\mapsto}u] = x$ for all u; or $|\mathcal{U}|$ is finite, so that in particular $\mathsf{atm}_{\mathcal{U}}$ maps every a to some constant element in $|\mathcal{U}|$.

— the (fresh-)finite limit of the singleton set $\{x\}$—nice for an introduction rule, giving us but a single proof-obligation.

See also Proposition 4.2.6 and Subsection 6.2.

Proposition 4.2.6 proves an equality between the infinite greatest lower bound $\bigwedge_u x[a{\mapsto}u]$ and the (fresh-)finite greatest lower bound $\bigwedge\!\!\!\!\bigwedge^{\#a} x$, provided one of these exists and the σ-action is monotone. That is, the limit of a diagram over infinitely many elements $x[a{\mapsto}u]$ is equivalent to the $a\#$limit of just x.

PROPOSITION 4.2.6. *Suppose $\mathcal{L}$ is a nominal poset with a monotone σ-action*[15] *and $x{\in}|\mathcal{L}|$. Then:*

(1) If $\bigwedge\!\!\!\!\bigwedge^{\#a} x$ exists then so does $\bigwedge_{u\in|\mathcal{U}|} x[a{\mapsto}u]$ the limit for $\{x[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\}$, and they are equal. In symbols:

$$\bigwedge\!\!\!\!\bigwedge^{\#a} x = \bigwedge_{u\in|\mathcal{U}|} x[a{\mapsto}u].$$

(2) If $\bigwedge_u x[a{\mapsto}u]$ exists then so does $\bigwedge\!\!\!\!\bigwedge^{\#a} x$, and they are equal.

Proof. Suppose $\bigwedge\!\!\!\!\bigwedge^{\#a} x$ exists.

— By Lemma 4.2.4 $\bigwedge\!\!\!\!\bigwedge^{\#a} x$ is a lower bound for $\{x[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\}$.
— Now suppose z is any lower bound for $\{x[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\}$. So $z \leq x[a{\mapsto}u]$ for every $u{\in}|\mathcal{U}|$. Note that we do not know *a priori* that $a\#z$. Choose b fresh (so $b\#z, x$) and take $u = b$. Then $z \leq x[a{\mapsto}b] \overset{\text{L3.1.10}}{=} (b\ a){\cdot}x$. We assumed that x has an $a\#$limit so by Theorem 2.3.2 also $(b\ a){\cdot}x$ has a $b\#$limit (which by Lemma 4.1.9 is equal to the $a\#$limit of x). Since $b\#z$ it follows that $z \leq \bigwedge\!\!\!\!\bigwedge^{\#b}(b\ a){\cdot}x \overset{\text{L4.1.9}}{=} \bigwedge\!\!\!\!\bigwedge^{\#a} x$.

It follows that $\bigwedge\!\!\!\!\bigwedge^{\#a} x = \bigwedge_u x[a{\mapsto}u]$.

Now suppose $\bigwedge_u x[a{\mapsto}u]$ exists. By Lemma 3.1.11 and part 2 of Theorem 2.3.2 we have that $a\#\bigwedge_u x[a{\mapsto}u]$.

— By assumption $\bigwedge_u x[a{\mapsto}u] \leq x[a{\mapsto}a] \overset{(\sigma\mathbf{id})}{=} x$. Thus $\bigwedge_u x[a{\mapsto}u]$ is an $a\#$lower bound for x.
— Now suppose $z \leq x$ and $a\#z$; we need to show that $z \leq \bigwedge_u x[a{\mapsto}u]$. This is direct from Lemma 4.2.4.

It follows that $\bigwedge_u x[a{\mapsto}u] = \bigwedge\!\!\!\!\bigwedge^{\#a} x$. □

REMARK 4.2.7. Note that Proposition 4.2.6 does *not* mean 'finitely fresh-complete = complete', where *being complete* means having limits for all sets (or rather, all finitely supported sets). Proposition 4.2.6 only shows that 'finitely fresh-complete' is the same as 'complete for sets that can be expressed as a finite union of sets of the form $\{x[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\}$'.

We briefly mention another characterisation of $\bigwedge\!\!\!\!\bigwedge^{\#a} x$ using a 'smaller' conjunction, though we never use it:

PROPOSITION 4.2.8. *Suppose $\mathcal{L}$ is a nominal poset with a monotone σ-action and $x{\in}|\mathcal{L}|$. Then:*

(1) If $\bigwedge\!\!\!\!\bigwedge^{\#a} x$ exists then so does $\bigwedge_n x[a{\mapsto}n]$ where n ranges over all atoms, and they are equal.
(2) If $\bigwedge_n x[a{\mapsto}n]$ exists then so does $\bigwedge\!\!\!\!\bigwedge^{\#a} x$, and they are equal.[16]

Proof. As the proof of Proposition 4.2.6. The important point is that by Lemma 3.1.11 and part 2 of Theorem 2.3.2 we have $a\#\bigwedge_n x[a{\mapsto}n]$. □

[15] We will only really care about the case that the σ-action satisfies the stronger property of being *compatible* (Definition 4.2.1). However, compatibility includes conditions involving $\bigwedge\!\!\!\!\bigwedge^{\#a}$, so we prefer to be more precise and only insist on what we need to make the result work, which is monotonicity.

[16] Strictly speaking we should write $\bigwedge_n x[a{\mapsto}\mathtt{atm}_{\mathcal{U}}(n)]$. See the notation in Definition 3.1.1.

4.3. Equality

Suppose $\mathcal{L}$ is a finitely fresh-complete nominal poset[17] with a compatible σ-algebra structure over a termlike σ-algebra $\mathcal{U}$.

DEFINITION 4.3.1. An **equality** is an element $(a{=}^{\mathcal{L}}b){\in}|\mathcal{L}|$ with $supp(a{=}^{\mathcal{L}}b) \subseteq \{a,b\}$ and such that:

(1) For every $u{\in}|\mathcal{U}|$,

$$(a{=}^{\mathcal{L}}b)[a{\mapsto}u, b{\mapsto}u] = \top.$$

(2) For every $u,v{\in}|\mathcal{U}|$ and $z{\in}|\mathcal{L}|$,

$$(a{=}^{\mathcal{L}}b)[a{\mapsto}u, b{\mapsto}v] \wedge z[a{\mapsto}u] = (a{=}^{\mathcal{L}}b)[a{\mapsto}u, b{\mapsto}v] \wedge z[a{\mapsto}v].$$

It might interest the reader to look briefly ahead to Definition 6.3.1, where a very special and useful equality element will be constructed.

REMARK 4.3.2. The choice of a and b is arbitrary (simultaneous σ-action $[a{\mapsto}u, b{\mapsto}v]$ is from Definition 3.5.1).

Anticipating the notation of Definition 5.2.5, if we write $(a{=}^{\mathcal{L}}b)[a{\mapsto}u, b{\mapsto}v]$ as $(u{=}^{\mathcal{L}}v)$ then properties 1 and 2 of Definition 4.3.1 can be rewritten in the following more economical form

$$(u{=}^{\mathcal{L}}u) = \top \qquad (u{=}^{\mathcal{L}}v) \wedge z[a{\mapsto}u] = (u{=}^{\mathcal{L}}v) \wedge z[a{\mapsto}v]$$

and this looks like a purely equational rendering of the sequent rules (**=R**) and (**=L**) of Figure 2 (sequent rules for equality). This is deliberate; see condition 5 of Theorem A.3.1.

In Definition 4.3.1 we talk about *an* equality. However—in the same spirit as Lemma 4.1.7—if one such element exists then it is unique:

PROPOSITION 4.3.3. *An equality in $\mathcal{L}$ is unique, up to choice of a and b, if it exists.*

Proof. Consider two equalities $a{=}^{\mathcal{L}}_1 b$ and $a{=}^{\mathcal{L}}_2 b$. Take $z = (a{=}^{\mathcal{L}}_2 b)$ and $u = a$ and $v = b$ in condition 2 of Definition 4.3.1. Then using (σ**id**),

$$(a{=}^{\mathcal{L}}_1 b) \wedge (a{=}^{\mathcal{L}}_2 b) = (a{=}^{\mathcal{L}}_1 b) \wedge (a{=}^{\mathcal{L}}_2 b)[b{\mapsto}a].$$

Now by condition 1 for $(a{=}^{\mathcal{L}}_2 b)$ we know that $(a{=}^{\mathcal{L}}_2 b)[b{\mapsto}a] = \top$. It follows that $(a{=}^{\mathcal{L}}_1 b) \leq (a{=}^{\mathcal{L}}_2 b)$. By symmetry also $(a{=}^{\mathcal{L}}_2 b) \leq (a{=}^{\mathcal{L}}_1 b)$ and we are done. □

It is not hard to similarly prove that equality is reflexive, symmetric, and transitive, if it exists.

4.4. Definition of a FOLeq algebra

DEFINITION 4.4.1. Suppose $\mathcal{L}$ is a finitely fresh-complete and finitely fresh-cocomplete nominal poset (Definition 4.1.1).

Call $\mathcal{L}$ **distributive** when:

(1) $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.
(2) If $a\#x$ then $x \vee (\bigwedge^{\#a} y) = \bigwedge^{\#a}(x \vee y)$.

These two conditions can be unified into a single condition as follows, for finite $A \subseteq \mathbb{A}$ and $X \subseteq |\mathcal{L}|$:

$$A{\cap}supp(x){=}\varnothing \;\Rightarrow\; x \vee \bigwedge^{\#A} X = \bigwedge^{\#A}\{x{\vee}x' \mid x'{\in}X\}$$

REMARK 4.4.2. Evidently, Definition 4.4.1 generalises the usual notion of distributivity. A dual version of part 1 of Definition 4.4.1 is $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ and by a standard argument [DP02, Lemma 4.3] the two are equivalent.

A dual version of part 2 of Definition 4.4.1 is that if $a\#x$ then $x \wedge (\bigvee^{\#a} y) = \bigvee^{\#a}(x \wedge y)$. This is *not* equivalent to part 2 of Definition 4.4.1 in general, but in the presence of complements it is, just by taking negation.

[17] In fact we only care about $\wedge$ here, not $\bigwedge^{\#a}$.

DEFINITION 4.4.3. A **FOLeq algebra** over a termlike σ-algebra $\mathcal{U}$ is a nominal poset $\mathcal{L}$ with the following properties:

— $\mathcal{L}$ is finitely fresh-complete and finitely fresh-cocomplete.
— $\mathcal{L}$ is distributive.
— $\mathcal{L}$ is complemented.
— $\mathcal{L}$ has a compatible σ-algebra structure over $\mathcal{U}$.
— $\mathcal{L}$ has an equality.

If $\mathcal{L}$ is as above except that it does not have an equality, then we call $\mathcal{L}$ a **FOL algebra**.

REMARK 4.4.4. For the reader's convenience we break down Definition 4.4.3 with precise references:

— *Nominal poset* and *finitely fresh-(co)complete* are Definition 4.1.1.
— *Distributive* is Definition 4.4.1.
— *Complements* are Definition 4.1.6.
— *σ-algebra structure* is Definition 3.1.5 and
— *Compatibility* is Definition 4.2.1.
— *Having equality* is Definition 4.3.1.

As the name suggests, the notion of FOLeq algebra is an abstract nominal specification of what it is to be a model of first-order logic with equality. The connection between this definition and first-order logic is surely clear:

— *finitely fresh-(co)complete* gives us $\wedge$, $\vee$, $\bigwedge^{\#a}$ and $\bigvee^{\#a}$, which are nominal algebraic versions of conjunction, disjunction, and universal and existential quantification;
— *distributivity* ensures that fresh-limits and fresh-colimits interact sensibly—generalising the distributivity we expect of Boolean algebras;
— being complemented gives us negation;
— the compatible σ-algebra structure gives us a 'substitution action'; and
— equality gives us equality.

We do not actually need to insist that a FOLeq algebra be finitely fresh-cocomplete, because we have complements. However, we still need to insist on distributivity, and this axiom is easier to write down if we assume $\vee$ (no complicated nested negations). We try to optimise for readability, even where this implies a little redundancy.

However, note that Definition 4.4.3 is *not* just a direct restatement of the axioms of first-order logic. This may be hard to see because the reader will see freshness like '$a\#x$' and substitution like '$x[a{\mapsto}u]$' and these are properties that, usually, only are applicable to syntax. But now, the things we are operating on x, a, and u are abstract and need not be syntax of terms and predicates. Soon, we shall see models built out of $\mathfrak{o}$-algebras of Definition 4.4.3 that are definitely not syntax-based. It may be worth noting that in previous work we build some more σ-algebra models, quite differently, out of models of set theory [Gab09b].

In short, Definition 4.4.3 is an abstract, not a concrete, definition.

REMARK 4.4.5. Calling a FOLeq algebra an *algebra* might seem misleading. To this author 'algebra' suggests an algebraic (equational) treatment, rather than the poset style of Definition 4.4.3. However, a nominal algebraic rendering of Definition 4.4.3 is possible and is given in Subsection A.3. The interesting axioms are the ones that look like the left- and right-introduction rules for the universal quantifier; see $(\forall\vee)$ and $(\forall\leq)$ from Figure 4.

Here, we express properties of limits, instead of using nominal algebra axioms as we do in Subsection A.3. The content is the same.

$$\frac{}{\Phi, \phi \vdash \phi, \Psi}\ (\mathbf{Hyp}) \qquad \frac{}{\Phi, \bot \vdash \Psi}\ (\bot\mathbf{L}) \qquad \frac{\Phi, r{=}r \vdash \Psi}{\Phi \vdash \Psi}\ (=\mathbf{R})$$

$$\frac{\Phi, \phi_1, \phi_2 \vdash \Psi}{\Phi, \phi_1 \wedge \phi_2 \vdash \Psi}\ (\wedge\mathbf{L}) \qquad \frac{\Phi \vdash \psi_1, \Psi \quad \Phi \vdash \psi_2, \Psi}{\Phi \vdash \psi_1 \wedge \psi_2, \Psi}\ (\wedge\mathbf{R}) \qquad \frac{\Phi, r'{=}r, \phi[a{\mapsto}r'] \vdash \Phi}{\Phi, r'{=}r, \phi[a{\mapsto}r] \vdash \Phi}\ (=\mathbf{L})$$

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

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

Fig. 2: Derivation rules of first-order logic with equality (FOLeq)

5. INTERPRETATION OF FIRST-ORDER LOGIC IN A FOLEQ ALGEBRA

5.1. Syntax and derivability of first-order logic

DEFINITION 5.1.1. A **signature** is a tuple (Σ, Π, ar) of disjoint sets of

— **function symbols** $\mathsf{f} \in \Sigma$ and
— **predicate symbols** $\mathsf{P} \in \Pi$

to each of which is associated an **arity** $ar(\mathsf{f}), ar(\mathsf{P}) \in \{0, 1, 2, \dots\}$.

Given a signature, **terms** and **predicates** are defined inductively by:

$$r ::= a \mid \mathsf{f}(r_1, \dots, r_{ar(\mathsf{f})})$$
$$\phi ::= \bot \mid r{=}r \mid \mathsf{P}(r_1, \dots, r_{ar(\mathsf{P})}) \mid \phi \wedge \phi \mid \neg\phi \mid \forall a.\phi$$

Above, a ranges over atoms—atoms play the role of variable symbols in this syntax—and f ranges over elements of Σ, and P ranges over elements of Π.

We take predicates up to α-equivalence as standard, and define **free atoms** $fa(r), fa(\phi)$ and **capture-avoiding substitution** $r[a{\mapsto}s]$ and $\phi[a{\mapsto}s]$ also as standard.[18]

NOTATION 5.1.2. Since our logic is classical, we use the following standard abbreviations:

— Write $\phi \vee \psi$ for $\neg((\neg\phi) \wedge (\neg\psi))$.
— Write $\phi \Rightarrow \psi$ for $(\neg\phi) \vee \psi$.
— Write $\phi \Leftrightarrow \psi$ for $(\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$.

NOTATION 5.1.3. Φ and Ψ will range over finite sets of predicates.

We may write ϕ, Φ for $\{\phi\} \cup \Phi$ and we may write $fa(\Phi)$ for $\bigcup_{\phi \in \Phi} fa(\phi)$.

The derivation rules of first-order logic are as standard:

DEFINITION 5.1.4. A **sequent** is a pair of finite sets of predicates $\Phi \vdash \Psi$. The **derivable sequents** are defined by the rules in Figure 2.

5.2. Sound interpretation in a FOLeq algebra

Definition 5.2.1 elaborates and specialises Subsection 2.2.3 (tensor product):

DEFINITION 5.2.1. Define $\otimes^n \mathbb{A}$ to be the set of n-tuples of distinct atoms with the pointwise action $\pi{\cdot}(a_1, \dots, a_n) = (\pi(a_1), \dots, \pi(a_n))$.

[18] The author is using nominal abstract syntax to do this [GP01; Gab11]. A treatment of this with all definitions and a full proof of equivalence with 'ordinary' syntax is in [Gab11, Section 5].

$$\begin{array}{rcl@{\qquad}rcl}
[\![a]\!] & = & \mathsf{atm}_{\mathcal{U}}(a) & [\![\bot]\!] & = & \bot \\
[\![\mathsf{f}(r_1,\ldots,r_n)]\!] & = & \mathsf{f}^{\mathcal{J}}([\![r_1]\!],\ldots,[\![r_n]\!]) & [\![r{=}s]\!] & = & [\![r]\!]{=^{\mathcal{J}}}[\![s]\!] \\
 & & & [\![\mathsf{P}(r_1,\ldots,r_n)]\!] & = & \mathsf{P}^{\mathcal{J}}([\![r_1]\!],\ldots,[\![r_n]\!]) \\
 & & & [\![\phi \mathsf{\wedge} \psi]\!] & = & [\![\phi]\!] \wedge [\![\psi]\!] \\
 & & & [\![\neg\phi]\!] & = & \neg[\![\phi]\!] \\
 & & & [\![\forall a.\phi]\!] & = & \bigwedge\nolimits^{\#a}[\![\phi]\!]
\end{array}$$

Fig. 3: The interpretation of terms and predicates

Definition 5.2.2. Suppose $\mathcal{L}$ is a FOLeq algebra over $\mathcal{U}$ (from Definition 4.4.3; so $\mathcal{L}$ is a finitely fresh-complete complemented nominal poset, with a compatible σ-algebra structure over $\mathcal{U}$, and with equality $(a{=^{\mathcal{L}}}b)$). We may write $\mathcal{L}$ also for $(|\mathcal{L}|,\cdot)$ the underlying nominal set of the FOLeq algebra.

An **interpretation** $\mathcal{J}$ of a signature (Σ,Π,ar) over $\mathcal{L}$ is an assignment

— to each $\mathsf{f}\in\Sigma$ with arity n, an equivariant function $\mathsf{f}^{\mathcal{J}}:\otimes^n\mathbb{A}{\Rightarrow}\mathcal{U}$,
— to each $\mathsf{P}\in\Pi$ with arity n, an equivariant function $\mathsf{P}^{\mathcal{J}}:\otimes^n\mathbb{A}{\Rightarrow}\mathcal{L}$.

(It is not hard to check from Subsection 4.3 that the equality $a{=^{\mathcal{L}}}b$ of $\mathcal{L}$ also gives us an equivariant function $\otimes^2\mathbb{A}{\Rightarrow}\mathcal{L}$, so equality could have been presented as one of the P in the signature, but we took it to be primitive to $\mathcal{L}$ instead. The treatments of equality and the P flow together in Definition 5.2.5.)

Remark 5.2.3. By Definition 2.1.6, $\mathsf{f}^{\mathcal{J}}$ equivariant means $\mathsf{f}^{\mathcal{J}}(\pi(a_1),\ldots,\pi(a_n))=\pi{\cdot}\mathsf{f}^{\mathcal{J}}(a_1,\ldots,a_n)$. Because of this, the information we need to calculate the output of $\mathsf{f}^{\mathcal{J}}$ for any input, is contained in its output for any one list of n distinct atoms. Similarly for $\mathsf{P}^{\mathcal{J}}$. This can be viewed as a generalisation of the N quantifier and has a surprisingly attractive general theory; see the *abstractive functions* of [Gab07].

In [Gab07] we considered generalisations of nominal sets to other permutation groups, whereas in this paper we are specifically interested in the monoidal case of a σ-action (the simultaneous σ-actions form a monoid). Or, to put this more simply: in this paper we can substitute for atoms and not just permute them. This leads us to Definition 5.2.5, for which we first need a little notation:

Notation 5.2.4. If $\mathcal{U}$ is a nominal set write $\mathcal{U}^n$ for $\overbrace{\mathcal{U}\times\cdots\times\mathcal{U}}^{n\text{ times}}$.

Definition 5.2.5. Extend $\mathsf{f}^{\mathcal{J}}$ and $\mathsf{P}^{\mathcal{J}}$ and $=^{\mathcal{L}}$ to functions $\mathsf{f}^{\mathcal{J}}:\mathcal{U}^n{\Rightarrow}\mathcal{U}$ and $\mathsf{P}^{\mathcal{J}}:\mathcal{U}^n{\Rightarrow}\mathcal{L}$ and $=^{\mathcal{J}}\!:\mathcal{U}^2{\Rightarrow}\mathcal{L}$ as follows

$$\begin{array}{rcl}
\mathsf{f}^{\mathcal{J}}(u_1,\ldots,u_n) & = & \mathsf{f}^{\mathcal{J}}(a_1,\ldots,a_n)[a_1{\mapsto}u_1,\ldots,a_n{\mapsto}u_n] \\
\mathsf{P}^{\mathcal{J}}(u_1,\ldots,u_n) & = & \mathsf{P}^{\mathcal{J}}(a_1,\ldots,a_n)[a_1{\mapsto}u_1,\ldots,a_n{\mapsto}u_n] \\
{=^{\mathcal{J}}}(u_1,u_2) & = & (a{=^{\mathcal{L}}}b)[a{\mapsto}u_1,b{\mapsto}u_2]
\end{array}$$

where we choose $a_1,\ldots,a_n$ to be n distinct atoms fresh for $\bigcup_1^n supp(u_i)$. We may write ${=^{\mathcal{J}}}(u_1,u_2)$ infix as $u_1{=^{\mathcal{J}}}u_2$.

Lemma 5.2.6. *Definition 5.2.5 is independent of the choice of fresh atoms $a_1,\ldots,a_n$ and a,b.*

Proof. A proof by direct calculations is not hard, or we can see this directly from [Gab07, Theorem 27] since by Lemma 3.1.8 $\mathsf{f}^{\mathcal{J}}$ and $\mathsf{P}^{\mathcal{J}}$ are *purely abstractive* (terminology of Theorem 27). □

Definition 5.2.7. Extend the interpretation $\mathcal{J}$ of Definition 5.2.5 to terms and predicates as shown in Figure 3.

Much as for $\mathsf{f}^{\mathcal{J}}$ and $\mathsf{P}^{\mathcal{J}}$ above, we technically need to check that Definition 5.2.7 does not depend on the choice of bound atom a in $\forall a.\phi$. This follows by easy calculations using the fact that by assumption, $a\notin supp(\bigwedge^{\#a}[\![\phi]\!])$.

Lemma 5.2.8. *$supp([\![r]\!])\subseteq fa(r)$ and $supp([\![\phi]\!])\subseteq fa(\phi)$.*

Proof. From Theorem 2.3.2 using the fact that the support of the abstract syntax of r is equal to $fa(r)$ and the support of the (nominal, since we have a binder $\forall$) abstract syntax of ϕ is equal to $fa(\phi)$ (see [Gab11, Section 5] for a detailed treatment of the nominal abstract syntax of the untyped λ-calculus). □

LEMMA 5.2.9. $[\![\mathsf{T}]\!] = \top \in |\mathcal{L}|$.

Proof. Unpacking Definition 5.2.7 and recalling that T is sugar for $\neg\bot$. □

LEMMA 5.2.10. *(1)* $\mathsf{f}^{\mathcal{I}}(u_1,\ldots,u_n)[a{:=}u] = \mathsf{f}^{\mathcal{I}}(u_1[a{:=}u],\ldots,u_n[a{:=}u])$.
(2) $(u_1 =^{\mathcal{I}} u_2)[a{:=}u] = (u_1[a{:=}u] =^{\mathcal{I}} u_2[a{:=}u])$.
(3) $\mathsf{P}^{\mathcal{I}}(u_1,\ldots,u_n)[a{:=}u] = \mathsf{P}^{\mathcal{I}}(u_1[a{:=}u],\ldots,u_n[a{:=}u])$.

Proof. By definition $\mathsf{f}^{\mathcal{I}}(u_1,\ldots,u_n)[a{:=}u]$ is equal to $\mathsf{f}^{\mathcal{I}}(a_1,\ldots,a_n)[a_1{:=}u_1]\ldots[a_n{:=}u_n][a{:=}u]$ where we are free to choose $a_1,\ldots,a_n$ fresh. We use Lemma 3.4.5 to rewrite this as

$$\mathsf{f}^{\mathcal{I}}(a_1,\ldots,a_n)[a{:=}u][a_1{:=}u_1[a{:=}u]]\ldots[a_n{:=}u_n[a{:=}u]].$$

By Theorem 2.3.2 $a\#\mathsf{f}^{\mathcal{I}}(a_1,\ldots,a_n)$ so by $(\sigma\#)$ we can garbage–collect the innermost $[a{:=}u]$. The result follows.

The cases of $=^{\mathcal{I}}$ and $\mathsf{P}^{\mathcal{I}}$ are similar. □

LEMMA 5.2.11. — $[\![t[a{:=}r]]\!] = [\![t]\!][a{:=}[\![r]\!]]$.
— $[\![\phi[a{:=}r]]\!] = [\![\phi]\!][a{:=}[\![r]\!]]$.

Proof. By a routine induction on t and ϕ.

— *The case of* a. $a[a{:=}r] = r$ and by $(\sigma\mathbf{a})$ $\mathsf{atm}_{\mathcal{U}}(a)[a{:=}[\![r]\!]] = [\![r]\!]$.
— *The cases of* $\mathsf{f}(r_1,\ldots,r_n)$, $\mathsf{P}(r_1,\ldots,r_n)$, *and* $r_1{=}r_2$. This is Lemma 5.2.10.
— *The case of* $\bot$. By Theorem 2.3.2 $supp(\bot) = \varnothing$. By $(\sigma\#)$ $\bot[a{:=}x] = \bot$.
— *The cases of* $\phi\wedge\psi$ *and* $\forall a.\phi$. From parts 1 and 3 of Lemma 4.2.3.
— *The case of* $\neg\phi$. Direct from our assumption that the σ-algebra structure is compatible (Definition 4.2.1).

□

DEFINITION 5.2.12. Extend the interpretation $\mathcal{I}$ of Definition 5.2.7 to sequents by defining

$$[\![\Phi \vdash \Psi]\!] \quad \text{is the assertion} \quad \bigwedge\{[\![\phi]\!] \mid \phi\in\Phi\} \leq \bigvee\{[\![\psi]\!] \mid \psi \in \Psi\}.$$

THEOREM 5.2.13 (Soundness). *If* $\Phi \vdash \Psi$ *is derivable then* $[\![\Phi \vdash \Psi]\!]$ *is true.*

Proof. It suffices to check validity of the rules in Figure 2.

— $(\bot\mathbf{L})$ is valid because $[\![\bot]\!] = \bot$ in Definition 5.2.7.
— $(=\mathbf{R})$ is valid from condition 1 of Definition 4.3.1.
— $(=\mathbf{L})$ is valid from condition 2 of Definition 4.3.1.
— $(\wedge\mathbf{L})$, $(\wedge\mathbf{R})$, $(\neg\mathbf{L})$, and $(\neg\mathbf{R})$ are by standard facts of Boolean algebras (bounded distributive complemented lattices).
— $(\forall\mathbf{R})$ and $(\forall\mathbf{L})$ are from Lemma 4.2.4 and Lemma 5.2.8. □

6. THE σ-POWERSET AS A FOLEQ ALGEBRA

Suppose $\mathcal{P}$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$. We know by Proposition 3.4.10 that $Pow_\sigma(\mathcal{P})$ from Definition 3.4.6 is a σ-algebra. It is also very easy to see that $Pow_\sigma(\mathcal{P})$ with the pointwise action $\pi{\cdot}X = \{\pi{\cdot}p \mid p \in X\}$ (Definition 3.4.1), ordered by subset inclusion $X \subseteq Y$, is a nominal poset (Definition 4.1.1).

We want now to show that $Pow_\sigma(\mathcal{P})$ is a FOLeq algebra (Theorem 6.4.1). That is, we want to show that $Pow_\sigma(\mathcal{P})$

— is finitely fresh-complete (has $\bigwedge^{\#A} X$),
— is complemented ($|\mathcal{P}| \setminus X$),
— has an equality,
— and the σ-action is compatible (Definition 4.2.1).

It will be obvious that complement is given by sets complement. Equality is non-evident but very natural and is discussed in Subsection 6.3. For $\bigwedge^{\#A} X$ we use Proposition 4.1.5 to split the problem into parts. We consider this next.

6.1. Intersection, complement, top element

Suppose $\mathcal{P} = (|\mathcal{P}|, \cdot, \mathcal{U}, \mathsf{amgis}_{\mathcal{P}})$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$. Recall the definitions of $NomPow(\mathcal{P})$ from Subsection 2.4.1 (finitely-supported powerset) and of $Pow_\sigma(\mathcal{P})$ from Definition 3.4.6 (the σ-powerset).

Lemmas 6.1.1 and 6.1.2 are perhaps not entirely obvious, but all the cases follow the same pattern of pointwise calculations on sets. These cover the technically simplest cases; in Subsection 6.2 we move on to quantification.

LEMMA 6.1.1. *Suppose $\mathcal{P}$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$. Suppose $u{\in}|\mathcal{U}|$ and $Y, Y'{\in}|NomPow(\mathcal{P})|$ and suppose $\mathcal{X} \subseteq |NomPow(\mathcal{P})|$ is strictly finitely supported (Definition 2.4.3). Then:*

(1) $(|\mathcal{P}|\setminus Y)[a{\mapsto}u] = |\mathcal{P}| \setminus (Y[a{\mapsto}u])$.
In words: σ commutes with sets complement.
(2) $(\bigcap_{X\in\mathcal{X}} X)[a{\mapsto}u] = \bigcap_{X\in\mathcal{X}}(X[a{\mapsto}u])$ *and* $(\bigcup_{X\in\mathcal{X}} X)[a{\mapsto}u] = \bigcup_{X\in\mathcal{X}}(X[a{\mapsto}u])$.
In words: σ commutes with strictly finitely supported sets intersections and unions.
In particular, σ commutes with finite *intersection and union.*
(3) *If $Y \subseteq Y'$ then $Y[a{\mapsto}u] \subseteq Y'[a{\mapsto}u]$.*
In words: σ is monotone.

Proof. We reason as follows:

$$\begin{array}{rll}
p \in (|\mathcal{P}|\setminus X)[a{\mapsto}u] & \Leftrightarrow \mathit{И}c.p[u{\leftrightarrow}c] \in (c\ a){\cdot}(|\mathcal{P}|\setminus X) & \text{Proposition 3.4.2}\\
& \Leftrightarrow \mathit{И}c.p[u{\leftrightarrow}c] \in |\mathcal{P}|\setminus(c\ a){\cdot}X & \text{Theorem 2.3.2}\\
& \Leftrightarrow \mathit{И}c.p[u{\leftrightarrow}c] \notin (c\ a){\cdot}X & \text{Fact}\\
& \Leftrightarrow p \notin X[a{\mapsto}u] & \text{Proposition 3.4.2}\\
& \Leftrightarrow p \in |\mathcal{P}|\setminus(X[a{\mapsto}u]) & \text{Fact}\\
p \in (\bigcap_{X\in\mathcal{X}} X)[a{\mapsto}u] & \Leftrightarrow \mathit{И}c.p[u{\leftrightarrow}c] \in \bigcap_{X\in\mathcal{X}}(c\ a){\cdot}X & \text{Prop 3.4.2, Thm 2.3.2}\\
& \Leftrightarrow \mathit{И}c.\forall X{\in}\mathcal{X}.p[u{\leftrightarrow}c] \in (c\ a){\cdot}X & \text{Fact}\\
p \in \bigcap_{X\in\mathcal{X}}(X[a{\mapsto}u]) & \Leftrightarrow \forall X{\in}\mathcal{X}.p \in X[a{\mapsto}u] & \text{Fact}\\
& \Leftrightarrow \forall X{\in}\mathcal{X}.\mathit{И}c.p[u{\leftrightarrow}c] \in (c\ a){\cdot}X & \text{Proposition 3.4.2}
\end{array}$$

We note that by Lemma 2.4.4, $c\#\mathcal{X}$ if and only if $c\#X$ for every $X \in \mathcal{X}$. This allows us to swap the $\forall$ and the $\mathit{И}$ quantifier, and the result follows.

Part 3 follows using part 2 and the fact that $Y[a{\mapsto}u] \subseteq Y'[a{\mapsto}u]$ if and only if $Y[a{\mapsto}u] \cap Y'[a{\mapsto}u] = Y[a{\mapsto}u]$. □

LEMMA 6.1.2. — *If* $X, Y \in |NomPow(\mathcal{P})|$ *then* $X \cap Y \in |NomPow(\mathcal{P})|$ *and if* $X, Y \in |Pow_\sigma(\mathcal{P})|$ *then* $X \cap Y \in |Pow_\sigma(\mathcal{P})|$.
— *If* $X \in |NomPow(\mathcal{P})|$ *then* $|\mathcal{P}| \backslash X \in |NomPow(\mathcal{P})|$, *and if* $X \in |Pow_\sigma(\mathcal{P})|$ *then* $|\mathcal{P}| \backslash X \in |Pow_\sigma(\mathcal{P})|$.
— $|\mathcal{P}| \in |NomPow(\mathcal{P})|$ *and* $|\mathcal{P}| \in |Pow_\sigma(\mathcal{P})|$.

Proof. Finite support follows from Theorem 2.3.2. All the results involving finitely-supported powerset $NomPow(\text{-})$ follow immediately.

It remains to check conditions 1 and 2 of Definition 3.4.6 for the σ-powerset $Pow_\sigma(\text{-})$. We do this just for $X \cap Y$:

(1) *If* a *is fresh (so* $a\#X, Y, u$*) then* $(X \cap Y)[a \mapsto u] = X \cap Y$.
We reason as follows:

$$\begin{array}{rll} (X \cap Y)[a \mapsto u] & = (X[a \mapsto b]) \cap (Y[a \mapsto u]) & \text{Lemma 6.1.1} \\ & = X \cap Y & \text{Cond 1 of Def 3.4.6} \end{array}$$

(2) *If* b *is fresh (so* $b\#X, Y$*) then* $(X \cap Y)[a \mapsto b] = (b\ a) \cdot (X \cap Y)$.
We reason as follows:

$$\begin{array}{rll} (X \cap Y)[a \mapsto b] & = (X[a \mapsto b]) \cap (Y[a \mapsto b]) & \text{Lemma 6.1.1} \\ & = ((b\ a) \cdot X) \cap ((b\ a) \cdot Y) & \text{Cond 2 of Def 3.4.6} \\ & = (b\ a) \cdot (X \cap Y) & \text{Theorem 2.3.2} \end{array}$$

□

PROPOSITION 6.1.3. $\varnothing$ *is the bottom element and* $|\mathcal{P}|$ *is the top element in* $NomPow(\mathcal{P})$ *and* $Pow_\sigma(\mathcal{P})$ *ordered by subset inclusion.* $X \cap Y$ *is the greatest lower bound for* $\{X, Y\}$. $|\mathcal{P}| \backslash X$ *is the complement of* X. *Thus* $NomPow(\mathcal{P})$ *and* $Pow_\sigma(\mathcal{P})$ *ordered by subset inclusion form Boolean algebras, and by construction* $Pow_\sigma(\mathcal{P})$ *is a subalgebra of* $NomPow(\mathcal{P})$.

Proof. By standard sets calculations. □

6.2. Quantification

In this subsection we explore what concrete sets operation corresponds to the fresh-finite limits $\bigwedge^{\#a}$ of Notation 4.1.3; in other words we explore quantification.

Suppose $\mathcal{P} = (|\mathcal{P}|, \cdot, \mathcal{U}, \mathsf{amgis}_{\mathcal{P}})$ is an $\mathfrak{o}$-algebra over a termlike σ-algebra $\mathcal{U}$. Recall the definitions of $NomPow(\mathcal{P})$ from Subsection 2.4.1 (finitely-supported powerset) and of $Pow_\sigma(\mathcal{P})$ from Definition 3.4.6 (the σ-powerset).

DEFINITION 6.2.1. If $X \in |NomPow(\mathcal{P})|$ then define

$$\bigcap^{\#a} X = \bigcap \{X[a \mapsto u] \mid u \in |\mathcal{U}|\}.$$

We work towards proving $\bigcap^{\#a} X \in |NomPow(\mathcal{P})|$; this is Theorem 6.2.5. We need the key technical Lemma 6.2.2 for Proposition 6.2.3:

LEMMA 6.2.2. *Suppose* $p \in |\mathcal{P}|$ *and* $X \in NomPow(\mathcal{P})$ *and* $v \in |\mathcal{U}|$ *and suppose* $a\#v$. *Then*

$$\begin{array}{l} И b'.\forall u \in |\mathcal{U}|.И a'.p[v \leftrightarrow b'][u \leftrightarrow a'] \in (b'\ b) \cdot (a'\ a) \cdot X \quad \textit{if and only if} \\ \forall u \in |\mathcal{U}|.И b'.И a'.p[u \leftrightarrow a'][v \leftrightarrow b'] \in (b'\ b) \cdot (a'\ a) \cdot X. \end{array}$$

Proof. We prove two implications:

— *The up-down implication.* Assume $И b'.\forall u \in |\mathcal{U}|.И a'.p[v \leftrightarrow b'][u \leftrightarrow a'] \in (b'\ b) \cdot (a'\ a) \cdot X$.
Choose $u \in |\mathcal{U}|$. Choose fresh b' and a' (so $b', a'\#X, v, u$). Then by assumption (since $b'\#X, v$ and $a'\#X, v, u$) $p[v \leftrightarrow b'][u \leftrightarrow a'] \in (b'\ b) \cdot (a'\ a) \cdot X$ so that by $(\mathfrak{o}\sigma)$ of Figure 1 (since $a'\#v$) $p[u[b' \mapsto v] \leftrightarrow a'][v \leftrightarrow b'] \in (b'\ b) \cdot (a'\ a) \cdot X$. Now by $(\sigma\#)$ $u[b' \mapsto v] = u$ (since $b'\#u$). Therefore $p[u \leftrightarrow a'][v \leftrightarrow b'] \in (b'\ b) \cdot (a'\ a) \cdot X$.
— *The down-up implication.* Assume $\forall u \in |\mathcal{U}|.И b'.И a'.p[u \leftrightarrow a'][v \leftrightarrow b'] \in (b'\ b) \cdot (a'\ a) \cdot X$.

Choose fresh b' (so $b'\#X, v$), choose $u{\in}|\mathcal{U}|$ (for which b need not necessarily be fresh), and choose fresh a' (so $a'\#X, v, u$). By Lemma 3.1.8 $b'\#u[b'{\mapsto}v]$ (since $b'\#v$). Therefore $p[u[b'{\mapsto}v]{\leftrightarrow}a'][v{\leftrightarrow}b'] \in (b'\ b){\cdot}(a'\ a){\cdot}X$ and by $(\circlearrowright\sigma)$ of Figure 1 (since $a'\#v$) $p[v{\leftrightarrow}b'][u{\leftrightarrow}a'] \in (b'\ b){\cdot}(a'\ a){\cdot}X$. □

We cannot use Lemma 6.1.1(2) to derive Proposition 6.2.3 because $\{X[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\}$ is not necessarily strictly finitely supported. The result still holds, by a proof using Lemma 6.2.2:

PROPOSITION 6.2.3. *Suppose* $X{\in}|NomPow(\mathcal{P})|$ *and* $v{\in}|\mathcal{U}|$ *and* $a\#v$. *Then*

$$(\bigcap\nolimits^{\#a} X)[b{\mapsto}v] = \bigcap\nolimits^{\#a}(X[b{\mapsto}v]).$$

Proof. Consider $p{\in}|\mathcal{P}|$. We reason as follows:

$$\begin{array}{lll}
p \in (\bigcap^{\#a} X)[b{\mapsto}v] & & \\
\quad \Leftrightarrow \mathit{И}b'.p[v{\leftrightarrow}b'] \in \bigcap^{\#a}(b'\ b){\cdot}X & & \text{Prop 3.4.2, Thrm 2.3.2} \\
\quad \Leftrightarrow \mathit{И}b'.p[v{\leftrightarrow}b'] \in \bigcap_{u\in|\mathcal{U}|}((b'\ b){\cdot}X)[a{\mapsto}u] & & \text{Definition 6.2.1} \\
\quad \Leftrightarrow \mathit{И}b'.\forall u{\in}|\mathcal{U}|.\mathit{И}a'.p[v{\leftrightarrow}b'][u{\leftrightarrow}a'] \in (b'\ b){\cdot}(a'\ a){\cdot}X & & \text{Proposition 3.4.2} \\
\quad \Leftrightarrow \forall u{\in}|\mathcal{U}|.\mathit{И}b'.\mathit{И}a'.p[u{\leftrightarrow}a'][v{\leftrightarrow}b'] \in (b'\ b){\cdot}(a'\ a){\cdot}X & & \text{Lemma 6.2.2} \\
\quad \Leftrightarrow \forall u{\in}|\mathcal{U}|.\mathit{И}a'.p[u{\leftrightarrow}a'] \in ((a'\ a){\cdot}X)[b{\mapsto}v] & & \text{Proposition 3.4.2} \\
\quad \Leftrightarrow \forall u{\in}|\mathcal{U}|.\mathit{И}a'.p[u{\leftrightarrow}a'] \in (a'\ a){\cdot}(X[b{\mapsto}v]) & & \text{T2.3.2, C2.1.12, } a', a\#v \\
\quad \Leftrightarrow \forall u{\in}|\mathcal{U}|.p \in X[b{\mapsto}v][a{\mapsto}u] & & \text{Proposition 3.4.2} \\
\quad \Leftrightarrow p \in \bigcap^{\#a}(X[b{\mapsto}v]) & & \text{Definition 6.2.1}
\end{array}$$ □

LEMMA 6.2.4. *Suppose* $X \in |NomPow(\mathcal{P})|$. *Then*

$$b\#X \quad \textit{implies} \quad \bigcap^{\#a} X = \bigcap^{\#b}(b\ a){\cdot}X.$$

As a corollary, $a\#\bigcap^{\#a} X$ *and* $supp(\bigcap^{\#a} X) \subseteq supp(X)\backslash\{a\}$.

Proof. The corollary follows by part 3 of Corollary 2.1.12 and by Theorem 2.3.2. For the first part, we reason as follows:

$$\begin{array}{lll}
\bigcap^{\#a} X = \bigcap\{X[a{\mapsto}u] \mid u{\in}|\mathcal{U}|\} & & \text{Definition 6.2.1} \\
\quad = \bigcap\{((b\ a){\cdot}X)[b{\mapsto}u] \mid u{\in}|\mathcal{U}|\} & & \text{Lemma 3.4.4} \\
\quad = \bigcap^{\#b}(b\ a){\cdot}X & & \text{Definition 6.2.1}
\end{array}$$ □

Recall $Pow_\sigma(\mathcal{P})$ from Definition 3.4.6:

THEOREM 6.2.5. *If* $X{\in}|Pow_\sigma(\mathcal{P})|$ *then* $\bigcap^{\#a} X{\in}|Pow_\sigma(\mathcal{P})|$.

Proof. Finite support of $\bigcap^{\#a} X$ is by Theorem 2.3.2. It remains to check conditions 1 and 2 of Definition 3.4.6:

(1) Suppose b is fresh (so $b\#X$) and suppose $v{\in}|\mathcal{U}|$. Using Lemma 6.2.4 suppose without loss of generality that $a\#v$. Then we reason as follows:

$$\begin{array}{lll}
(\bigcap^{\#a} X)[b{\mapsto}v] = \bigcap^{\#a}(X[b{\mapsto}v]) & & \text{Proposition 6.2.3} \\
\quad = \bigcap^{\#a} X & & \text{Cond 1 of Def 3.4.6, } b\#X
\end{array}$$

(2) Suppose b' is fresh (so $b'\#X$). Then we reason as follows:

$$\begin{array}{lll}
(\bigcap^{\#a} X)[b{\mapsto}b'] = \bigcap^{\#a}(X[b{\mapsto}b']) & & \text{Proposition 6.2.3} \\
\quad = \bigcap^{\#a}((b'\ b){\cdot}X) & & \text{Cond 2 of Def 3.4.6, } b'\#X \\
\quad = (b'\ b){\cdot}(\bigcap^{\#a} X) & & \text{Theorem 2.3.2}
\end{array}$$ □

COROLLARY 6.2.6. *If* $X{\in}|Pow_\sigma(\mathcal{P})|$ *then* $\bigcap^{\#a} X$ *from Definition 6.2.1 is equal to* $\bigwedge^{\#a} X$ *the fresh-finite limit in the lattice* $Pow_\sigma(\mathcal{P})$ *ordered by subset inclusion.*

Proof. By Theorem 6.2.5 $\bigcap^{\#a} X \in |Pow_\sigma(\mathcal{P})|$. We use Proposition 4.2.6. □

REMARK 6.2.7. The proofs leading up to Corollary 6.2.6 are somewhat subtle, and it is useful to illustrate how they work by tracing through how a similar 'proof' fails:

By Proposition 4.2.8 $\bigwedge^{\#a} X = \bigwedge_{n\in\mathbb{A}} X[a{\mapsto}n]$. However it does *not* follow from this that $\bigcap^{\#a} X = \bigcap_{n\in\mathbb{A}} X[a{\mapsto}n]$. This is because Theorem 6.2.5 does not hold of $\bigcap_{n\in\mathbb{A}} X[a{\mapsto}n]$—it is not in $Pow_\sigma(\mathcal{P})$—and this is because Proposition 6.2.3 fails, which is because Lemma 6.2.2 fails.

That is, if we replace every $\forall u{\in}|\mathcal{U}|$ in Lemma 6.2.2 with $\forall n{\in}\mathbb{A}$, then Lemma 6.2.2 no longer works (and this is because $\mathbb{A}$ is not closed under applying substitutions $[a{\mapsto}u]$, whereas $\mathcal{U}$ is).

6.3. Equality in the σ-powerset

We have seen how to interpret conjunction, union, negation, truth, false, and quantification in the σ-powerset of an $\mathfrak{o}$-algebra. Now we show that we can also interpret equality.

Suppose $\mathcal{P} = (|\mathcal{P}|, \cdot, \mathcal{U}, \mathsf{amgis}_{\mathcal{P}})$ is an exact $\mathfrak{o}$-algebra (Definition 3.3.2) over a termlike σ-algebra $\mathcal{U}$.

DEFINITION 6.3.1. If $u, v{\in}|\mathcal{U}|$ then define $(u{=}^{\mathcal{P}}v) \subseteq |\mathcal{P}|$ by

$$(u{=}^{\mathcal{P}}v) \;=\; \{p{\in}|\mathcal{P}| \mid И c.(p[u{\leftarrow}c] = p[v{\leftarrow}c])\}.$$

REMARK 6.3.2. Some motivation for Definition 6.3.1: Suppose $\mathcal{X}$ is a σ-algebra over $\mathcal{U}$ (Definition 3.1.5) and suppose just for this remark that $\mathcal{P} = Pow_{\mathfrak{o}}(\mathcal{X})$ (Definition 3.2.4). Recall from Proposition 3.3.3 that $\mathcal{P}$ is indeed exact.

Suppose $p \in (u{=}^{\mathcal{P}}v)$, meaning by Definition 6.3.1 that $И c.(p[u{\leftarrow}c] = p[v{\leftarrow}c])$. Using Proposition 3.2.2 $p[u{\leftarrow}c] = p[v{\leftarrow}c]$ when $x[c{\mapsto}u] \in p \Leftrightarrow x[c{\mapsto}v] \in p$ for every $x{\in}|\mathcal{X}|$. Thus

$$p \in (u{=}^{\mathcal{P}}v) \quad\Leftrightarrow\quad И c.\forall x{\in}\mathcal{X}.(x[c{\mapsto}u] \in p \Leftrightarrow x[c{\mapsto}v] \in p).$$

Intuitively this means p cannot discern any difference between u and v. This is in the spirit of *Leibnitz equality*; two things that equal when they are indiscernible.

More discussion is in Appendix B.

REMARK 6.3.3. It is not necessarily the case that $supp(u){\cup}supp(v) \subseteq supp(u{=}^{\mathcal{P}}v)$. For instance, if $\mathcal{P}$ has the *trivial* $\mathfrak{o}$-action that $p[u{\leftarrow}a] = p$ for all u and a (for instance, take $|\mathcal{P}| = \{*\}$ the one-element set) then $(a{=}^{\mathcal{P}}b) = |\mathcal{P}|$, which has support $\varnothing$.

Lemma 6.3.4 is where we really use the exactness property of $\mathcal{P}$ (Definition 3.3.2):

LEMMA 6.3.4. $(u_1{=}^{\mathcal{P}}u_2)[a{\mapsto}v] = ((u_1[a{\mapsto}v]){=}^{\mathcal{P}}(u_2[a{\mapsto}v]))$.

Proof. We reason as follows, where we write $u_1' = (a'\ a){\cdot}u_1$ and $u_2' = (a'\ a){\cdot}u_2$ for brevity:

$$\begin{array}{lll}
p{\in}(u_1{=}^{\mathcal{P}}u_2)[a{\mapsto}v] & & \\
\Leftrightarrow И a'.\, p[v{\leftarrow}a'] \in (u_1'{=}^{\mathcal{P}}u_2') & & \text{Prop 3.4.2} \\
\Leftrightarrow И a', c.\, p[v{\leftarrow}a'][u_1'{\leftarrow}c] = p[v{\leftarrow}a'][u_2'{\leftarrow}c] & & \text{Def 6.3.1} \\
\Leftrightarrow И a', c.\, p[u_1'[a'{\mapsto}v]{\leftarrow}c][v{\leftarrow}a'] = p[u_2'[a'{\mapsto}v]{\leftarrow}c][v{\leftarrow}a'] & & (\mathfrak{o}\sigma),\ c\#v \\
\Leftrightarrow И a', c.\, p[u_1'[a'{\mapsto}v]{\leftarrow}c] = p[u_2'[a'{\mapsto}v]{\leftarrow}c] & & \text{Exactness} \\
\Leftrightarrow И c.\, p[u_1[a{\mapsto}v]{\leftarrow}c] = p[u_2[a{\mapsto}v]{\leftarrow}c] & & (\sigma\alpha),\ a'\#u_1, u_2 \\
\Leftrightarrow p \in ((u_1[a{\mapsto}v]){=}^{\mathcal{P}}u_2[a{\mapsto}v]) & & \text{Def 6.3.1}
\end{array}$$

□

PROPOSITION 6.3.5. *Suppose* $u, v{\in}|\mathcal{U}|$. *Then:*

(1) *If* $a\#u, v$ *then* $(u{=}^{\mathcal{P}}v)[a{\mapsto}w] = (u{=}^{\mathcal{P}}v)$.
(2) *If* $b\#u, v$ *then* $(u{=}^{\mathcal{P}}v)[a{\mapsto}b] = (a\ b){\cdot}(u{=}^{\mathcal{P}}v)$.

As a corollary, $(u{=}^{\mathcal{P}}v){\in}|Pow_\sigma(\mathcal{P})|$.

Proof. The first part is from Lemma 6.3.4 and $(\sigma\#)$. The second part is from Lemmas 6.3.4 and 3.1.10 and Theorem 2.3.2.

For the corollary, finite support is from Theorem 2.3.2 and conditions 1 and 2 of Definition 3.4.6 are just the first two parts of this result. □

LEMMA 6.3.6. *Suppose* $u, v, w \in |\mathcal{U}|$ *and suppose* $b\#u, v$. *Then*

$$(u=^{\mathcal{P}} v)[a\mapsto w] = ((b\ a)\cdot u =^{\mathcal{P}} (b\ a)\cdot v)[b\mapsto v].$$

Proof. From Lemma 6.3.4 and $(\sigma\alpha)$ (a direct proof by Lemma 3.4.4 is also possible). □

COROLLARY 6.3.7. $p \in (u=^{\mathcal{P}} v)$ *if and only if* $p \in (a=^{\mathcal{P}} b)[a\mapsto u, b\mapsto v]$.

Proof. Using Lemma 6.3.6 we may assume without loss of generality that $a, b\#u, v$ so that $(a=^{\mathcal{P}} b)[a\mapsto u, b\mapsto v] = (a=^{\mathcal{P}} b)[a\mapsto u][b\mapsto v]$. It follows by Lemma 6.3.4 that $(a=^{\mathcal{P}} b)[a\mapsto u, b\mapsto v] = (u=^{\mathcal{P}} v)$. □

A converse to Proposition 6.3.8 is valid for a specific model; see Proposition 7.3.7.

PROPOSITION 6.3.8. *Suppose* $X \in |Pow_\sigma(\mathcal{P})|$ *and* $u, v \in |\mathcal{U}|$. *Then:*

(1) $(u=^{\mathcal{P}} u) = |\mathcal{P}|$.
(2) $(u=^{\mathcal{P}} v) \cap (X[a\mapsto u]) = (u=^{\mathcal{P}} v) \cap (X[a\mapsto v])$.

Proof. Part 1 follows from the fact that $p[u\leftarrow c] = p[u\leftarrow c]$ for every p and every c.

For part 2, using condition 2 of Definition 3.4.6 to rename if necessary, suppose $a\#u, v$. By Definition 6.3.1 $p \in (u=^{\mathcal{P}} v)$ means $\mathit{И}a.p[u\leftarrow a] = p[v\leftarrow a]$. Using Proposition 3.4.2 and part 2 of Proposition 6.3.5 and part 2 of Lemma 6.1.1 we have

$$\begin{aligned} p \in (u=^{\mathcal{P}} v) \cap (X[a\mapsto u]) &\Leftrightarrow \mathit{И}a.\big(p[u\leftarrow a] = p[v\leftarrow a] \wedge p[u\leftarrow a] \in X\big) \\ &\Leftrightarrow \mathit{И}a.\big(p[u\leftarrow a] = p[v\leftarrow a] \wedge p[v\leftarrow a] \in X\big) \\ &\Leftrightarrow p \in (u=^{\mathcal{P}} v) \cap (X[a\mapsto v]). \end{aligned}$$ □

THEOREM 6.3.9. *If* $\mathcal{P}$ *is an exact* $\mathfrak{o}$*-algebra then* $(a=^{\mathcal{P}} b)$ *from Definition 6.3.1 is an equality in* $Pow_\sigma(\mathcal{P})$ *in the sense of Definition 4.3.1.*

Proof. By Proposition 6.3.5 $(a=^{\mathcal{P}} b)$ is in $Pow_\sigma(\mathcal{P})$, and $(a=^{\mathcal{P}} b)$ is an equality by Corollary 6.3.7 and Proposition 6.3.8. □

6.4. Interpreting first-order logic in the σ-powerset

We can now prove Theorem 6.4.1:

THEOREM 6.4.1. *Suppose* $\mathcal{P}$ *is an* $\mathfrak{o}$*-algebra over a termlike* σ*-algebra* $\mathcal{U}$. *Then* $Pow_\sigma(\mathcal{P})$ *is a FOL algebra if we define the following:*

$$\begin{aligned} [\![\bot]\!] &= \varnothing \\ [\![\phi \wedge \psi]\!] &= [\![\phi]\!] \cap [\![\psi]\!] \\ [\![\neg\phi]\!] &= |\mathcal{P}| \setminus [\![\phi]\!] \\ [\![\forall a.\phi]\!] &= \bigcap\nolimits^{\#a} [\![\phi]\!] \qquad \textit{(Def 6.2.1)} \end{aligned}$$

If in addition $\mathcal{P}$ *is exact (Definition 3.3.2) then* $Pow_\sigma(\mathcal{P})$ *is a FOLeq algebra if we define the following:*

$$[\![a{=}b]\!] = \{p \mid \mathit{И}c.(p[a\leftarrow c] = p[b\leftarrow c])\}$$

Proof. — The Boolean structure ($\bot$, $\wedge$, and $\neg$) is treated in Proposition 6.1.3.
— The σ-action ($[a\mapsto u]$) is compatible by Lemma 6.1.1 and Proposition 6.2.3.
— We observe by Lemma 4.2.3(2) that the σ-action is monotone, and by Theorem 6.2.5 that $\bigcap^{\#a} X \in |Pow_\sigma(\mathcal{P})|$. We combine these observations with Definition 6.2.1 and Proposition 4.2.6 to conclude that the fresh-finite limit $\bigwedge^{\#a} X$ exists in $Pow_\sigma(\mathcal{P})$.
— Equality ($=$) is treated in Theorem 6.3.9. □

REMARK 6.4.2. For the reader's convenience we assemble the maths so far by unpacking what Definitions 5.2.2 and 5.2.7 mean:

— *Definition 5.2.2.* Given a signature (Σ, Π, ar) an interpretation maps each $\mathsf{f} \in \Sigma$ with arity n to an equivariant function $\mathsf{f}^{\mathcal{I}} : \otimes^n \mathbb{A} \Rightarrow \mathcal{U}$ and each $\mathsf{P} \in \Pi$ with arity n to an equivariant function $\mathsf{P}^{\mathcal{I}} : \otimes^n \mathbb{A} \Rightarrow Pow_\sigma(\mathcal{P})$ ($\otimes^n$ is from Definition 5.2.1).

— *Definition 5.2.7.* The interpretation extends as follows:

$$
\begin{array}{rll}
[\![a]\!] & = \mathsf{atm}_{\mathcal{U}}(a) & \text{Def 3.1.1} \\
[\![\mathsf{f}(r_1,\ldots,r_n)]\!] & = \mathsf{f}^{\mathcal{I}}(a_1,\ldots,a_n)[a_1 \mapsto [\![r_1]\!],\ldots,a_n \mapsto [\![r_n]\!]] & \text{Defs 5.2.5 \& 5.2.7} \\
[\![\bot]\!] & = \varnothing & \text{Prop 6.1.3} \\
[\![r \mathbin{=\!\!=} s]\!] & = \{p \mid \mathcal{N}c.(p[[\![r]\!] \leftarrow\!\!\rightarrow c] = p[[\![s]\!] \leftarrow\!\!\rightarrow c])\} & \text{Def 6.3.1} \\
[\![\mathsf{P}(r_1,\ldots,r_n)]\!] & = \mathsf{P}^{\mathcal{I}}(a_1,\ldots,a_n)[a_1 \mapsto [\![r_1]\!],\ldots,a_n \mapsto [\![r_n]\!]] & \text{Defs 5.2.5 \& 5.2.7} \\
[\![\phi \wedge \psi]\!] & = [\![\phi]\!] \cap [\![\psi]\!] & \text{Prop 6.1.3} \\
[\![\neg\phi]\!] & = |\mathcal{P}| \setminus [\![\phi]\!] & \text{Prop 6.1.3} \\
[\![\forall a.\phi]\!] & = \bigcap\nolimits^{\#a} [\![\phi]\!] = \bigcap_{u \in |\mathcal{U}|} [\![\phi]\!][a \mapsto u] & \text{Def 6.2.1}
\end{array}
$$

Recall also that $\mathsf{atm}_{\mathcal{U}}(a)$ comes from Definition 3.1.1, and where $\mathcal{U}$ is assumed we will usually write it just as a.

— *Theorem 5.2.13.* From Theorem 6.4.1 we conclude that if $\phi_1,\ldots,\phi_n \vdash \psi_1,\ldots,\psi_m$ then with the definitions above, $\bigcap_i [\![\phi_i]\!] \subseteq \bigcup_j [\![\psi_j]\!]$.

7. COMPLETENESS

We now set about proving Theorem 7.3.10, which states that if $[\![\Phi \vdash \Psi]\!]$ is true in every interpretation in every FOLeq algebra, then indeed $\Phi \vdash \Psi$ is derivable in first-order logic. The proof follows the general outline that the reader familiar with such proofs might expect: we build a notion of *point* out of maximal consistent sets of predicates (Definition 7.2.2), map ϕ to a set of points (Definition 7.3.1), show that every consistent ϕ is contained in some point, note that ϕ maps precisely to the set of points containing ϕ, and then use that to prove completeness.

The main results are Theorems 7.1.20, 7.3.6 and 7.3.10.

The constructions are subtle:

(1) We have more structure than usual: points form not just a set but an σ-algebra, and sets of points form not just a set but a σ-algebra, and not just a lattice but a lattice with fresh-finite limits and equality. This means more properties to verify, and so more proofs.
(2) We are used to seeing valuations being used to build the final semantics; not so here. The semantics, as discussed in the Introduction, is absolute. This requires a certain change of perspective.
(3) We cannot directly use Zorn's lemma and need to use a carefully designed increasing chain of filter-ideal pairs, and indeed, the definitions of filter and ideal also require careful design.

Detailed exposition follows below; see in particular Remarks 7.1.2, 7.1.10, and 7.1.21, and also Remark 7.3.8.

An argument is possible that the proofs that follow are the *real* proof, and the usual development is a projection of that proof to a Zermelo-Fraenkel sets (ZF) universe. This is reasonable, since ZF is consistent if and only if FM is, but the ZF construction is not optimal, since FOL syntax interrogates the models just for fresh-finite limits and equality, that is, for T, $\wedge$, $\forall a$, and $\mathbin{=\!\!=}$—which is precisely the structure that FOLeq algebras provide using nominal sets.

7.1. Filters and points

7.1.1. Filters. Recall *predicates* ϕ from Definition 5.1.1 and the *entailment* relation $\Phi \vdash \phi$ from Figure 2:

DEFINITION 7.1.1. A **filter** is a nonempty set p of predicates (which need not have finite support) such that:

(1) $\bot \notin p$ (we call p **consistent**).
(2) If $\phi \in p$ and $\phi \vdash \phi'$ then $\phi' \in p$ (we call p **deductively closed**).
(3) If $\phi \in p$ and $\phi' \in p$ then $\phi \mathsf{\wedge} \phi' \in p$.
(4) If $\mathsf{И}b.(b\ a)\cdot\phi \in p$ then $\forall a.\phi \in p$.

REMARK 7.1.2. p above need not have finite support. This means that in Definition 7.1.1 it is *not* the case that $\mathsf{И}b.(b\ a)\cdot\phi \in p$ if and only if $\neg\mathsf{И}b.(b\ a)\cdot\phi \notin p$. The 'standard' equivalent decompositions of $\mathsf{И}$ into '$\forall$+freshness' and '$\exists$+freshness' (Theorem 2.17 of [Gab14], Theorem 6.5 of [Gab11], or Theorem 9.4.6 of [Gab01]) might not work with respect to p. However, x in Definition 7.1.1 is still assumed to have finite support, and that gives us all we need.

This is why $\mathfrak{o}$-algebras (Definition 3.1.6) are not assumed to be nominal sets, but are only assumed to be sets with a permutation action—and why σ-algebras (Definition 3.1.5) are assumed to be nominal sets.

More discussion of the support of p in Remark 7.1.10.

Ideals are dual to filters; Definition 7.1.3 is standard:

DEFINITION 7.1.3. An **ideal** is a nonempty set Z of predicates (which need not have finite support) such that:

(1) $\top \notin Z$.
(2) If $\psi \in Z$ and $\psi' \leq \psi$ then $\psi' \in Z$ (we call Z **down-closed**).
(3) If $\psi \in Z$ and $\psi' \in Z$ then $\psi \mathsf{\vee} \psi' \in Z$.

REMARK 7.1.4. Definition 7.1.3 is not a perfect dual to Definition 7.1.1: we do not have a fourth condition corresponding to condition 4 of Definition 7.1.1.

This is deliberate and will be important; see Remark 7.1.21.

DEFINITION 7.1.5. If ϕ is a predicate then define $\phi{\uparrow}$ and $\phi{\downarrow}$ by

$$\phi{\uparrow} = \{\xi \mid \phi \vdash \xi\} \qquad \text{and} \qquad \phi{\downarrow} = \{\xi \mid \xi \vdash \phi\}.$$

LEMMA 7.1.6. *If $\phi \nvdash \bot$ then $\phi{\uparrow}$ from Definition 7.1.5 is a filter and $\phi{\downarrow}$ is an ideal.*

Proof. It is routine to verify conditions 1 to 3 of Definition 7.1.1. We now consider condition 4. Suppose $\phi \vdash (b\ a)\cdot\xi$ for all but finitely many b. We take one particular $b\#\phi, \xi$ and conclude using ($\forall\mathbf{R}$) that $\phi \vdash \forall a.\xi$. The case of $\phi{\downarrow}$ is no harder. □

7.1.2. On universal quantification in filters. Lemma 7.1.7 is needed for Proposition 7.1.8. It is a fact of syntax related to Lemma 3.1.10.[19]

LEMMA 7.1.7. *Suppose ϕ is a predicate and suppose b is fresh (so $b\#\phi$). Then $\phi[a{\mapsto}b]$ is α-equivalent to $(b\ a)\cdot\phi$.*

Sketch proof. The substitution $[a{\mapsto}b]$ in $\phi[a{\mapsto}b]$ is the standard capture-avoiding substitution on the syntax of the predicate ϕ, from Definition 5.1.1. The permutation $(b\ a)\cdot\phi$ is the permutation acting on the syntax of ϕ. But when we write 'b fresh', what exactly do we mean; fresh for what? For ϕ? For the α-equivalence class of ϕ? Or are we using the $\mathsf{И}$-quantifier? In fact, all of these are equivalent. The

[19] It would be easy to hide this with a bit of hand-waving. However, in the context of nominal techniques, we give it its own result and indicate the issues involved.

interested reader can find the key technical devices for a slick proof in [Gab11, Section 5], notably in Lemma 5.16. □

PROPOSITION 7.1.8. *Suppose p is a filter and a is an atom. Then:*

(*1*) *The following conditions are equivalent (below, n ranges over all atoms, including a):*

$$\forall a.\phi \in p \quad \Leftrightarrow \quad \forall u.\phi[a{\mapsto}u] \in p \quad \Leftrightarrow \quad \forall n{\in}\mathbb{A}.\phi[a{\mapsto}n] \in p \quad \Leftrightarrow \quad \nu b.(b\ a){\cdot}\phi \in p$$

(*2*) *If furthermore p is finitely supported and $a\#p$ then the following conditions are equivalent:*

$$\forall a.\phi \in p \quad \Leftrightarrow \quad \phi \in p$$

Proof. First we prove part 1. Suppose $\forall a.\phi \in p$. By ($\forall$**L**) from Figure 2 and condition 2 of Definition 7.1.1 also $\phi[a{\mapsto}u] \in p$ for every term u. It follows in particular that $\phi[a{\mapsto}n] \in p$ for every atom (i.e. variable symbol) n. It follows that $\phi[a{\mapsto}b] \in p$ for all $b\#\phi$ so by Lemma 7.1.7 also $\nu b.(b\ a){\cdot}\phi \in p$. Finally if $\nu b.(b\ a){\cdot}\phi \in p$ then by condition 4 of Definition 7.1.1 also $\forall a.\phi \in p$.

Part 2 follows from part 1 using Theorem 2.3.8. □

It will be useful to have Notation 7.1.9:

NOTATION 7.1.9. Call a filter p an **ultrafilter** when for every ϕ, precisely one of $\phi \in p$ and $\neg\phi \in p$ holds.

REMARK 7.1.10. Proposition 7.1.8 has some interesting ramifications which relate back to Remark 7.1.2.

We will shortly see how to construct ultrafilters (which are also maximally consistent sets of predicates) in Subsections 7.1.5 and 7.1.6, and organise them into an σ-algebra of *points* in Subsection 7.2 (see in particular Proposition 7.2.3). We will use this to prove completeness in Subsection 7.3.

Proposition 7.1.8 lets us make some interesting observations about what these ultrafilters look like. It is not hard to prove that if an ultrafilter p has finite support then by Proposition 7.1.8 for any $a\#p$ and ϕ,

$$\forall a.\phi \in p \quad \Leftrightarrow \quad \phi \in p \quad \Leftrightarrow \quad \exists a.\phi \in p$$

where $\exists a.\phi$ is shorthand for $\neg(\forall a.\neg\phi)$, and so for any a (even if it is not fresh for p) and ϕ,

$$((\forall a.\phi) \Leftrightarrow \exists a.\phi) \in p.$$

So a finitely supported ultrafilter 'believes' that $\forall = \exists$, and so 'believes' that quantification is trivial. Quantification motivates this paper, so a finitely supported ultrafilter is for us degenerate and uninteresting.

In that sense Proposition 7.1.8 implies that ultrafilters must have infinite support—at least, the ones we care about must have this property.

Ultrafilters are dual to predicates in the sense of duality theory; the dual notion to 'must have finite support' (like predicates and open sets) is evidently 'must *not* have finite support' (like ultrafilters).[20]

Informally this seems reasonable—but to some extent, Proposition 7.1.8 makes it formal.

7.1.3. Growing filters. It will be useful to make larger filters out of smaller filters:

DEFINITION 7.1.11. Suppose p is a set of predicates and ψ is a predicate. Then define:

$$p{+}\psi = \{\xi \mid \phi{\wedge}\psi \vdash \xi,\ \phi \in p\}$$

LEMMA 7.1.12. *Suppose p is a finitely supported filter and suppose ψ is a predicate. Then:*

— $p \subseteq p{+}\psi$.
— $\psi \in p{+}\psi$.

[20] Note that as discussed above, the story is more subtle than that. Ultrafilters can have finite support; it is just that the finitely supported ultrafilters represent worlds in which quantification is not interesting. More work could be done to make this discussion fully formal.

— *$p{+}\psi$ is closed under conditions 2 to 4 of Definition 7.1.1.*

As a corollary, if Z is an ideal and $p{+}y \cap Z = \varnothing$ then $p{+}y$ is a filter.

Proof. The corollary follows from the body of this result because since from condition 2 of Definition 7.1.3 $\bot \in Z$, so $\top \notin p$.

We now consider the body of this result. It is clear from the construction that $p \subseteq p{+}\psi$ and $\psi \in p{+}\psi$. We now check that $p{+}\psi$ satisfies conditions 2 to 4 of Definition 7.1.1:

(2) *If $\xi \in p{+}\psi$ and $\xi \vdash \xi'$ then $\psi' \in p{+}\psi$.* By construction.
(3) *If $\xi \in p{+}\psi$ and $\xi' \in p{+}\psi$ then $\xi{\wedge}\xi' \in p{+}\psi$.* Suppose $\phi{\wedge}\psi \vdash \xi$ and $\phi'{\wedge}\psi \vdash \xi'$ for $\phi, \phi' \in p$. Then by condition 3 of Definition 7.1.1 $\phi{\wedge}\phi' \in p$, and it is a fact that $(\phi{\wedge}\phi'){\wedge}\psi \vdash \xi{\wedge}\xi'$.
(4) *If $\nabla b.((b\ a){\cdot}\xi \in p{+}\psi)$ then $\forall a.\xi \in p{+}\psi$.* Suppose for cofinitely many b there exists a $\phi_b \in p$ such that $\phi_b{\wedge}\psi \vdash (b\ a){\cdot}\xi$. Then there certainly exists some b such that $b\#\psi, \xi, p$ and $\phi_b{\wedge}\psi \vdash (b\ a){\cdot}\xi$. Note by Proposition 7.1.8(2) that also $\forall b.\phi_b \in p$.
We apply $\forall b$ to both sides and we conclude that
$$(\forall b.\phi_b){\wedge}\psi \vdash \forall b.(b\ a){\cdot}\xi = \forall a.\xi.$$
By Lemma 4.1.9 and condition 2 of Definition 7.1.1 we conclude that $\forall a.\xi \in p{+}\psi$ as required.

□

7.1.4. Growing ideals. It will be useful to make larger ideals out of smaller ideals:

Definition 7.1.13. Suppose Z and Y are sets of predicates. Then define $Z{+}Y$ by

$$Z{+}Y = \{\xi' \mid \exists \xi{\in}Z, n{\in}\mathbb{N}, \psi_1, \ldots, \psi_n{\in}Y.\xi' \leq \xi{\vee}\psi_1{\vee}\ldots{\vee}\psi_n\}.$$

Lemma 7.1.14 is a version of Lemma 7.1.12 for ideals. It is the simpler result, because Definition 7.1.3 has nothing corresponding to condition 2 of Definition 7.1.1:

Lemma 7.1.14. *Suppose Z is an ideal and Y is some set of predicates. Then:*

— *$Z \subseteq Z{+}Y$ and $Y \subseteq Z{+}Y$.*
— *$Z{+}Y$ is closed under conditions 2 and 3 of Definition 7.1.3 (so that if $\top \notin Z{+}Y$ then it is an ideal).*

Proof. By routine calculations. □

7.1.5. Prime filters and ultrafilters

Definition 7.1.15. — Call a filter p **prime** when $\phi_1{\vee}\phi_2 \in p$ implies either $\phi_1 \in p$ or $\phi_2 \in p$.
— Suppose p is a filter and Z is an ideal. Call p **maximal with respect to** Z when $p{\cap}Z = \varnothing$ and for every filter p' with $p'{\cap}Z = \varnothing$, if $p \subseteq p'$ then $p = p'$.
— Call p **maximal** when it is maximal with respect to the ideal $\{\bot\}$.

We will use the terms *prime filter* and **point** synonymously henceforth (see also Definition 7.2.2).

Lemma 7.1.16 is standard, but we still check carefully that being 'nominal' does not interfere with the classical propositional structure; it all works. We need the result for Theorem 7.3.6:

Lemma 7.1.16. *A filter p is prime if and only if it is an ultrafilter (Notation 7.1.9).*

Proof. Suppose p is prime. We note that $(\xi{\vee}\neg\xi) \Leftrightarrow \top$ so by condition 2 of Definition 7.1.1 and non-emptiness of p, $\xi{\vee}\neg\xi \in p$ (we use the conditions of Definition 7.1.1 silently henceforth). By primeness of p, at least one of $\xi \in p$ and $\neg\xi \in p$ holds. Similarly $(\xi{\wedge}\neg\xi) \Leftrightarrow \bot$ so that $(\xi{\wedge}\neg\xi) \notin p$. It follows that precisely one of $\xi \in p$ and $\neg\xi \in p$ holds, so that p is an ultrafilter.

Conversely suppose p is an ultrafilter and suppose $\xi{\vee}\xi' \in p$ and yet $\xi \notin p$ and $\xi' \notin p$. Then $\neg\xi{\wedge}\neg\xi' \in p$, so that $\bot \in p$, a contradiction. □

Proposition 7.1.17. *Suppose p is a filter and Z is an ideal. If p is a maximal filter with respect to Z then it is prime.*

Proof. Suppose $\psi_1 \vee \psi_2 \in p$ and $\psi_1, \psi_2 \notin p$. By Lemma 7.1.12 and maximality we have that $p{+}\psi_1 \cap Z \neq \varnothing$ and $p{+}\psi_2 \cap Z \neq \varnothing$. It follows that there exist $\phi_1, \phi_2 \in p$ with $\phi_1 \wedge \psi_1 \in Z$ and $\phi_2 \wedge \psi_2 \in Z$. By condition 3 of Definition 7.1.3

$$(\phi_1 \wedge \psi_1) \vee (\phi_2 \wedge \psi_2) \in Z.$$

Now we rearrange the left-hand side to deduce that

$$\zeta = (\phi_1 \vee \phi_2) \wedge (\phi_1 \vee \psi_2) \wedge (\psi_1 \vee \phi_2) \wedge (\psi_1 \vee \psi_2) \in Z.$$

We now note that $\phi_1 \vee \phi_2 \in p$ (since $\phi_1 \in p$, and indeed also $\phi_2 \in p$) and $\phi_1 \vee \psi_2 \in p$ (since $\phi_1 \in p$) and $\psi_1 \vee \phi_2 \in p$ (since $\phi_2 \in p$) and $\psi_1 \vee \psi_2 \in p$ by assumption. But then $\zeta \in p$, so that by condition 2 of Definition 7.1.1 $\zeta \in p \cap Z$, a contradiction. □

Lemma 7.1.18 is a technical result which will be useful for proving Theorem 7.1.20:

Lemma 7.1.18. *Suppose*

— *p is a finitely supported filter and Z is an ideal, and suppose*
— *$\bot \in p{+}\forall a.\psi$ and $p \cap Z = \varnothing$.*

Write $Y = \{(b\ a)\cdot\psi \mid b \in \mathbb{A} \setminus (supp(p) \cup supp(\psi) \cup \{a\})\}$. Then:

(1) $p \cap (Z{+}Y) = \varnothing$.
(2) As a corollary, $Z{+}Y$ is an ideal (and by part 1 is disjoint from p).

Proof. Suppose $\bot \in p{+}\forall a.\psi$ and $p \cap Z = \varnothing$ and $p \cap (Z{+}Y) \neq \varnothing$. We note the following:

— Since $p \cap (Z{+}Y) \neq \varnothing$, there exist $b_1, \ldots, b_n \# p, \psi$ and $\xi \in Z$ with $\xi \vee (b_1\ a)\cdot\psi \vee \ldots \vee (b_n\ a)\cdot\psi \in p$.
— Since $\bot \in p{+}\forall a.\psi$ there exists $\phi \in p$ with $\vdash \phi \wedge \forall a.\psi \Leftrightarrow \bot$.
By Proposition 7.1.8(2) (since $b_1, \ldots, b_n \# p$) $\forall b_1 \ldots \forall b_n.\phi \in p$, and we see that we may assume without loss of generality that $b_1, \ldots, b_n \# \phi$.
— Since $p \cap Z = \varnothing$ we have $\forall \phi' \in p, \xi' \in Z.\ (\phi' \wedge \xi' = \bot)$.

By condition 3 of Definition 7.1.1

$$\phi \wedge (\xi \vee (b_1\ a)\cdot\psi \vee \ldots \vee (b_n\ a)\cdot\psi) \in p$$

and therefore by distributivity (Definition 4.4.1)

$$(\phi \wedge \xi) \vee (\phi \wedge (b_1\ a)\cdot\psi) \vee \ldots \vee (\phi \wedge (b_n\ a)\cdot\psi) \overset{\phi \wedge \xi = \bot}{=\!=} (\phi \wedge (b_1\ a)\cdot\psi) \vee \ldots \vee (\phi \wedge (b_n\ a)\cdot\psi) \in p.$$

By assumption $b_1, \ldots, b_n \# p$ so by Proposition 7.1.8(2) we deduce that

$$\forall b_1 \ldots b_n.\big((\phi \wedge (b_1\ a)\cdot\psi) \vee \ldots \vee (\phi \wedge (b_n\ a)\cdot\psi)\big) \in p.$$

Recall that by assumption $b_1, \ldots, b_n \# \phi, \psi$; by properties of first-order logic (including α-equivalence and distributivity) we conclude that

$$\phi \wedge \forall a.\psi \in p \quad \text{and so that} \quad \bot \in p.$$

This contradicts condition 1 of Definition 7.1.1.

For the corollary, it follows from condition 2 of Definition 7.1.1 that $\top \in p$ so that $\top \notin Z{+}Y$. We use Lemma 7.1.14. □

7.1.6. The Zorn argument

Lemma 7.1.19. *If $Z_1 \subseteq Z_2 \subseteq Z_3 \subseteq \ldots$ is an ascending chain of ideals then $\bigcup_i Z_i$ is an ideal.*

Proof. By standard calculations on the three conditions of Definition 7.1.3. □

A version of Lemma 7.1.19 for filters does not hold, because condition 4 of Definition 7.1.1 is not closed under ascending chains of filters. We can still make a Zorn-like argument (for a carefully selected chain) to prove the existence of maximal filters:

THEOREM 7.1.20. *Suppose p is a finitely supported filter and Z is a finitely supported ideal and suppose $p \cap Z = \varnothing$. Then there exists a prime filter q with $p \subseteq q$ and $q \cap Z = \varnothing$.*

As corollaries:

— *If $\phi \nvdash \psi$ then there exists a prime filter q such that $\phi \in q$ and $\psi \notin q$.*
— *If $\phi \nvdash \bot$ then there exists a prime filter q such that $\phi \in q$.*

Proof. The corollaries follow by considering $\phi\uparrow$ and $\psi\downarrow$ (see Definition 7.1.5 and Lemma 7.1.6). We now consider the main part of this result.

Write $\mathit{Predicates}$ for the set of all predicates from Definition 5.1.1. Enumerate $\mathbb{A} \times \mathit{Predicates}$ as a list of pairs $(a_i, \phi_i)_{i\in\mathbb{N}}$; by assumption in Definition 2.1.1 atoms are countable, and by construction so are predicates, so we can do this.

We define a sequence of (by Theorem 2.3.2) finitely supported disjoint filter-ideal pairs $(p_i, Z_i)_{i\in\mathbb{N}}$ as follows:

(1) $(p_0, Z_0) = (p, Z)$. By assumption $p \cap Z = \varnothing$.
(2) Suppose $i{\geq}1$ and $p_{i\text{-}1}{+}\forall a_i.\phi_i \cap Z_{i\text{-}1} = \varnothing$. Then take
$$(p_i, Z_i) = (p_{i\text{-}1}{+}\forall a_i.\phi_i, Z_{i\text{-}1}).$$
It follows from Lemma 7.1.12 that p_i is a filter and from Theorem 2.3.2 that it is finitely supported.
(3) Suppose $i{\geq}1$ and $p_{i\text{-}1}{+}\forall a_i.\phi_i \cap Z_{i\text{-}1} \neq \varnothing$. Then take
$$Y = \{(b\ a_i){\cdot}\phi_i \mid b{\in}\mathbb{A} \setminus (\mathit{supp}(p_{i\text{-}1}){\cup}\mathit{supp}(\phi_i){\cup}\{a_i\})\} \quad \text{and}$$
$$(p_i, Z_i) = (p_{i\text{-}1}, Z_{i\text{-}1}{+}Y).$$
It follows from Lemma 7.1.18 that Z_i is an ideal and $p_i \cap Z_i = \varnothing$.

We can note the following:

— *We note that $p \subseteq \bigcup_i p_i$ and $Z \subseteq \bigcup_i Z_i$.* Since we took $(p_0, Z_0) = (p, Z)$ and assumed $p \cap Z = \varnothing$.
— *We note that $\bigcup_i p_i$ is disjoint from $\bigcup_i Z_i$ and thus from Z.* By construction.
— *We note that $\bigcup_i Z_i$ is an ideal.* This is Lemma 7.1.19.
— *We note that $\bigcup_i p_i$ is a filter.* Conditions 1 to 3 of Definition 7.1.1 are routine. To check condition 4, suppose $\text{И}b.(b\ a){\cdot}\phi \in \bigcup_i p_i$; we need to prove $\forall a.\phi \in \bigcup_i p_i$. So let j be that index in the enumeration above such that $(a, \phi) = (a_j, \phi_j)$. At stage j, when we built (p_j, Z_j), there were two possibilities:
  — If $p_{j\text{-}1}{+}\forall a.\phi \cap Z = \varnothing$ then we must have put $\forall a.\phi$ into p_j, so that $\forall a.\phi \in \bigcup_i p_i$ and we are done.
  — If $p_j{+}\forall a.\phi \cap Z \neq \varnothing$ then we must have put $(b\ a){\cdot}\phi$ into Z_j for cofinitely many b, so that $\text{И}b.(b\ a){\cdot}\phi \in \bigcup_i Z_i$.
  But this is impossible because we assumed that cofinitely many $(b\ a){\cdot}\phi$ were in $\bigcup_i p_i$, which is disjoint from $\bigcup_i Z_i$.
— *We note that $\bigcup_i p_i$ is a maximal filter disjoint from the ideal $\bigcup_i Z_i$.* Consider any ϕ and choose some fresh a (so $a\#\phi$). Note that $\vdash \forall a.\phi \Leftrightarrow \phi$.
  Let j be that index in the enumeration above such that $(a, \phi) = (a_j, \phi_j)$. It follows from the structure of the algorithm above that precisely one of $\phi \in p_j$ or $\phi \in Z_j$ will hold. Maximality follows.

Therefore by Proposition 7.1.17 $\bigcup_i p_i$ is a prime filter, and by construction $p \subseteq \bigcup_i p_i$ and $(\bigcup_i p_i) \cap Z = \varnothing$. □

REMARK 7.1.21. The proof of Theorem 7.1.20 is unusual in that we consider a chain of filter-ideal pairs instead of a chain of filters.

The reasons for this may not be obvious. Why bother? When we add Y to $Z_{i\text{-}1}$, why not add $Y' = \{\neg\psi \mid \psi \in Y\}$ to $p_{i\text{-}1}$ instead? After all, we have negation.

However if we do this then the set we might write as $p{+}Y'$ would fail to be a filter, because it would contain $(b\ a){\cdot}\neg\phi$ for cofinitely many b, but not $\forall a.\neg\phi$. Furthermore we cannot just add $\forall a.\neg\phi$ because we may not *want* this to be in p—what we want is subtly different: that $(b\ a){\cdot}\phi$ not be added for cofinitely many b 'by accident' at later stages (possibly infinitely many later stages).

Thus the asymmetry between Definitions 7.1.1 and 7.1.3, that Definition 7.1.1 has a condition for $\forall$ (condition 4) whereas Definition 7.1.3 does not, is a useful part of the design.

Perhaps we should write about *strong* filters and ideals which satisfy a $\forall/\exists$-condition, and *weak* ones which need not. In that terminology, we can note the interesting fact that our proof of the existence of maximal strong filters / ideals requires us to consider also weak ideals / filters.

7.2. The amgis-action on (prime) filters

Recall from Definition 3.2.1 the pointwise $\mathfrak{o}$-action $p[u{\leftarrow}a] = \{x \mid x[a{\mapsto}u] \in p\}$. In this subsection we check that this preserves the property of being a (prime) filter (Definitions 7.1.1 and 7.1.15) of predicates (so p is a set of predicates and u in the $\mathfrak{o}$-action is a term from Definition 5.1.1).

The work of this subsection happens in Lemma 7.2.1; Proposition 7.2.3 then puts the result in a some nice packaging.

LEMMA 7.2.1. *If p is a filter then so is $p[u{\leftarrow}a]$. Furthermore, if p is prime then so is $p[u{\leftarrow}a]$.*

Proof. We check the conditions in Definition 7.1.1. We use Proposition 3.2.2 without comment:

— $\bot \notin p[u{\leftarrow}a]$. Since it is a fact of syntax that $\bot[a{\mapsto}u] = \bot$. We use condition 1 of Definition 7.1.1.
— *If* $\phi{\in}p[u{\leftarrow}a]$ *and* $\phi \vdash \phi'$ *then* $\phi'{\in}p[u{\leftarrow}a]$. If $\phi \vdash \phi'$ then it is a fact of first-order logic derivability that also $\phi[a{\mapsto}u] \vdash \phi'[a{\mapsto}u]$. We use condition 2 of Definition 7.1.1.
— *If* $\phi{\in}p[u{\leftarrow}a]$ *and* $\phi'{\in}p[u{\leftarrow}a]$ *then* $\phi{\wedge}\phi' \in p[u{\leftarrow}a]$. It is a fact of syntax that $(\phi{\wedge}\phi')[a{\mapsto}u] = \phi[a{\mapsto}u]{\wedge}(\phi'[a{\mapsto}u])$. We use condition 3 of Definition 7.1.1.
— *If* $\mathit{N}b'.((b'\ b){\cdot}\phi \in p[u{\leftarrow}a])$ *then* $\forall b.\phi \in p[u{\leftarrow}a]$. Choose some fresh c (so $c\#x, u$). By Corollary 2.1.12 $(b'\ c){\cdot}(c\ b){\cdot}\phi = (b'\ b){\cdot}\phi$ and by $(\forall\alpha)$ also $\forall b.\phi = \forall c.(c\ b){\cdot}\phi$. Thus, we may assume without loss of generality that $b\#u$.
Now suppose $((b'\ b){\cdot}\phi)[a{\mapsto}u] \in p$ for all but finitely many b'; so suppose $b'\#u$. By Corollary 2.1.12 $(b'\ b){\cdot}u = u$, so that $(b'\ b){\cdot}(\phi[a{\mapsto}u]) \in p$ for all but finitely many b'. By condition 4 of Definition 7.1.1 $\forall b.(\phi[a{\mapsto}u]) \in p$ and since $b\#u$ it follows that $(\forall b.\phi)[a{\mapsto}u] \in p$.

Now suppose p is prime and suppose $(\psi_1{\vee}\psi_2)[a{\mapsto}u] \in p$. Then $\psi_1[a{\mapsto}u]{\vee}(\psi_2[a{\mapsto}u]) \in p$, so that either $\psi_1[a{\mapsto}u] \in p$ or $\psi_2[a{\mapsto}u] \in p$. □

Recall from Definition 7.1.15 that we called prime filters *points*:

DEFINITION 7.2.2. Write *Points* for the $\mathfrak{o}$-algebra determined by prime filters and the pointwise actions from Definition 3.2.1. That is:

— $|\mathit{Points}| = \{p \mid p \text{ is a prime filter}\}$.
— $\pi{\cdot}p = \{\pi{\cdot}\phi \mid \phi \in p\}$ and $p[u{\leftarrow}a] = \{\phi \mid \phi[a{\mapsto}u] \in p\}$.

PROPOSITION 7.2.3. *Points is indeed an $\mathfrak{o}$-algebra. Furthermore, Points is exact (Definition 3.3.2).*

Proof. The first part is just Lemma 7.2.1 combined with Proposition 3.2.6.

We now prove exactness. Suppose $\mathit{N}c.p[u{\leftarrow}c] = q[u{\leftarrow}c]$.

Consider some predicate ϕ. Note that cofinitely many (all but finitely many) atoms c satisfy all of the following properties, since cofinitely many atoms satisfy each of them, and a finite intersection of cofinite sets is cofinite:

— $p[u{\leftarrow}c] = q[u{\leftarrow}c]$ (by assumption),
— $c\#\phi$, so that also $\phi = \phi[c{\mapsto}u]$ (a fact of syntax),
— $\phi[c{\mapsto}u] \in p \Leftrightarrow \phi \in p[u{\leftarrow}c]$ (Proposition 3.2.2), and
— $\phi[c{\mapsto}u] \in q \Leftrightarrow \phi \in q[u{\leftarrow}c]$ (Proposition 3.2.2).

So there exists at least one c satisfying all the above. Then

$$\phi \in p \Leftrightarrow \phi[c{\mapsto}u] \in p \Leftrightarrow \phi \in p[u{\leftarrow}c] \Leftrightarrow \phi \in q[u{\leftarrow}c] \Leftrightarrow \phi[c{\mapsto}u] \in q \Leftrightarrow \phi \in q. \qquad \square$$

7.3. Proof of completeness

Recall from Definition 7.2.2 the set of *points* $Points$, which by Proposition 7.2.3 is an exact $\mathfrak{o}$-algebra. Recall from Definition 3.4.6 the definition of σ-powerset of an $\mathfrak{o}$-algebra, for example $Pow_\sigma(Points)$, and recall that by Theorem 6.4.1 it is a FOLeq algebra.

Recall from Example 3.1.3 and Definition 5.1.1 the syntax of terms r with real substitution $r[a{\mapsto}t]$. Then:

Definition 7.3.1. Define an interpretation $\mathcal{I}$ (Definition 5.2.2) as follows:

— Take $\mathcal{U}$ to be the syntax of terms with real substitution, and $\mathsf{f}^{\mathcal{I}}(a_1,\ldots,a_n) = \mathsf{f}(a_1,\ldots,a_n)$.
— Take $\mathcal{L}$ to be $Pow_\sigma(Points)$ considered as a FOLeq algebra.
— Take $\mathsf{P}^{\mathcal{I}}(a_1,\ldots,a_n) = \{p \in Points \mid \mathsf{P}(a_1,\ldots,a_n) \in p\}$.

As we shall see, Definition 7.3.1 is what we need to build a complete model of first-order logic with equality. To check this, it only remains to connect Definition 7.3.1 to the sets machinery we have built so far, and verify what comes out the other end.

Lemma 7.3.2. *If r is a term (Definition 5.1.1) then $[\![r]\!] = r$.*

Proof. An easy consequence of taking $\mathcal{U}$ to be syntax with substitution, and of Definition 5.2.5. $\square$

Remark 7.3.3. We do not have to specify the interpretation of equality $=$ because it is fixed by Definition 6.3.1, just as the intepretations of $\bot$, $\neg$, $\wedge$, and $\forall$ are fixed. This was observed in Theorem 6.4.1.

We briefly revisit Remark 6.4.2 and simplify it using Lemma 7.3.2, to sum up how the definitions instantiate to our case of $Pow_\sigma(Points)$:

$$\begin{array}{rl}
[\![\bot]\!] &= \varnothing \\
[\![r{=}s]\!] &= \{p{\in}Points \mid \nu c.(p[r{\leftarrow}c] = p[s{\leftarrow}c])\} \\
[\![\phi{\wedge}\psi]\!] &= [\![\phi]\!] \cap [\![\psi]\!] = \{p{\in}Points \mid p{\in}[\![\phi]\!] \wedge p{\in}[\![\psi]\!]\} \\
[\![\neg\phi]\!] &= Points \setminus [\![\phi]\!] = \{p{\in}Points \mid p{\notin}[\![\phi]\!]\} \\
[\![\forall a.\phi]\!] &= \bigcap\nolimits^{\#a} [\![\phi]\!] = \bigcap\nolimits_r [\![\phi]\!][a{\mapsto}r]
\end{array}$$

Lemma 7.3.4. *(1)* $r_1{=}^{\mathcal{I}}r_2 = \{p \in Points \mid (r_1{=}r_2) \in p\}$.
(2) $\mathsf{P}^{\mathcal{I}}(r_1,\ldots,r_n) = \{p \in Points \mid \mathsf{P}(r_1,\ldots,r_n) \in p\}$.

Proof. For part 1, we prove two implications:

— Suppose $(r_1{=}r_2) \in p$. We will show that $p \in (r_1{=}^{\mathcal{I}}r_2)$, that is, that $\nu c.p[r_1{\leftarrow}c] = p[r_2{\leftarrow}c]$.
We note by Proposition 3.2.2 that for any atom c,

$$\phi \in p[r_1{\leftarrow}c] \Leftrightarrow \phi[c{\mapsto}r_1] \in p \quad \text{and} \quad \phi \in p[r_2{\leftarrow}c] \Leftrightarrow \phi[c{\mapsto}r_2] \in p.$$

By assumption $(r_1{=}r_2) \in p$ so by properties of first-order logic and condition 2 of Definition 7.1.1

$$\phi[c{\mapsto}r_1] \in p \quad \text{if and only if} \quad \phi[c{\mapsto}r_2] \in p.$$

It follows by Proposition 3.2.2 that $\phi \in p[r_1{\leftarrow}c] \vdash \phi \Leftrightarrow \phi \in p[r_2{\leftarrow}c]$.
Since ϕ and c were arbitrary, we conclude in particular that $\nu c.p[r_1{\leftarrow}c] = p[r_2{\leftarrow}c]$.
— Suppose $p \in (r_1{=}^{\mathcal{I}}r_2)$, so $\nu c.p[r_1{\leftarrow}c] = p[r_2{\leftarrow}c]$. We will show that $(r_1{=}r_2) \in p$.
Consider some c such that $c \notin fa(r_1) \cup fa(r_2)$ and such that $p[r_1{\leftarrow}c] = p[r_2{\leftarrow}c]$. By condition 2 of Definition 7.1.1, $(r_1{=}r_1) \in p$. Also since $c \notin fa(r_1)$ we have that $(r_1{=}r_1) = (r_1{=}c)[c{\mapsto}r_1]$, so that $(r_1{=}c)[c{\mapsto}r_1] \in p$, and by Proposition 3.2.2 $(r_1{=}c) \in p[r_1{\leftarrow}c]$. Therefore $(r_1{=}c) \in p[r_2{\leftarrow}c]$ and again by Proposition 3.2.2 and our assumption that $c \notin fa(r_2)$ we conclude that $(r_1{=}r_2) \in p$ as required.

For part 2 we reason as follows:

$$\begin{array}{lll}
\mathsf{P}(r_1,\dots,r_n) \in p \Leftrightarrow \mathit{И}a_1,\dots,a_n.\mathsf{P}(a_1,\dots,a_n)[a_1{\mapsto}r_1,\dots,a_n{\mapsto}r_n] \in p & & \text{Fact of syntax}\\
\Leftrightarrow \mathit{И}a_1,\dots,a_n.\mathsf{P}(a_1,\dots,a_n) \in p[r_1{\leftarrow}a_1]\dots[r_n{\leftarrow}a_n] & & \text{Prop 3.2.2}\\
\Leftrightarrow \mathit{И}a_1,\dots,a_n.p[r_1{\leftarrow}a_1]\dots[r_n{\leftarrow}a_n] \in \mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n) & & \text{Def 7.3.1}\\
\Leftrightarrow \mathit{И}a_1,\dots,a_n.p \in \mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)[a_1{\mapsto}r_1,\dots,a_n{\mapsto}r_n] & & \text{Prop 3.4.2}\\
\Leftrightarrow p \in \mathsf{P}^{\mathsf{J}}(r_1,\dots,r_n) & & \text{Def 5.2.5}
\end{array}$$

□

We should briefly check that Definition 7.3.1 is well-defined, in the sense that $\mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)$ from clause 3 is indeed an element of $\mathcal{L}$ from clause 2. This is routine:

LEMMA 7.3.5. *Continuing the notation of Definition 7.3.1, $\mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)$ which is equal to $\{p \in Points \mid \mathsf{P}(a_1,\dots,a_n) \in p\}$, is an element of $Pow_\sigma(Points)$.*

Proof. Finite support is from Theorem 2.3.2 since the support of $\mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)$ is bounded by $\{a_1,\dots,a_n\}$, which is finite. It remains to check conditions 1 and 2 of Definition 3.4.6, namely:

— For fresh a (so $a \notin \{a_1,\dots,a_n\}$) and any term r, $\mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)[a{\mapsto}r] = \mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)$.
— For fresh b (so $b \notin \{a_1,\dots,a_n\}$) and any atom m (so possibly $m \in \{a_1,\dots,a_n\}$), $\mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)[m{\mapsto}r] = (b\ m){\cdot}\mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)$.

For the first condition, by Proposition 3.4.2 $p \in \mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)[a{\mapsto}r]$ if and only if $\mathit{И}a'.p[r{\leftarrow}a'] \in \mathsf{P}^{\mathsf{J}}(a_1,\dots,a_n)$. By Definition 7.3.1 this is if and only if $\mathit{И}a'.\mathsf{P}(a_1,\dots,a_n) \in p[r{\leftarrow}a']$. By Proposition 3.2.2 this is if and only if $\mathit{И}a'.\mathsf{P}(a_1,\dots,a_n)[a'{\mapsto}r] \in p$. It is now a fact of syntax that $\mathsf{P}(a_1,\dots,a_n)[a{\mapsto}r] = \mathsf{P}(a_1,\dots,a_n)$.

The second condition follows much as the first, using the fact of syntax that $\mathsf{P}(a_1,\dots,a_n)[m{\mapsto}b] = (b\ m){\cdot}\mathsf{P}(a_1,\dots,a_n)$. □

THEOREM 7.3.6. $[\![\phi]\!] = \{p \in Points \mid \phi \in p\}$.

Proof. By induction on ϕ:

— *The case of* $\bot$. By assumption in Definition 7.1.1 $p \in Points$ is consistent (meaning $\bot \notin p$). Furthermore, $[\![\bot]\!] = \varnothing$.
— *The cases of* $r_1{=}r_2$ *and* $\mathsf{P}(r_1,\dots,r_n)$. This is Lemma 7.3.4.
— *The case of* $\phi\wedge\psi$. From the inductive hypothesis and conditions 2 and 3 of Definition 7.1.1.
— *The case of* $\neg\phi$. By assumption p is prime (Definition 7.1.15), and it follows by Lemma 7.1.16 that $\neg\phi \in p$ if and only if $\phi \in p$. Also by assumption $[\![\neg\phi]\!] = Points \backslash [\![\phi]\!]$.
— *The case of* $\forall a.\phi$. Combining Remark 7.3.3 with Proposition 3.4.2 $p \in [\![\forall a.\phi]\!]$ means that $\mathit{И}c.p[r{\leftarrow}c] \in (c\ a){\cdot}[\![\phi]\!]$ for every r. By inductive hypothesis this is if and only if $\mathit{И}c.(c\ a){\cdot}\phi \in p[r{\leftarrow}c]$ for every r, which by Proposition 3.2.2 and α-conversion is if and only if $\phi[a{\mapsto}r] \in p$ for every r. We use Proposition 7.1.8(1). □

By Proposition 6.3.8 $(u{=}^{\mathsf{J}}u) = Points$. In Proposition 7.3.7 we show that specifically for the interpretation of Definition 7.3.1, a converse holds:

PROPOSITION 7.3.7. *If $(r_1{=}^{\mathsf{J}}r_2) = Points$ then $r_1 = r_2$ (that is, r_1 and r_2 are syntactically identical terms).*

Proof. Suppose $(r_1{=}^{\mathsf{J}}r_2) = Points$ and $r_1 \neq r_2$. It is a fact of first-order logic that $\neg(r_1{=}r_2) \nvdash \bot$ (easily proved using cut-elimination and the syntax-directed nature of the derivation rules). By Theorem 7.1.20 there exists a point p such that $\neg(r_1{=}r_2) \in p$. Now by assumption $p \in (r_1{=}^{\mathsf{J}}r_2)$, so by Theorem 7.3.6 $(r_1{=}^{\mathsf{J}}r_2) \in p$. Thus $\bot \in p$, a contradiction. □

REMARK 7.3.8. It is worth pausing to note, informally but mathematically, why Propositions 6.3.8 and 7.3.7 are remarkable.

We constructed $Pow_\sigma(Points)$ by taking the σ-powerset of points. The 'nominal' aspects of our construction in this paper allows us to interpret $\top$, $\wedge$, and *also* $\forall$ as three aspects of a single unifying notion of *fresh-finite limits*—but let us set that aside for now.

In addition to the above, Propositions 6.3.8 and 7.3.7 make formal an idea that even if the underlying notion of predicate did not contain an equality—in the sense of going back to Definition 5.1.1 and erasing equality $=$ from predicates, and going to the first-order logic sequent rules in Figure 2 and erasing $(=\mathbf{R})$ and $(=\mathbf{L})$—then even so, $Pow_\sigma(Points)$ would *still* have an equality.

In summary: $Pow_\sigma(Points)$ 'generates' a sets-based notion of equality, just as it 'generates' sets-based notions of conjunction and universal quantification as fresh-finite limits, and negation as sets complement.

COROLLARY 7.3.9. *If* $\phi \nvdash \psi$ *then* $[\![\phi]\!] \not\subseteq [\![\psi]\!]$.

Proof. Suppose $\phi \nvdash \psi$. By Theorem 7.1.20 there exists a point q with $\psi \notin q$ and $\phi{\uparrow}{\subseteq}q$ so that $\phi \in q$. By Theorem 7.3.6 it follows that $q \in [\![\phi]\!]$ and $q \notin [\![\psi]\!]$, and the result follows. □

Theorem 7.3.10 is a converse to Theorem 5.2.13:

THEOREM 7.3.10 (Completeness). *If* $[\![\Phi \vdash \Psi]\!]$ *is true for every interpretation in every FOLeq algebra, then* $\Phi \vdash \Psi$.

Proof. From Corollary 7.3.9. □

COROLLARY 7.3.11. $[\![\phi]\!] = Points$ *if and only if* $\vdash \phi$ *in FOLeq.*

8. TARSKI MODELS

We recall the usual definition of model for first-oder logic, which goes back to [Tar44] and is based on valuations. We will show how to lift this to a nominal model.

The main definition is Definition 8.2.1 and the main proofs are in Corollary 8.2.6 and Proposition 8.2.15.

The reader will probably be familiar with the valuation-based models and may ask: why bother taking something familiar and translating it to something *less* familiar? Usually, we expect to see it the other way around—the unfamiliar translated to the familiar.

However, we will argue in Remark 8.2.16 that the nominal semantics is more natural.

8.1. Tarski-style model of first-order logic

We briefly sketch the standard 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 8.1.1. To avoid the confusion between sets and nominal sets, we may write *ordinary set* for the former.

DEFINITION 8.1.2. Suppose X is an ordinary set. Write $\mathbb{A}{\Rightarrow}X$ for the set of functions from atoms to X. Let ς range over elements of $\mathbb{A}{\Rightarrow}X$ and call these **valuations** (to X).

If $x \in X$ then define $\varsigma[a{:=}x]$ to be the function on atoms given by:

$$\begin{array}{ll}(\varsigma[a{:=}x])(a) = x & \\ (\varsigma[a{:=}x])(b) = \varsigma(b) & \text{for all other } b\end{array}$$

DEFINITION 8.1.3. Define a permutation action on $\varsigma \in \mathbb{A}{\Rightarrow}X$ by

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

REMARK 8.1.4. $\mathbb{A}{\Rightarrow}X$ is a set with a permutation action (Definition 2.1.4). It is not in general a nominal set because it has elements without finite support.[21] For our purposes that will not be a problem.

The action from Definition 8.1.3 is a special case of the *conjugation action* $(\pi{\cdot}\varsigma)(a) = \pi{\cdot}\varsigma(\pi^{-1}(a))$ (already mentioned in Subsection 2.2.4), where we use the trivial action $\pi{\cdot}x = x$ for every $x \in X$. This is standard; for a specifically 'nominal' discussion see [Gab13, Definition 2.4.2].

Recall from Definition 5.1.1 the notion of *signature* (Σ, Π, ar) of *function symbols* $\mathsf{f} \in \Sigma$ and *predicate symbols* $\mathsf{P} \in \Pi$ with their associated *arities* $ar(\mathsf{f})$, $ar(\mathsf{P})$. For the rest of this subsection fix some signature.

DEFINITION 8.1.5. 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 $\mathsf{f}^{\mathcal{N}} : |\mathcal{N}|^{ar(\mathsf{f})}{\Rightarrow}|\mathcal{N}|$ and to each predicate-former P a function $\mathsf{P}^{\mathcal{N}} : |\mathcal{N}|^{ar(\mathsf{P})}{\Rightarrow}\{\bot, \top\}$.

DEFINITION 8.1.6. $\{\bot, \top\}$ is a complete Boolean algebra. We use standard definitions, such as $\wedge$, $\neg$, $\bigwedge$, and $\bigvee$ without comment

Recall the syntax of first-order logic from Definition 5.1.1.

DEFINITION 8.1.7. Define a (standard) **interpretation** function $[\![\text{-}]\!]^{\mathcal{N}}_{\varsigma}$ mapping terms and predicates to elements of $|\mathcal{N}|$ and $\{\bot, \top\}$ (considered as a Boolean algebra) respectively, as follows:

$$[\![a]\!]^{\mathcal{N}}_{\varsigma} = \varsigma(a) \qquad [\![\mathsf{f}(r_1, \ldots, r_n)]\!]^{\mathcal{N}}_{\varsigma} = \mathsf{f}^{\mathcal{N}}([\![r_1]\!]^{\mathcal{N}}_{\varsigma}, \ldots, [\![r_n]\!]^{\mathcal{N}}_{\varsigma})$$
$$[\![\bot]\!]^{\mathcal{N}}_{\varsigma} = \bot \qquad [\![\mathsf{P}(r_1, \ldots, r_n)]\!]^{\mathcal{N}}_{\varsigma} = \mathsf{P}^{\mathcal{N}}([\![r_1]\!]^{\mathcal{N}}_{\varsigma}, \ldots, [\![r_n]\!]^{\mathcal{N}}_{\varsigma})$$
$$[\![\phi' \mathsf{A} \phi]\!]^{\mathcal{N}}_{\varsigma} = [\![\phi']\!]^{\mathcal{N}}_{\varsigma} \wedge [\![\phi]\!]^{\mathcal{N}}_{\varsigma} \qquad [\![\mathsf{V}a.\phi]\!]^{\mathcal{N}}_{\varsigma} = \textstyle\bigwedge_{x \in |\mathcal{N}|} [\![\phi]\!]^{\mathcal{N}}_{\varsigma[a:=x]}$$
$$[\![\neg\phi]\!]^{\mathcal{N}}_{\varsigma} = \neg[\![\phi]\!]^{\mathcal{N}}_{\varsigma}$$

Theorem 8.1.8 expresses the usual soundness and completeness result for first-order logic; for details and proofs see e.g. [vD94, Subsection 1.5]:

THEOREM 8.1.8. *$\Phi \vdash \Psi$ 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 $\bigvee_{\psi\in\Psi}[\![\psi]\!]^{\mathcal{N}}_{\varsigma} = \top$.*

8.2. Lifting to a FOLeq algebra

We now show how to 'lift' a model over ordinary sets to a nominal model (Proposition 8.2.15). We then deduce completeness for nominal Boolean algebras (Corollary 8.2.18).

8.2.1. Lifting to a sigma-algebra

DEFINITION 8.2.1. Given ordinary sets X and Y define a termlike σ-algebra $Tarski(X, X)$, and define a σ-algebra $Tarski(X, Y)$ over $Tarski(X, X)$ by:

$$Tarski(X, X) = (|\,Tarski(X, X)|, \cdot, \mathsf{sub}_{Tarski(X,Y)}, \mathsf{atm}_{Tarski(X,X)})$$
$$Tarski(X, Y) = (|\,Tarski(X, Y)|, \cdot, \mathsf{sub}_{Tarski(X,Y)})$$

as follows:

— $|\,Tarski(X, Y)|$ is the set of functions f from $\mathbb{A}{\Rightarrow}X$ to Y such that there exists a finite set $A_f \subseteq \mathbb{A}$ such that

$$\forall a{\in}A_f.\varsigma(a) = \varsigma'(a) \quad \text{implies} \quad f(\varsigma) = f(\varsigma') \tag{1}$$

(if $X = Y$ then we obtain $|\,Tarski(X, X)|$).
— The permutation action $\cdot$ is defined by

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

[21] 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.

— If $f \in \mathbb{A}{\Rightarrow}Y$ and $u \in \mathbb{A}{\Rightarrow}X$ define

$$(f[a{\mapsto}u])(\varsigma) = f(\varsigma[a{:=}u(\varsigma)]).$$

— If $X = Y$ then define $\mathsf{atm}_{Tarski(X,X)}(a)(\varsigma) = \varsigma(a)$.

NOTATION 8.2.2. $\mathsf{atm}_{Tarski(X,X)}(a)$ is a bit of a mouthful so we may just write it as a. It will always be clear whether we mean '$a \in \mathbb{A}$' or '$a \in Tarski(X,X)$'.

LEMMA 8.2.3. $(|Tarski(X,Y)|, \cdot)$ *from Definition 8.2.1 is indeed a nominal set (Definition 2.1.5).*

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

Suppose $\pi(a) = a$ for every $a \in A_f$ where A_f is the finite set of atoms whose existence for each f is assumed in Definition 8.2.1. By Definition 8.2.1 $(\pi{\cdot}f)(\varsigma) = f(\pi^{-1}{\cdot}\varsigma)$. By Definition 8.1.3 $(\pi^{-1}{\cdot}\varsigma)(a) = \varsigma(\pi(a))$. Now by assumption $\varsigma(a) = \varsigma(\pi(a))$ for every $a \in A_f$. Therefore, $(\pi^{-1}{\cdot}\varsigma)(a) = \varsigma(a)$ for every $a \in A_f$, and so $f(\varsigma) = f(\pi^{-1}{\cdot}\varsigma)$, and so $(\pi{\cdot}f)(\varsigma) = f(\varsigma)$. Thus, f has finite support (and is supported by A_f). □

LEMMA 8.2.4. *Suppose* $f \in |Tarski(X,Y)|$. *Then* $\forall a{\in}supp(f).\varsigma(a){=}\varsigma'(a)$ *implies* $f(\varsigma) = f(\varsigma')$.

Proof. It suffices to show that if $a \in A_f \backslash supp(f)$ and $x \in X$ then $f(\varsigma[a{:=}x]) = f(\varsigma)$. Choose fresh b (so $b \notin A_f$). By part 1 of Corollary 2.1.12 $(b\ a){\cdot}f = f$, since $a, b \notin supp(f)$. We reason as follows:

$$f(\varsigma[a{:=}x]) \overset{b\notin A_f}{=} f(\varsigma[a{:=}x][b{:=}\varsigma(a)]) \overset{(b\ a)\cdot f=f}{=} f(\varsigma[b{:=}x]) \overset{b\notin A_f}{=} f(\varsigma) \qquad \square$$

REMARK 8.2.5. Lemma 8.2.4 does not follow from Lemma 8.2.3: to see this, take $X{=}Y{=}\{0,1\}$ and $f(\varsigma) = min\{\varsigma(a) \mid a{\in}\mathbb{A}\}$. Then by Theorem 2.3.2 $supp(f){=}\varnothing$ so that $\forall a{\in}supp(f).\varsigma(a){=}\varsigma'(a)$ for any ς and ς', yet $f(\varsigma) \neq f(\varsigma')$ where $\varsigma = \lambda a{\in}\mathbb{A}.0$ and $\varsigma' = \lambda a{\in}\mathbb{A}.1$.

What makes Lemma 8.2.4 work is the interaction of support with condition (1) of Definition 8.2.1.

COROLLARY 8.2.6. $Tarski(X,X)$ *is indeed a termlike* σ*-algebra, and* $Tarski(X,Y)$ *is a* σ*-algebra over* $Tarski(X,X)$.

Proof. By Definitions 3.1.1 and 8.2.1 and see that we need to check equivariance and the $(\sigma*)$ axioms from Figure 1. This is routine:

— *Rule* $(\sigma\mathbf{a})$ *(only for* $Tarski(X,X)$*).* Unpacking definitions, $a[a{\mapsto}u](\varsigma) = \varsigma[a{:=}u(\varsigma)](a) = u(\varsigma)$.
— *Rule* $(\sigma\mathbf{id})$. Again we unpack definitions: $f[a{\mapsto}a](\varsigma) = f(\varsigma[a{:=}\varsigma(a)]) = f(\varsigma)$.
— *Rule* $(\sigma\#)$. Using Lemma 8.2.4.
— *Rule* $(\sigma\alpha)$. Suppose $f \in Tarski(X,Y)$ and $b\#f$, and $u \in Tarski(X,X)$. Then $f[a{\mapsto}u](\varsigma) = f(\varsigma[a{:=}u(\varsigma)])$. Also, unpacking Definitions 8.1.3 and 8.2.1 $((b\ a){\cdot}f)[b{\mapsto}u] = f(\varsigma[a{:=}u(\varsigma)][b{:=}\varsigma(a)])$. We use Lemma 8.2.4.
— *Rule* $(\sigma\sigma)$. Suppose $f \in Tarski(X,Y)$ and $u, v \in Tarski(X,X)$ and $a\#v$. We just unpack definitions:

$$\begin{aligned} f[a{\mapsto}u][b{\mapsto}v](\varsigma) &= f[a{\mapsto}u](\varsigma[b{:=}v(\varsigma)]) \\ &= f(\varsigma[a{:=}u(\varsigma[b{:=}v(\varsigma)])][b{:=}v(\varsigma)]) \end{aligned}$$

$$\begin{aligned} f[b{\mapsto}v][a{\mapsto}u[b{\mapsto}v]](\varsigma) &= f[b{\mapsto}v](\varsigma[a{:=}u(\varsigma[b{:=}v(\varsigma)])]) \\ &= f(\varsigma[a{:=}u(\varsigma[b{:=}v(\varsigma)])][b{:=}v(\varsigma[a{:=}u(\varsigma[b{:=}v(\varsigma)])])]) \\ &= f(\varsigma[a{:=}u(\varsigma[b{:=}v(\varsigma)])][b{:=}v(\varsigma)]) \end{aligned}$$

The final step is valid using Lemma 8.2.4 since $a\#v$. (There are two symbols here: $\mapsto$ and $:=$. $f[a{\mapsto}u]$ is 'f with u substituted for a' from Definition 8.2.1, and $\varsigma[a{:=}u(\varsigma)]$ is 'ς with a maps to $u(\varsigma)$' from Definition 8.1.2.) □

8.2.2. Interlude: more on limits in nominal posets. For this subsection, fix a nominal poset $\mathcal{L}$ (Definition 4.1.1).

It is useful to continue and extend the maths from Subsection 4.1. Recall from Notation 2.1.3 the definition of fix:

DEFINITION 8.2.7. Following [Gab09b; GC11] define $x\langle\!\langle a \rangle\!\rangle$ by

$$\begin{aligned} x\langle\!\langle a \rangle\!\rangle &= \{\pi{\cdot}x \mid \pi \in fix(supp(x)\backslash\{a\})\}. \\ &= \{x\} \cup \{(b\ a){\cdot}x \mid b\#x\} \end{aligned}$$

Write $\bigwedge x\langle\!\langle a \rangle\!\rangle$ for the $\leq$-greatest lower bound of $x\langle\!\langle a \rangle\!\rangle$, if this exists.

Lemma 8.2.8 is related to Lemma 3.1.11 and Proposition 8.2.9 is related to Propositions 4.2.6 and 4.2.8:

LEMMA 8.2.8. $a\#x\langle\!\langle a \rangle\!\rangle$.

Proof. We note by the pointwise action that $\pi'{\cdot}x\langle\!\langle a \rangle\!\rangle = \{(\pi' \circ \pi){\cdot}x \mid \pi \in fix(supp(x)\backslash\{a\})\}$. We choose a fresh b (so $b\#x$, that is, $b{\notin}supp(x)$) and use Corollary 2.1.12(3) and routine calculations. □

PROPOSITION 8.2.9. *Suppose $a{\in}\mathbb{A}$ and $x{\in}|\mathcal{L}|$. Then:*

(1) *If $\bigwedge^{\#a}x$ exists then so does $\bigwedge x\langle\!\langle a \rangle\!\rangle$, and they are equal.*
(2) *Suppose $\mathcal{L}$ has a monotone σ-action (Definition 4.2.1). Then if $\bigwedge x\langle\!\langle a \rangle\!\rangle$ exists, then so does $\bigwedge^{\#a}x$, and they are equal.*

Proof. (1) By Definition 4.1.1 $\bigwedge^{\#a}x \leq x$ and $a\#\bigwedge^{\#a}x$. It follows by equivariance of $\leq$ and Corollary 2.1.12 that $\bigwedge^{\#a}x \leq \pi{\cdot}x$ for every $\pi \in fix(supp(x)\backslash\{a\})$. Therefore $\bigwedge^{\#a}x$ is a lower bound for $x\langle\!\langle a \rangle\!\rangle$.

Now consider z some other lower bound for $x\langle\!\langle a \rangle\!\rangle$, so that $\forall\pi{\in}fix(supp(x)\backslash\{a\}).z \leq \pi{\cdot}x$. Choose fresh b (so $b\#x, z$); by Theorem 2.3.2 $\bigwedge^{\#b}(b\ a){\cdot}x$ exists (and by Lemma 4.1.9 it is equal to $\bigwedge^{\#a}x$). It follows by equivariance of $\leq$ and Corollary 2.1.12 (since $a, b\#z$) that $z \leq (b\ a){\cdot}x$ so $z \leq \bigwedge^{\#b}(b\ a){\cdot}x \overset{\text{L4.1.9}}{=} \bigwedge^{\#a}x$.

(2) By Lemma 8.2.8 and Theorem 2.3.2 $a\#\bigwedge x\langle\!\langle a \rangle\!\rangle$ so $\bigwedge x\langle\!\langle a \rangle\!\rangle$ is an $a\#$lower bound for x.
Consider any other z such that $z \leq x$ and $a\#z$. By $(\sigma\#)$ $z[a{\mapsto}n] = z$ for every $n{\in}\mathbb{A}$. By monotonicity $z = z[a{\mapsto}n] \leq x[a{\mapsto}n]$ for every $n{\in}\mathbb{A}$. It follows using Lemma 3.1.10 that $z \leq \bigwedge x\langle\!\langle a \rangle\!\rangle$. □

REMARK 8.2.10. We have seen multiple characterisations of quantification:

— Fresh-finite limits from Notation 4.1.3.
— Limits of permutation orbits, above in Proposition 8.2.9.
— Limits of substitution instances (if there is a monotone σ-action), from Propositions 4.2.6 and 4.2.8.

Note of Proposition 8.2.9 that we only need atom-for-atom substitution (what this author calls a *renaming action*) in the proof.

DEFINITION 8.2.11. Call a nominal poset $\mathcal{L} = (|\mathcal{L}|, \cdot, \leq)$ **nominally complete** when every finitely supported subset $\mathcal{X} \subseteq |\mathcal{L}|$ has a greatest lower bound.

In another terminology: $\mathcal{L}$ is nominally complete when it has limits of finitely supported diagrams.

PROPOSITION 8.2.12. *If $\mathcal{L}$ is nominally complete and has a monotone σ-action, then it is finitely fresh-complete (Definition 4.1.1).*

In other words: having finitely supported limits and substitution implies having fresh-finite limits.[22]

[22] The reverse implication does not hold; having finitely supported limits and substitution is a far stronger condition.

Proof. We use Proposition 4.1.5. By Theorem 2.3.2 $\varnothing \subseteq |\mathcal{L}|$ and $\{x,y\} \subseteq |\mathcal{L}|$ and $x\langle\!\langle a \rangle\!\rangle$ are finitely supported. By Proposition 8.2.9 the a#limit $\bigwedge^{\#a} x$ exists in $\mathcal{L}$ if and only if the limit of $x\langle\!\langle a \rangle\!\rangle$ does. The result follows. □

8.2.3. Lifting the logical structure

Definition 8.2.13. Suppose $f', f \in |\mathit{Tarski}(X, \{\bot, \top\})|$. Write $f' \leq f$ when $\forall \varsigma. f'(\varsigma) \leq f(\varsigma)$.

Lemma 8.2.14. *With the partial order $f' \leq f$ of Definition 8.2.13, the σ-action from Definition 8.2.1 is monotone.*

Proof. Suppose $f' \leq f$ and $u \in |\mathit{Tarski}(X,X)|$ and $\varsigma \in \mathbb{A} \Rightarrow X$. We reason as follows:

$$\begin{array}{lll} (f'[a{\mapsto}u])(\varsigma) = f'(\varsigma[a{:=}u(\varsigma)]) & & \text{Definition 8.2.1} \\ \leq f(\varsigma[a{:=}u(\varsigma)] & & \text{Definition 8.2.13 } f' \leq f \\ = (f[a{\mapsto}u])(\varsigma) & & \text{Definition 8.2.1} \end{array}$$ □

Proposition 8.2.15. *$(|\mathit{Tarski}(X, \{\bot, \top\})|, \cdot, \leq)$ is nominally complete, complemented, and has a compatible σ-action and an equality.*

As a corollary, $(|\mathit{Tarski}(X, \{\bot, \top\})|, \cdot, \leq)$ is also a FOLeq algebra (Definition 4.4.3).

Proof. The greatest lower bound of a finitely supported set $\mathcal{X} \subseteq |\mathit{Tarski}(X, \{\bot, \top\})|$, and the complement of $f \in |\mathit{Tarski}(X, \{\bot, \top\})|$, are defined pointwise by

$$(\bigwedge \mathcal{X})(\varsigma) = \bigwedge_{f \in \mathcal{X}} f(\varsigma) \qquad \text{and} \qquad (\neg f)(\varsigma) = \neg(f(\varsigma)).$$

Equality is defined by

$$(u_1 =^{\mathit{Tarski}(X,\{\bot,\top\})} u_2)(\varsigma) = \begin{cases} \top & u_1(\varsigma) = u_2(\varsigma) \\ \bot & u_1(\varsigma) \neq u_2(\varsigma). \end{cases}$$

Checking that the σ-action is compatible and that the definitions above are correct, is routine.

By Lemma 4.2.3(2) the σ-action is monotone. The corollary follows by Proposition 8.2.12. □

Remark 8.2.16. By Proposition 8.2.15 every Tarski-style model is a FOLeq algebra. However, not every FOLeq algebra is a Tarski-style model, because by Proposition 8.2.15 Tarski-style models have limits for all, possibly infinite, finitely supported subsets.

In FOLeq algebras the notion of completeness used is *fresh-finite* completeness (Subsection 4.1); a concept which is natural to express in a nominal universe. This makes FOLeq algebras complete enough to interpret $\wedge$ and $\forall$, and no more.

Thus, Tarski-style models have more limits than first-order logic requires. This is not detectable from inside first-order logic, since first-order logic is sound and complete for the Tarski-style models and this is why Tarski-style models suffice to model first-order logic in a ZF universe.

However, completeness is not the only issue: if it were, we would only ever need the Herbrand model of syntax quotiented by derivable equivalence (Section 9). The issue is to capture in abstract semantic terms exactly that structure necessary to interpret first-order logic. Neither the Tarski models nor indeed the Herbrand models quite do this; but FOLeq algebras do. In this sense, Tarski-style models are less natural than the nominal semantics proposed in this paper.

Before we are done, we need to build an interpretation of term-formers and predicate-formers in the sense of Definition 5.2.2, using the interpretation in 'ordinary' sets from Definition 8.1.5. This is not hard:

Definition 8.2.17. Suppose $\mathcal{N}$ is an ordinary model (Definition 8.1.5), so that by Proposition 8.2.15 we have that $\mathit{Tarski}(|\mathcal{N}|, \{\bot, \top\})$ is a FOLeq algebra.

Define an interpretation $\mathcal{I}$ over $Tarski(|\mathcal{N}|,\{\bot,\top\})$ in the sense of Definition 5.2.2 as follows:

$$\mathsf{f}^{\mathcal{I}}(a_1,\ldots,a_{ar(\mathsf{f})})(\varsigma) = \mathsf{f}^{\mathcal{N}}(\varsigma(a_1),\ldots,\varsigma(a_{ar(\mathsf{f})}))$$
$$\mathsf{P}^{\mathcal{I}}(a_1,\ldots,a_{ar(\mathsf{P})})(\varsigma) = \mathsf{P}^{\mathcal{N}}(\varsigma(a_1),\ldots,\varsigma(a_{ar(\mathsf{P})}))$$

where the a_i are distinct atoms (the a_i above left are short for $\mathtt{atm}_{Tarski(X,X)}(a_i)$ from Definition 8.2.1).

We obtain an interpretation of first-order logic immediately from Definition 5.2.7 and soundness immediately from Theorem 5.2.13. If we like, we can also leverage the completeness theorem with respect to ordinary models:

COROLLARY 8.2.18. *If $\Phi \nvdash \Psi$, then there exists a termlike σ-algebra $\mathcal{U}$ and a FOLeq algebra $\mathcal{M}$ over $\mathcal{U}$ such that $\bigwedge_{\phi\in\Phi}[\![\phi]\!]^{\mathcal{I}} \not\leq \bigvee_{\psi\in\Psi}[\![\psi]\!]^{\mathcal{I}}$.*

Proof. By completeness of first-order logic there exists some ordinary model $\mathcal{N}$ (Definition 8.1.5) and valuation ς to $\mathcal{N}$ such that $\bigwedge_{\phi\in\Phi}[\![\phi]\!]^{\mathcal{N}}_{\varsigma} = \top$ and $\bigvee_{\psi\in\Psi}[\![\psi]\!]^{\mathcal{N}}_{\varsigma} = \bot$.

So consider $Tarski(|\mathcal{N}|,\{\bot,\top\})$ and consider the interpretation $\mathcal{I}$ from Definition 8.2.17. It follows from Definition 8.2.13 that $\bigwedge_{\phi\in\Phi}[\![\phi]\!]^{\mathcal{I}} \not\leq \bigvee_{\psi\in\Psi}[\![\psi]\!]^{\mathcal{I}}$. □

9. HERBRAND MODELS (LINDENBAUM-TARSKI ALGEBRAS)

We conclude, briefly, by observing that predicate syntax quotiented by derivable equality is a FOLeq algebra. This construction is variously called a *Herbrand*, *Lindenbaum*, or *Lindenbaum-Tarski* construction, algebra, or model—or just 'syntax quotiented by derivable equivalence'.

There is not intended to be much new in this observation; just to show how syntax quotiented by derivable equivalence fits with the idea of a FOLeq algebra.

If there is any subtlety it is that we build our model from possibly open syntax, and the familiar syntactic notion of variable will be handled in our nominal abstract machinery as a special case of the background Fraenkel-Mostowski notion of atom. It all works perfectly, which is part of the point.

Recall the syntax of terms and predicates from Definition 5.1.1, and the notion of logical entailment from Figure 2.

DEFINITION 9.0.19. Define a **logical equivalence** relation $\sim$ on predicates by

$$\phi \sim \phi' \quad \text{when} \quad \phi \vdash \phi' \ \wedge \ \phi' \vdash \phi.$$

Write $[\phi]_\sim$ for the $\sim$-equivalence class of ϕ.

If x and x' are $\sim$-equivalence classes then impose a partial order by $x \leq x'$ when $\phi \in x$ and $\phi' \in x'$ and $\phi \vdash \phi'$. It is a fact that this is well-defined (does not depend on the choice of ϕ and ϕ').

Give predicates the natural permutation action where π acts on ϕ by acting on the atoms in ϕ. Theorem 2.3.2 tells us the following:

LEMMA 9.0.20. *If $\phi \sim \phi'$ then $\pi{\cdot}\phi \sim \pi{\cdot}\phi'$.*

DEFINITION 9.0.21. Give $\sim$-equivalence classes the **pointwise** permutation action by $\pi{\cdot}[\phi]_\sim = [\pi{\cdot}\phi]_\sim$.

DEFINITION 9.0.22. Give $\sim$-equivalence classes a **pointwise** σ-action by $[\phi]_\sim[a{\mapsto}u] = [\phi[a{\mapsto}u]]_\sim$.

We need to check that $[\phi]_\sim[a{\mapsto}u]$ is well-defined, that is, if $\phi \sim \phi'$ then $\phi[a{\mapsto}u] \sim \phi'[a{\mapsto}u]$. This is a fact of first-order logic.

DEFINITION 9.0.23. Define $\mathcal{U}$ to be the termlike σ-algebra of terms where substitution is real substitution on syntax, and take $\mathcal{L} = (|\mathcal{L}|,\cdot,\leq,\mathcal{U},\mathsf{sub}_{\mathcal{U}})$ to be $\sim$-equivalence classes of predicates, with the pointwise permutation and σ-actions.

THEOREM 9.0.24. *$\mathcal{L}$ is a FOLeq algebra if we take $\top$ to be $[\mathsf{T}]_\sim$, $[\phi]_\sim \wedge [\phi']_\sim$ to be $[\phi\mathsf{\wedge}\phi']_\sim$, $\bigwedge^{\#a}[\phi]_\sim$ to be $[\forall a.\phi]_\sim$, and $r{=}s$ to be $[r\mathsf{=}s]_\sim$.*

Proof. Well-definedness and limit properties are all just properties of first-order logic. Finite support is from Theorem 2.3.2, since syntax is finite. □

10. CONCLUSIONS

10.1. Semantics out of context

Traditionally we admit numbers as a primitive datatype, and we emulate names using numbers (since numbers are countably infinite) or functional arguments. This is neatly packaged up by Tarski-style semantics, which broadly speaking has the following shape:

$$(\text{Natural numbers} \Rightarrow \text{Semantics of terms}) \Rightarrow \text{Semantics of predicates}$$

Tarski-style semantics have exerted a powerful, almost subliminal, influence on how semantics have been designed, and how the relationship between syntax and semantics has been understood. Proposing a good alternative to this view is one of our main goals.

In Tarski-style semantics the valuation (mapping numbers to semantics of terms as illustrated above) is a *context* of variable-to-denotation assignments. The semantics of a predicate ϕ exists *in the context* of a valuation ς, and we write $[\![\phi]\!]_\varsigma$. ϕ provides the syntax and ς provides the context, and what Tarski really taught us was that semantics exists *in context*.

Yet, from our point of view context is only compensating for a mathematical foundation that is too poor to directly represent variables. We would consider the slogan *"there is no such thing as a free variable"* from [Per82] to be succinct, eloquent—and incorrect.

In this paper we do things differently. We interpret names more-or-less as themselves, using urelemente in Fraenkel-Mostowski foundations. Denotationally speaking there *is* such a thing as a free variable, and it is an *urelement*—an *atom*.

In this paper we apply this idea to give semantics to first-order logic. Given a set, its powerset is naturally a Boolean algebra: in a nutshell, this paper is notes that given a set of terms, the σ-powerset of its $\mathfrak{o}$-powerset is naturally a model of first-order logic with equality (see Example 3.4.11)—and furthermore, this concrete powersets model is susceptible to simple and attractive nominal algebraic and lattice characterisations (and a topological treatment is possible too, though not in this paper). More on this in Subsection 10.2.

The Tarski notion of variable context is transmuted into the nominal notion of support (Definition 2.1.8).[23] In the terminology of the title of this paper, the semantics we obtain is *absolute*; we talk about *the* semantics of ϕ, and write $[\![\phi]\!]$. Once the model is fixed, so is $[\![\phi]\!]$ the meaning of ϕ.

We can recover the minimal relevant context of $[\![\phi]\!]$ by calculating its support $supp([\![\phi]\!])$. However, if we do not care about support we can ignore it. This simplifies reasoning; it is easier to talk about 'an element' than 'an element in context'.

But support is more general than 'the free variables of'. It can be read as expressing what a nominal element x depends on—without needing to know what logic, program, or computation x represents (if any). If we use our foundations correctly then the support of x will relate sensibly to what x is being used to represent, and furthermore, the general nominal constructs will give us a useful structure within which to work. For instance in this paper:

— Lemma 5.2.8 notes that $supp([\![\phi]\!]) \subseteq fa(\phi)$—the nominal element $[\![\phi]\!]$ is supported by (at most) the atoms free in ϕ. This connects support of syntax with the corresponding notion of support of semantics.
— Similarly $(\sigma\#)$ from Figure 1 expresses that if $a \notin supp(x)$ then substitution $[a{\mapsto}u]$ does not change x. This connects nominal support—being affected by a permutation like $(b\ a)$—with the notion of semantic dependency that comes from being in a σ-algebra and being affected by $[a{\mapsto}u]$.

The two points above are visible in syntax: if a is not free in ϕ then ϕ with a substituted for a term r is equal to ϕ. By the two points above, if a is not free in ϕ then also $[\![\phi]\!][a{\mapsto}u] = [\![\phi]\!]$. In fact, the semantics is 'compositional' with respect to substitution; $[\![\phi[a{\mapsto}r]]\!] = [\![\phi]\!][a{\mapsto}[\![r]\!]]$ (Lemma 5.2.11). The reader

[23] In [GM11] are nominal Henkin-style semantics for higher-order logic in a typing context, but not a valuation context. We calculated denotation in a context of static (typing) information, but we were not forced to *also* use an explicit context of valuations—and we did not.

used to Tarski semantics might recognise the corresponding result $[\![\phi[a{\mapsto}r]]\!]_\varsigma = [\![\phi]\!]_{\varsigma[a\mapsto[r]_\varsigma]}$. From the point of view of this paper, that result is a special case of a more general result for a more general semantics.

Because an element x stands on its own, we do not need to label everything with a context, or think or prove lemmas about weakening or strengthening contexts, or about making the contexts of x and y equal so that we can combine them, or even about α-renaming contexts. We do not need valuations, or higher types (though we can have them if we like, as noted in Section 8). This is taken care of in the background.

Name-management is simpler in nominal techniques, and name-management is *the* thing that separates first-order logic from Boolean algebra, and it should be clear that things other than first-order logic would be susceptible to the mathematical style and nominal semantics we have introduced in this paper; some are suggested in the *Future work* below.

10.2. Recalling the trio of semantics

In this paper we have considered a trio of semantics: lattices, sets, and algebra.[24] Let us recall what they look like:

— A lattice-flavoured semantics specifies the meaning of a logic in terms of a partially-ordered set (a set with a transitive reflexive antisymmetric relation). The order relation models logical entailment; that is, the semantics is designed such that $\phi \vdash \psi$ should imply $[\![\phi]\!] \leq [\![\psi]\!]$.
We model logical connectives as limits; $\top$ is modelled as a top element (a greatest lower bound for nothing), $\wedge$ is modelled as a greatest lower bound for two elements. This is standard, and in categorical language we say that we assume *finite limits* of the lattice.
The new idea here is to model $\forall a$ using fresh-finite limits $\bigwedge^{\#a}$ (Definition 4.1.1; this was proposed in [DG12a]).
So by this semantics, $[\![\phi]\!]$ is an element of a poset with certain *fresh*-finite limits. Instead of finite limits, we assume *fresh-finite limits*.
— The nominal algebraic semantics is an algebra not over sets, but over *nominal* sets.
Nominal algebra enriches the syntax, judgement form, and semantics of universal algebra, while retaining (somewhat surprisingly) a purely equational flavour [GM09; Gab09a; Gab13; Gab12].
Nominal algebra, the nominal algebra axiomatisation of substitution (called σ-algebras here), and that of first-order logic, were proposed in a sequence of papers [GM06a; GM06b; GM08a; GM08c; Gab14; DG12b]. $\mathfrak{o}$-algebras are more recent and were created specifically to prove completeness/duality results.
By this semantics, $[\![\phi]\!]$ is an element of a nominal set which happens to be equipped with operations satisfying certain nominal equalities (see Appendix A.3).
— The sets semantics builds on the semantics of [GLP11; Gab14].[25]
Conjunction is interpreted as sets intersection and negation as sets complement. This is again standard.
We see universal quantification interpreted as an infinite intersection in Definition 6.2.1, and Proposition 4.2.6 notes that this is equal to a fresh-finite limit. Proposition 4.2.6 is one of a family of characterisation results[26] and an algebraic one is in Appendix A.3.
So by this semantics, $[\![\phi]\!]$ is a set which thanks to its σ and $\mathfrak{o}$ structure turns out to be able to interpret universal quantification, and as we saw in Subsection 6.3 also equality. We can write: σ + $\mathfrak{o}$ + powersets = First-Order Logic with equality.

[24] A pure nominal treatment of syntax-with-binding, rather than its semantics, is also possible. Indeed, that problem is what motivated the initial developments behind nominal techniques [Gab01; GP01; Gab11].
[25] *Note:* The logic of [Gab14] did not include equality and considers duality instead of completeness.
[26] E.g. $\{x'{\in}|\mathcal{L}| \mid supp(x'){\subseteq}supp(x){\setminus}\{a\} \wedge x'{\leq}x\}$ and $\bigwedge_{n\in\mathbb{A}} x[a{\mapsto}n]$.

The Stone duality theorem for Boolean algebras adds a fourth framework: topology. A topological duality result for FOL (but not FOLeq) algebras exists and uses technology similar to some of the material in this paper; see a sister paper [Gab14].

10.3. Related work

10.3.1. Nominal precedents. The idea of nominal axiomatisations of mathematics dates back to *nominal algebra* which was used to axiomatise in the first instance substitution and first-order logic [GM06a; GM06b; GM08a; GM08c; GM09].

The most developed such logic is at the time of writing *permissive-nominal logic* [DG10; DG12b] which extends nominal algebra with quantification (of unknowns: $\forall X$). We use it to finitely axiomatise first-order logic and arithmetic in a first-order setting.

Concerning the sets semantics we see in this paper, precursors are [GG08; Gab09b], which technically have little to do with this paper but which do build families of sets representations of what in this paper we call a σ-action, and [Gab14] which as mentioned above is a sister paper considering topology and with a slightly different technical treatment (different version of $\mathfrak{o}$-algebras, no equality, no lattices, and so on; at the time of writing, it has not yet been published).

More distantly related are the term equational systems of [FH08; FH10]. These have no concrete sets representations and the method of axiomatisation is quite different—so technically, this work is not very relevant to what we do in this paper—*but* it is based on presheaf semantics, and nominal sets admit a presheaf presentation (as noted in Subsection 10.3.7), so there is some overlap.

10.3.2. Tarski-style valuations and Herbrand-style quotiented syntax. We consider Tarski-style valuation-based semantics in Section 8, and predicates up to derivable equivalence in Section 9. We showed how both can be considered subclasses of FOLeq algebras.

As usual the Herbrand semantics is easy to construct but uninformative; building it is more a 'sanity check'.

Concerning the valuation semantics, we noted in Remark 8.2.16 that we get more limits than needed. We only really need fresh-finite limits to interpret first-order logic, but valuation semantics will not let us express that. In contrast, FOLeq algebras have only exactly those limits necessary to interpret first-order logic, which seems a significant point in their favour.

10.3.3. Combinators. Variables are not indispensable. This is the idea behind combinatory logic [Sch67; CF58], and it can be pushed a long way [DBB98; TG87] (the latter paper formalises set theory without using variables).

We see combinatory techniques as orthogonal to this paper. Yes, languages can be constructed without variables; but that does not make variables obsolete. Variables are useful because humans like to use them; combinators do not change that.

Sometimes people talk about combinators 'reducing' variables to a simpler problem. Not so. With names (nominal or otherwise) we can talk about a location independently of its site of binding. With combinators we cannot talk about a location except by exhibiting it as being at the other end of a combinator. These may be able to express the same computable functions, but that does not make them equivalent.

10.3.4. Cylindric and polyadic algebras. Cylindric and polyadic algebras treat quantification as a modality satisfying certain axioms. To model languages with infinitely many variables, we must admit infinitely many modalities.

The interested reader is referred to [Hal06, Chapter 8, page 243] (whose exposition is incredibly clear) and to [Mon00] and [HMT85].

The resemblance to parts of this paper are interesting; for instance, the infinite schemata of axioms $(\mathbf{P1})$ to $(\mathbf{P7})$ of a polyadic algebra in [Hal06, Chapter 8, page 244] are similar to projections of the axioms of Subsection A.3 into a first-order setting (where there is a symbol for every π and $\forall a$)—except, that polyadic and cylindric algebras are based on the monoid of finitely supported atoms-

substitutions (so: not necessarily bijective functions like permutations), and also, these systems exist in ZF so do not always insist on finite support, or in their terminology on *local finite dimensionality*.

(Tangentially we can also mention Fine's notion of *arbitrary objects* [Fin85], not because it has anything to do with algebra or first-order logic as such, but just because it too is based on a monoid of substitutions.)

Cylindric and polyadic algebras, and Fine's arbitrary objects, are distinctive because they do *not* belong to the Tarski family of variables-with-valuations and denotations-in-context. In that sense they are closer to the spirit of this paper.

However, we are not aware of anything like the mathematics of this paper being executed in the frameworks above. This is partly because we take permutations as primitive and so can capture finitely what it takes infinite axiom schemes to capture otherwise.[27] It is also because the nominal framework enforces symmetry (as discussed in the opening to Subsection 2.3) and finite support. These are all things that somehow fit together to make possible what we have done in this paper. Perhaps the research above *could not* do what we have done in this paper, because of the influence of the implicit Zermelo-Fraenkel foundation.

We should also mention *hyperdoctrines*. These are a categorical framework within which to consider semantics for logics and are very general, so we briefly sketch how hyperdoctrines work for the specific case of a one-sorted first-order classical logic.[28]

Consider a category T whose objects are natural numbers $0, 1, \ldots$ and for each term-former f of arity n an arrow f^{T} from n to 0. A hyperdoctrine is a functor F from T to the category of Boolean algebras, along with an assignment to each P of arity n of an element P^F in $F(n)$.

In spite of the categorical generality in which this idea is phrased, it really just re-states Tarski-style semantics. Yes, there are no valuations as such, but this is replaced by a finite context and the categorical framework enforces well-formedness (we only give a predicate semantics if its free variables are in the context). Useful as hyperdoctrines are for the reader wanting to work entirely within the vocabulary of categories, all they do is translate—and update—the ideas of Tarski into a new categorical language. The basic nature of the thing has not changed. Nominal techniques as applied in this paper are doing something different.

10.3.5. Boolean algebras with operators. A *Boolean algebra with operators* is a Boolean algebra $\mathcal{B}$ equipped with functions $|\mathcal{B}|^n \to |\mathcal{B}|$ which preserve intersections on each component. These have been studied extensively; first by Jonnson and Tarski [JT52], and then by Goldblatt [Gol89]. A concise and very readable survey is in the Introduction of [Hai00].

It is oversimplistic but reasonable to characterise FOLeq algebras as just being fancy Boolean algebras with operators. The operator concerned is not $\forall$ but the σ-algebra structure $[a{\mapsto}u]$—which since we are oversimplifying we might as well call *substitution*, though it is not necessarily syntactic.

Viewed as an operator, the substitution $[a{\mapsto}u]$ is a modality (a unary operator) with special properties: it commutes with meets $\wedge$, joins $\vee$, and negation $\neg$. Thus, $[a{\mapsto}u]$ can be viewed as both a modal Box and a modal Diamond operator, which is also a homomorphism of Boolean algebras, and the corresponding Kripke semantics is functional in the sense that each world has precisely one future world.

The flavour of our representation theorem is in keeping with this. Notably, the pointwise actions of Definitions 3.2.1 and 3.4.1

$$X[a{\mapsto}u] = \{p \mid \mathit{И}c.p[u{\leftarrow}c] \in (c\ a){\cdot}X\} \qquad \text{and} \qquad p[u{\leftarrow}a] = \{x \mid x[a{\mapsto}u] \in X\}$$

[27] Polyadic and cylindric algebras are ϵ away from us here, but by using *monoids* instead of *groups* they cannot guarantee that atoms that were distinct, remain distinct.

In this light, our *permutative convention* from Definition 2.1.1, which goes back to [GM06a; GM06b; GM08c], that in informal mathematical discourse in nominal techniques a, b, and c range over *distinct* atoms, seems very important.

[28] ...which is what we axiomatise in this paper. We hope it is quite clear by now how a multi-sorted logic, or an intuitionistic logic, would work in our nominal semantics.

are like the functional preimage under the relevant accessibility relation—though not quite, as we see in the use of $\mathit{И}$ on the left above, which hard-wires capture-avoidance and α-equivalence (Lemma 3.4.4).

Still, from the point of view of [JT52; Gol89] it is not unreasonable to view this paper as being 'just' about a particular kind of Boolean algebra with a particular family of operators satisfying certain axioms described in Figure 1. We mention this because it is not a bad way of viewing how the material of this paper fits in to the broader mathematical context.

Of course there is more going on than *just* that. Notably: we are working in FM sets so we have to make sure that our constructions preserve finite support; also we do not just have $[a{\mapsto}u]$ but also the universal quantifier $\forall$; but perhaps most interestingly, the axioms for σ are determined by properties of substitution—but the axioms of $\mathfrak{o}$ are *not* so determined. They emerge from the specific requirements of the specific proofs we need to carry out. So even if a FOLeq algebra is just a Boolean algebra with operators, the operators are organised and axiomatised into a specific and non-trivial structure, and how that structure gets 'inverted' under a modified capture-avoiding functional preimage operation, to form $\mathfrak{o}$-algebras, is subtle and certainly not obvious.

10.3.6. Varieties of FOLeq algebras. In Appendix A we give a purely nominal equational axiomatisation of FOLeq algebras, using *nominal algebra* [GM09]. The syntax of nominal algebra is based on equalities between nominal terms subject to syntactic freshness side-conditions which behave a bit like typing conditions. These do not impact on the algebraic flavour of the logic and one way of making that formal is with a version of the HSP theorem, or Birkhoff's theorem. We get a notion of *variety* suitable for nominal sets, and a theorem that a nominal algebra theory is precisely characterised by the variety of its models, for a suitable nominal notion of variety. For more details see [GM09; Gab09a].

10.3.7. The Schanuel topos: putting context back in, if we want it. The reader may know that nominal sets are equivalent to the Schanuel Topos, which can be presented as pullback-preserving presheaves (for a proof with calculations see [Gab11, Theorem 9.14]).

What this means in plain English is that every nominal $x{\in}|\mathcal{X}|$ can be thought of as a family of $A \vdash x$ for $A \subseteq \mathbb{A}$ such that $\mathit{supp}(x) \subseteq A$. In other words, x 'exists' in a context if and only if the context contains the support of x.

So in fact, everything we have done in this paper, admits a 'contextual' presentation. Furthermore, it may admit generalisations to presheaves not necessarily preserving pullbacks (to be more precise, not necessarily preserving pullbacks of *monos* [GH08]).

And yet, even if for the sake of argument we imagine such a generalisation is found and written, the proofs in this paper would be harder to discover, harder to present, and harder to disseminate, if we did not first have the simpler 'context-free' sets-based reasoning of nominal sets.

10.4. Future work

(1) *Apply the sets semantics to the λ-calculus.* A nominal algebra axiomatisation of the λ-calculus is in [GM08b; GM10], building on the nominal rewrite systems for the λ-calculus from [FGM04; FG07]. A sequent logic style presentation of the λ-calculus is in [GG10], along with a (non-nominal) Kripke style semantics.
In a sequel to the current paper, we build on this work to give for the first time a topological duality result for the λ-calculus [GG16].

(2) *Generalise the language beyond first-order logic.* This paper has only scratched the surface of what might be possible with our semantics. First-order logic has binding, but it does not have binding term-formers nor does it have the ability to reason directly on variable names. We do this all the time in informal mathematical practice—e.g. when we specify first-order logic itself.
It is a tricky question what kind of logic is being used here, though it seems to be a nominal one. Permissive-nominal logic is one such [DG10; DG12b]; it enriches first-order logic with binding term-formers and we have used it to axiomatise first-order logic and arithmetic [DG12b].
However, there may be much more to say here. We hope that the sets-based semantics of this paper, which is new, will be a good guide to us here.

(3) *Category theory in nominal sets.* The fresh-finite limit from Subsection 4.1 is natural if we build category theory internally to nominal sets; because a category is a set of objects and a set of arrows, it is natural in a nominal context to talk about 'the limit object satisfying a freshness side-condition'. Our treatment of posets in this paper begs generalisation to categories, and we look forward to finding out what story might be told here.[29] A hint of this is already in the paper; freshness conditions and adjoints (especially the adjoint explanation of quantifiers) are clearly related. This must be the topic of a later paper.
Another interesting generalisation is to take the notion of σ-algebra seriously as a categorical framework. The clue to this is to view a category as a kind of σ-algebra in which each arrow has one 'free variable' (its source). In a similar spirit, we might consider a (termlike) σ-algebra as a kind of semigroup in which we consider $x[a{\mapsto}y]$ as composition of x and y 'at a'. If we do this, it will not just be because we can: in computing, composition often happens at a location, in some sense.

(4) *Stone duality.* The representation theorem of this paper uses maximally consistent filters (which we call *points*), for a suitable notion of filter (Definitions 7.1.1 and 7.2.2).
Our logic is classical, so maximally consistent filters do indeed coincide with ultrafilters; see Lemma 7.1.16.
We can develop this further by characterising the right notion of topology on a (nominal) topological space and obtain a Stone duality result for FOLeq algebras.
This has been studied for the simpler case of first-order logic without equality. See [Gab14] (a sister paper to this one) and in particular see Definition 6.18 of [Gab14] where the topology on points is described. See also [For07], which uses different techniques but has similar goals.
Adapting [Gab14] to the definitions and results here would probably require a distinct paper.

(5) *The many-sorted case.* Our definition of termlike σ-algebra in Definition 3.1.1 is single-sorted; $\mathcal{U}$ is a single set with a σ-action over itself. For applications in computer science it would be useful to have a many-sorted version of FOLeq algebras.
This should be quite easy. It would suffice to generalise Definition 3.1.1: instead of $\mathcal{U}$ we have a type-indexed family $\mathcal{U}_\tau$ along with atoms for each type and a σ-action of type $\mathcal{U}_\tau \times \mathbb{A}_{\tau'} \times \mathcal{U}_{\tau'} \to \mathcal{U}_\tau$. The rest of the mathematics in this paper would be orthogonal and should remain unaffected.

10.5. Summary

We hope that the mathematics of this paper supports the following two arguments:

— The correct notion of model of first-order logic is a FOLeq algebra; this contains just what is necessary to model first-order logic and furthermore it supports good sets-based and poset-based characterisations. Underlying these are σ- and $\mathfrak{o}$-powersets, which are remarkable structures, and nominal algebra (itself a logic of independent interest). To express these we need nominal techniques—ZF sets and algebras are not expressive enough.
— Nominal foundations give other new, and not necessarily obvious, opportunities in logic and semantics. We study first-order logic in this paper and the untyped λ-calculus in [GG16]. We hope and suspect that the ideas behind these examples, and many of the tools we had to develop to carry out these examples, will have further uses (cf. the *Future work* above).

So this paper gives non-trivial answers to some specific technical questions, but stepping back to look at the larger picture, we see this work as one example of an exciting new way in which to apply nominal techniques in logic and semantics.

REFERENCES

[CF58] Haskell B. Curry and R. Feys, *Combinatory logic*, vol. I, North Holland, 1958.

[29] There is a general theory of categories in universes other than ZF sets [Kel82]. But we are not just enriching the notion of category; we are also enriching the notion of *limit*. So whatever mathematics emerges from that seems unlikely to be just a special case of the general machinery.

[DBB98] Wil Dekkers, Martin Bunder, and Henk Barendregt, *Completeness of two systems of illative combinatory logic for first-order propositional and predicate calculus*, Archive für Mathematische Logik **37** (1998), 327–341.

[DG10] Gilles Dowek and Murdoch J. Gabbay, *Permissive Nominal Logic*, Proceedings of the 12th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPDP 2010) (New York), ACM Press, 2010, pp. 165–176.

[DG12a] ———, *Nominal Semantics for Predicate Logic: Algebras, Substitution, Quantifiers, and Limits*, Proceedings of the 9th Italian Convention on Computational Logic (CILC 2012), CEUR workshop proceedings, vol. 857, 2012.

[DG12b] ———, *Permissive Nominal Logic (journal version)*, Transactions on Computational Logic **13** (2012), no. 3.

[DP02] B. A. Davey and Hilary A. Priestley, *Introduction to lattices and order*, 2 ed., Cambridge University Press, 2002.

[FG07] Maribel Fernández and Murdoch J. Gabbay, *Nominal rewriting (journal version)*, Information and Computation **205** (2007), no. 6, 917–965.

[FGM04] Maribel Fernández, Murdoch J. Gabbay, and Ian Mackie, *Nominal Rewriting Systems*, Proceedings of the 6th ACM SIGPLAN symposium on Principles and Practice of Declarative Programming (PPDP 2004), ACM Press, August 2004, pp. 108–119.

[FH08] Marcelo Fiore and Chung-Kil Hur, *Term equational systems and logics*, Electronic Notes in Theoretical Computer Science **218** (2008), 171–192.

[FH10] ———, *Second-order equational logic*, Proceedings of the 19th EACSL Annual Conference on Computer Science Logic (CSL 2010) (Berlin), Lecture Notes in Computer Science, Springer, 2010.

[Fin85] Kit Fine, *Reasoning with arbitrary objects*, Blackwell, 1985.

[For07] Henrik Forssell, *First-order logical duality*, Ph.D. thesis, Carnegie Mellon University, December 2007.

[Gab01] Murdoch J. Gabbay, *A Theory of Inductive Definitions with alpha-Equivalence*, Ph.D. thesis, University of Cambridge, UK, March 2001.

[Gab07] ———, *A General Mathematics of Names*, Information and Computation **205** (2007), no. 7, 982–1011.

[Gab09a] ———, *Nominal Algebra and the HSP Theorem*, Journal of Logic and Computation **19** (2009), no. 2, 341–367.

[Gab09b] ———, *A study of substitution, using nominal techniques and Fraenkel-Mostowski sets*, Theoretical Computer Science **410** (2009), no. 12-13, 1159–1189.

[Gab11] ———, *Foundations of nominal techniques: logic and semantics of variables in abstract syntax*, Bulletin of Symbolic Logic **17** (2011), no. 2, 161–229.

[Gab12] ———, *Unity in nominal equational reasoning: The algebra of equality on nominal sets*, Journal of Applied Logic **10** (2012), 199–217.

[Gab13] ———, *Nominal terms and nominal logics: from foundations to meta-mathematics*, Handbook of Philosophical Logic, vol. 17, Kluwer, 2013, (author's/publisher's numbering).

[Gab14] ———, *Stone duality for First-Order Logic: a nominal approach*, HOWARD-60. A Festschrift on the Occasion of Howard Barringer's 60th Birthday, Easychair books, 2014.

[GC11] Murdoch J. Gabbay and Vincenzo Ciancia, *Freshness and name-restriction in sets of traces with names*, Foundations of software science and computation structures, 14th International Conference (FOSSACS 2011), Lecture Notes in Computer Science, vol. 6604, Springer, 2011, pp. 365–380.

[GG08] Murdoch J. Gabbay and Michael Gabbay, *Substitution for Fraenkel-Mostowski foundations*, Proceedings of the 2008 AISB Symposium on Computing and Philosophy, 2008, pp. 65–72.

[GG10] Michael J. Gabbay and Murdoch J. Gabbay, *A simple class of Kripke-style models in which logic and computation have equal standing*, International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR 2010), 2010.

[GG16] Murdoch J. Gabbay and Michael J. Gabbay, *Representation and duality of the untyped lambda-calculus in nominal lattice and topological semantics, with a proof of topological completeness*, Annals of Pure and Applied Logic (2016), submitted 2012. See also arXiv preprint 1305.5968.

[GH08] Murdoch J. Gabbay and Martin Hofmann, *Nominal renaming sets*, Proceedings of the 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2008), Springer, November 2008, pp. 158–173.

[GLP11] Murdoch J. Gabbay, Tadeusz Litak, and Daniela Petrişan, *Stone duality for nominal Boolean algebras with NEW*, Proceedings of the 4th international conference on algebra and coalgebra in computer science (CALCO 2011), Lecture Notes in Computer Science, vol. 6859, Springer, 2011, pp. 192–207.

[GM06a] Murdoch J. Gabbay and Aad Mathijssen, *Capture-avoiding Substitution as a Nominal Algebra*, Proceedings of the 3rd International Colloquium on Theoretical Aspects of Computing (ICTAC 2006) (Berlin), Lecture Notes in Computer Science, vol. 4281, Springer, November 2006, pp. 198–212.

[GM06b] ———, *One-and-a-halfth-order logic*, Proceedings of the 8th ACM-SIGPLAN International Symposium on Principles and Practice of Declarative Programming (PPDP 2006), ACM, July 2006, pp. 189–200.

[GM07] ———, *A Formal Calculus for Informal Equality with Binding*, WoLLIC'07: 14th Workshop on Logic, Language, Information and Computation, Lecture Notes in Computer Science, vol. 4576, Springer, July 2007, pp. 162–176.

[GM08a] ———, *Capture-Avoiding Substitution as a Nominal Algebra*, Formal Aspects of Computing **20** (2008), no. 4-5, 451–479.

[GM08b] ———, *The lambda-calculus is nominal algebraic*, Reasoning in simple type theory: Festschrift in Honour of Peter B. Andrews on his 70th Birthday (Christoph Benzmüller, Chad Brown, Jörg Siekmann, and Rick Statman, eds.), Studies in Logic and the Foundations of Mathematics, IFCoLog, December 2008.

[GM08c] ———, *One-and-a-halfth-order Logic*, Journal of Logic and Computation **18** (2008), no. 4, 521–562.

[GM09] ———, *Nominal universal algebra: equational logic with names and binding*, Journal of Logic and Computation **19** (2009), no. 6, 1455–1508.

[GM10] ———, *A nominal axiomatisation of the lambda-calculus*, Journal of Logic and Computation **20** (2010), no. 2, 501–531.

[GM11] Murdoch J. Gabbay and Dominic P. Mulligan, *Nominal Henkin Semantics: simply-typed lambda-calculus models in nominal sets*, Proceedings of the 6th International Workshop on Logical Frameworks and Meta-Languages (LFMTP 2011), EPTCS, vol. 71, September 2011, pp. 58–75.

[Gol89] Robert Goldblatt, *Varieties of complex algebras*, Annals of Pure and Applied Logic **44** (1989), no. 3, 173–242.

[GP01] Murdoch J. Gabbay and Andrew M. Pitts, *A New Approach to Abstract Syntax with Variable Binding*, Formal Aspects of Computing **13** (2001), no. 3–5, 341–363.

[Hai00] Mariana Haim, *Duality for lattices with operators: A modal logic approach*, Ph.D. thesis, University of Amsterdam, 2000.

[Hal06] Paul R. Halmos, *Algebraic logic*, AMS Chelsea Publishing, 2006.

[HMT85] Leon Henkin, J. Donald Monk, and Alfred Tarski, *Cylindric algebras*, North Holland, 1971 and 1985, Parts I and II.

[ID96] C. Norris Ip and David L. Dill, *Better verification through symmetry*, Formal Methods in System Design **9** (1996), 41–75.

[Joh03] Peter T. Johnstone, *Sketches of an elephant: A topos theory compendium*, Oxford Logic Guides, vol. 43 and 44, OUP, 2003.

[JT52] Bjarni Jonnson and Alfred Tarski, *Boolean algebras with operators*, American Journal of Mathematics **74** (1952), no. 1, 127–162.

[Kel82] Max Kelly, *Basic concepts of enriched category theory*, London Mathematical Society Lecture Note Series, no. 64, Cambridge University Press, 1982.

[KW96] Edward Keenan and Dag Westerståhl, *Generalized quantifiers in linguistics and logic*, Handbook of Logic and Language (J. Van Benthem and A. Ter Meulen, eds.), Elsevier, 1996, pp. 837–894.

[MM92] Saunders Mac Lane and Ieke Moerdijk, *Sheaves in geometry and logic: A first introduction to topos theory*, Universitext, Springer, 1992.

[Mon00] Donald Monk, *An introduction to cylindric set algebras*, Logic journal of the IGPL **8** (2000), no. 4, 451–492.

[MVF+02] William McCune, Robert Veroff, Branden Fitelson, Kenneth Harris, Andrew Feist, and Larry Wos, *Short single axioms for boolean algebra*, Journal of Automated Reasoning **1** (2002), no. 29, 1–16.

[Per82] Alan J. Perlis, *Epigrams on programming*, September 1982, Available online `www.cs.yale.edu/quotes.html`, pp. 7–13.

[Pit13] Andrew M. Pitts, *Nominal sets: Names and symmetry in computer science*, Cambridge University Press, May 2013.

[Sch67] Moses Schönfinkel, *On the building blocks of mathematical logic*, From Frege to Gödel: a source book in mathematical logic, 1879-1931 (Jean van Heijenoort, ed.), Harvard University Press, 1967, Translated from the German by Stefan Bauer-Mengelberg.

[SP05] Mark R. Shinwell and Andrew M. Pitts, *Fresh objective Caml user manual*, Tech. Report UCAM-CL-TR-621, University of Cambridge, 2005.

[Tar44] Alfred Tarski, *The semantic conception of truth and the foundations of semantics*, Philosophy and Phenomenological Research **4** (1944).

[TG87] Alfred Tarski and Steven Givant, *A formalization of set theory without variables*, vol. 41, American Mathematical Society Colloquium Publications, 1987.

[vD94] Dirk van Dalen, *Logic and structure*, Universitext, Springer, 1994, Third, augmented edition.

[Wes89] Dag Westerståhl, *Quantifiers in formal and natural languages*, Handbook of Philosophical Logic, Synthèse, vol. 4, Reidel, 1989, pp. 1–131.

A. NOMINAL ALGEBRAIC AXIOMATISATION OF FRESH-FINITE LIMITS AND EQUALITY

Definition 4.1.1 set up the notion of a fresh-finite limit $\bigwedge^{\#A} X$. Proposition 4.1.5 split this notion up into three parts which are sometimes more convenient for proofs: $\top$, $\wedge$, and $\bigwedge^{\#a}$. These definitions are 'poset-flavoured'; that is, we talk about $\leq$ and greatest lower bounds for $\leq$.

We now set about axiomatising this structure in *nominal algebra* [GM09]. That is, we will characterise FOLeq algebras in terms of an underlying nominal set and functions on that set satisfying nominal equalities.

It is convenient to do this in three stages: bounded meet-semilattices with $\forall$, then bounded lattices with $\forall$, then FOLeq algebras.

A.1. Bounded meet-semilattices with $\forall$

Definition A.1.1. A **bounded meet-semilattice** in nominal sets is a tuple $\mathcal{M} = (|\mathcal{M}|, \cdot, \wedge, \top)$ where $(|\mathcal{M}|, \cdot)$ is a nominal set and $\top \in |\mathcal{M}|$ is an equivariant **top element** and $\wedge : (\mathcal{M} \times \mathcal{M}) \Rightarrow \mathcal{M}$ is an equivariant function, such that $\wedge$ and $\top$ form an idempotent monoid:

$$(x \wedge y) \wedge z = x \wedge (y \wedge z) \qquad x \wedge y = y \wedge x \qquad x \wedge x = x \qquad x \wedge \top = x$$

Here, x, y, z range over elements of $|\mathcal{M}|$.

DEFINITION A.1.2. As standard, a meet-semilattice acquires a partial order by setting $x \leq y$ when $x \wedge y = x$. A bounded meet-semilattice yields a poset with a top element ($\top$).

DEFINITION A.1.3. A **generalised universal quantifier** $\forall$ on a bounded meet-semilattice $\mathcal{M}$ is an equivariant map $\forall : (\mathbb{A} \times \mathcal{M}) \Rightarrow \mathcal{M}$ such that:

$$\begin{array}{lll} (\forall\alpha) & b\#x \Rightarrow & \forall b.(b\ a)\cdot x = x \\ (\forall\wedge) & & \forall a.(x \wedge y) = (\forall a.x) \wedge (\forall a.y) \\ (\forall\#) & a\#x \Rightarrow & \forall a.x = x \\ (\forall\leq) & & \forall a.x \leq x \end{array}$$

(Recall that $\forall a.x \leq x$ is shorthand for $(\forall a.x) \wedge x = \forall a.x$.)

LEMMA A.1.4. *If $x \leq y$ then $\forall a.x \leq \forall a.y$.*

Proof. $x \leq y$ means $x \wedge y = x$. We apply $\forall a$ to both sides and use $(\forall\wedge)$. □

PROPOSITION A.1.5. *Suppose $\mathcal{M}$ is a bounded meet-semilattice with $\forall$. Then with the partial order from Definition A.1.2 we have the following:*

— $\top$ is a top element.
— $x \wedge y$ is a limit for $\{x, y\}$.
— $\forall a.x$ is an $a\#$limit for $\{x\}$.

Proof. The first two parts are standard; the interesting case is part 3.

From $(\forall\alpha)$ and part 3 of Corollary 2.1.12 we derive that $a\#\forall a.x$. With this and $(\forall\leq)$ we have that $\forall a.x$ is an $a\#$lower bound for x.

Now suppose $z \leq x$ and $a\#z$. By Lemma A.1.4 and $(\forall\#)$ we deduce $z \leq \forall a.x$. □

COROLLARY A.1.6. *Every bounded meet-semilattice with $\forall$ can be viewed as a nominal poset with fresh-finite limits, and vice versa.*

Proof. Using the constructions of Propositions 4.1.5 and A.1.5. □

Definition A.1.7 and Proposition A.1.8 merely package the observations above in categorical language:

DEFINITION A.1.7. Given $\mathcal{M}$ and $\mathcal{M}'$ bounded meet-semilattices with $\forall$, a **homomorphism** $F : \mathcal{M} \Rightarrow \mathcal{M}'$ is a function on the underlying nominal sets such that:

— $F(\pi\cdot x) = \pi\cdot F(x)$, meaning that F is *equivariant* (Definition 2.1.6).
— $F(\top_{\mathcal{M}}) = \top_{\mathcal{M}'}$ and $F(x \wedge_{\mathcal{M}} y) = F(x) \wedge_{\mathcal{M}'} F(y)$ and $F(\forall_{\mathcal{M}} a.x) = \forall_{\mathcal{M}'} a.x$.

Given $\mathcal{N}$ and $\mathcal{N}'$ nominal posets with fresh-finite limits, a **homomorphism** $G : \mathcal{N} \Rightarrow \mathcal{N}'$ is a function on the underlying nominal sets such that:

— $G(\pi\cdot x) = \pi\cdot G(x)$ (so G is *equivariant*).
— $G(\bigwedge^{\#A} X) = \bigwedge^{\#A} \{G(x) \mid x \in X\}$.

Write $\mathsf{Semi}\forall$ for the category of bounded meet-semilattices with $\forall$ and homomorphisms between them, and $\mathsf{Poset}\bigwedge^{\#A}$ for the category of nominal posets with fresh-finite limits.

PROPOSITION A.1.8. *The natural functors mapping between $\mathsf{Semi}\forall$ and $\mathsf{Poset}\bigwedge^{\#A}$ define an isomorphism of categories.*

A.2. Bounded lattices with $\forall$

Our version of FOLeq is classical and has negation. This means that it has $\wedge$, but by dualising with negation it also has $\vee$. This has a minor effect on the natural axiomatisation of $\forall$. We therefore briefly sketch the case of $\forall$ for bounded lattices, of which FOLeq is a special case.

$$
\begin{array}{lll}
(\forall\alpha) & b\#x \Rightarrow & \forall b.(b\ a)\cdot x = x \\
(\forall\wedge) & & \forall a.(x \wedge y) = (\forall a.x) \wedge (\forall a.y) \\
(\forall\vee) & a\#y \Rightarrow & \forall a.(x \vee y) = (\forall a.x) \vee y \\
(\forall\leq) & & \forall a.x \leq x
\end{array}
$$

Fig. 4: Nominal algebra axioms for $\forall$ in a bounded lattice

Definition A.2.1. A **bounded lattice** in nominal sets is a tuple $\mathcal{L} = (|\mathcal{L}|, \cdot, \wedge, \vee, \top, \bot)$ where $(|\mathcal{L}|, \cdot)$ is a nominal set which we may just write $\mathcal{L}$, and $\top, \bot \in |\mathcal{L}|$ are equivariant **top** and **bottom** elements and $\wedge, \vee : (\mathcal{L} \times \mathcal{L}) \Rightarrow \mathcal{L}$ are equivariant functions, such that $\wedge$ and $\top$ form an idempotent monoid and so do $\vee$ and $\bot$,

$$
\begin{array}{llll}
(x \wedge y) \wedge z = x \wedge (y \wedge z) & x \wedge y = y \wedge x & x \wedge x = x & x \wedge \top = x \\
(x \vee y) \vee z = x \vee (y \vee z) & x \vee y = y \vee x & x \vee x = x & x \vee \bot = x
\end{array}
$$

and $\wedge$ and $\vee$ satisfy *absorption*

$$x \wedge (x \vee y) = x \qquad\qquad x \vee (x \wedge y) = x.$$

Here, x, y, z range over elements of $|\mathcal{L}|$.

Similarly to Definition A.1.2, a bounded lattice is a poset by taking $x \leq y$ to mean $x \wedge y = x$ or $x \vee y = y$ (the two conditions are provably equivalent).

Definition A.2.2. A **generalised universal quantifier** $\forall$ on a bounded lattice $\mathcal{L}$ is an equivariant map $\forall : (\mathbb{A} \times \mathcal{L}) \Rightarrow \mathcal{L}$ satisfying the equalities in Figure 4.

Remark A.2.3. Comparing Definitions A.1.3 and Definition A.2.2, we note that $(\forall\#)$ has vanished, and that $(\forall\vee)$ is not a perfect dual to $(\forall\wedge)$.

The disappearance of $(\forall\#)$ is explained in Proposition A.2.4.

The lack of duality between $(\forall\wedge)$ and $(\forall\vee)$ is natural; $(\forall\leq)$ orients $\forall$ with respect to $\leq$ so it would be surprising (and wrong) if $\forall$ displayed perfectly dual behaviour with respect to $\vee$.[30]

Proposition A.2.4. *Suppose $\mathcal{L}$ is a bounded lattice with $\forall$ and $x \in |\mathcal{L}|$. Then $(\forall\#)$ from Definition A.1.3 follows from the other axioms of a generalised universal quantifier on a bounded lattice.*

That is, $a\#x$ implies $\forall a.x = x$

Proof. We note that $\forall a.x = \forall a.(x \vee x) \overset{(\forall\vee)}{=} (\forall a.x) \vee x$ so that $x \leq \forall a.x$. Furthermore by $(\forall\leq)$ $\forall a.x \leq x$, and we are done. □

Corollary A.2.5. *Suppose $\mathcal{L}$ is a bounded lattice with a generalised universal quantifier $\forall$. Then $\forall a.x$ is the $a\#$limit for $\{x\}$.*

Proof. By Proposition A.2.4 $(\forall\#)$ holds, so the restriction of $\mathcal{L}$ to $\top$, $\wedge$, and $\forall$ forms a bounded meet-semilattice, and it is a fact that this has the same partial order. We use Proposition A.1.5. □

Axioms of the type presented in Definition A.2.2 underlie the nominal algebraic axiomatisation of first-order quantification in e.g. [GM08c; Gab14].

A.3. FOLeq algebras algebraically

We are now ready to give a full axiomatisation of a FOLeq algebra, in nominal algebra in the sense of the formal language described in [GM09]. This nominal algebra theory builds on the nominal algebra axiomatisations of substitution and first-order logic originally developed in [GM06a; GM06b; GM08a; GM08c].

[30] This is to be contrasted with the axiomatisation of and Stone duality result for the $\mathsf{И}$ quantifier in [GLP11]. The $\mathsf{И}$ quantifier is self-dual, and *is* symmetric in its interaction between $\wedge$ and $\vee$.

THEOREM A.3.1. *Suppose $\mathcal{U}$ is a termlike σ-algebra. The FOLeq algebras over $\mathcal{U}$ (Definition 4.4.3) are precisely characterised as an underlying nonempty nominal set $\mathcal{L}$ along with*

— an equivariant σ-action $\mathsf{sub} : (\mathcal{L} \times \mathbb{A} \times \mathcal{U}) \Rightarrow \mathcal{L}$*, written infix* $x[a \mapsto u]$*;*
— an equivariant function $\wedge : \mathcal{L} \times \mathcal{L} \Rightarrow \mathcal{L}$*;*
— an equivariant function $\neg : \mathcal{L} \Rightarrow \mathcal{L}$*,*
— an equivariant function $\forall : (\mathbb{A} \times \mathcal{L}) \Rightarrow \mathcal{L}$*, and*
— an equivariant function $=^{\mathcal{L}} : (\mathcal{U} \times \mathcal{U}) \Rightarrow \mathcal{L}$*,*

such that:

(*1*) sub *satisfies the axioms of a σ-action over $\mathcal{U}$ from Figure 1 and*

$$\begin{aligned} (x \wedge y)[a \mapsto u] &= (x[a \mapsto u]) \wedge (y[a \mapsto u]) \\ (\neg x)[a \mapsto u] &= \neg(x[a \mapsto u]) \\ b\#u \Rightarrow \quad (\forall b.y)[a \mapsto u] &= \forall b.(y[a \mapsto u]), \\ (v' =^{\mathcal{L}} v)[a \mapsto u] &= (v'[a \mapsto u]) =^{\mathcal{L}} (v[a \mapsto u]) \end{aligned}$$

(*2*) *if we take $\top = \neg(x \wedge \neg x)$ (for any x) and $x \vee y = \neg(\neg x \wedge \neg y)$ and $\bot = \neg\top$ then $\top$, $\wedge$, $\bot$, and $\vee$ satisfy the axioms of a bounded lattice from Definition A.2.1,*
(*3*) *in addition*[31]

$$\begin{aligned} x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) \\ \neg\neg x &= x, \end{aligned}$$

(*4*) *$\forall$ satisfies the axioms of a generalised universal quantifier from Figure 4, and finally*
(*5*) *$=^{\mathcal{L}}$ satisfies the axioms*

$$\begin{aligned} (u =^{\mathcal{L}} u) &= \top \\ (u =^{\mathcal{L}} v) \wedge z[a \mapsto u] &= (u =^{\mathcal{L}} v) \wedge z[a \mapsto v] \end{aligned}$$

Proof. We have all the pieces, we just need to put them together.

Suppose $\mathcal{L}$ is a FOLeq algebra. We take $\top$ to be the top element, $x \wedge y$ to be the limit of $\{x, y\}$, $\forall a.x$ to be $\bigwedge^{\#a} x$ the $a\#$limit of $\{x\}$, $\neg x$ to be the complement of x in $\mathcal{L}$ (which is unique by Lemma 4.1.7), and $u =^{\mathcal{L}} v$ to be $(a=b)[a \mapsto u, b \mapsto v]$ where $=^{\mathcal{L}}$ is the equality of $\mathcal{L}$ (the simultaneous σ-action is from Definition 3.5.2). Then:

— By assumption in Definition 4.4.3 the σ-action satisfies the axioms of Figure 1.
— By properties of limits and complements $\top$, $\wedge$, $\bot$, and $\vee$ satisfy the axioms of a bounded lattice.
— By assumption $\mathcal{L}$ is distributive (Definition 4.4.1) and complemented, and it follows that $\wedge$ distributes over $\vee$ and $\neg\neg x = x$.
— By Lemma 4.1.9 $\forall$ satisfies $(\forall\alpha)$ from Figure 4. By routine calculations on limits $\forall$ satisfies $(\forall\wedge)$.[32] By assumption $\mathcal{L}$ is distributive (Definition 4.4.1), so $(\forall\vee)$ holds. By the definition of a fresh-finite limit $\forall a.x \leq x$.

Now suppose $\mathcal{L}$ has all the structure of the statement of this theorem. Definition 4.4.3 defines a FOLeq algebra to be a finitely fresh-complete complemented nominal poset with a compatible σ-algebra structure and an equality. So we consider each property in turn:

— *Finitely fresh-complete.* By standard arguments using the axioms $\top$ is a top element and $x \wedge y$ is a limit for $\{x, y\}$. By Corollary A.2.5 $\forall a.x$ is an $a\#$limit for $\{x\}$. It follows by Proposition 4.1.5 that $\mathcal{L}$ is finitely fresh-complete.
— *Complements.* Again it follows from the axioms that $\neg$ is a complement.
— *Compatible σ-algebra structure.* By assumption sub satisfies the axioms from Figure 1, so $\mathcal{L}$ has a σ-algebra structure of $\mathcal{U}$. By our assumptions (condition 1 in the statement of this theorem) it is compatible.

[31] This axiomatisation is not minimal, but it is readable. The interested reader is referred elsewhere [MVF+02] for what can be achieved in terms of trading off the number of axioms against readability.
[32] It is not quite that simple; we are using *fresh-finite* limits. But it all works out.

— *Equality.* It follows from our assumptions that $a{=^{\mathcal{L}}}b$ is an equality in the sense of Definition 4.3.1.[33] □

B. ON THE LEIBNITZ EQUALITY

REMARK B.0.2. In Remark 6.3.2 we noted that Definition 6.3.1 is a form of Leibnitz equality: $p \in (a =^{\mathcal{P}} b)$ when it 'cannot distinguish between a and b'.

Could it be that $(a =^{\mathcal{P}} b)$ is also equal to $\{p{\in}|\mathcal{P}| \mid (a\ b){\cdot}p = p\}$? After all, if $(a\ b){\cdot}p = p$ then p also 'cannot distinguish a and b'.

This notion is too weak. To see this intuitively, consider that the unordered pair $\{a, b\}$ is invariant under swapping a and b, but is not invariant under converting b into a to obtain $\{a\}$.

We note that a strengthened version of Definition 6.3.1 is possible:

LEMMA B.0.3. $p \in (u =^{\mathcal{P}} v)$ *if and only if for* all *atoms* c, $p[u{\leftarrow\!\shortmid}c] = p[v{\leftarrow\!\shortmid}c]$.

In symbols: $(u =^{\mathcal{P}} v) = \{p{\in}|\mathcal{P}| \mid \forall c.p[u{\leftarrow\!\shortmid}c] = p[v{\leftarrow\!\shortmid}c]\}$.

Proof. Clearly if for *all* atoms c, $p[u{\leftarrow\!\shortmid}c] = p[v{\leftarrow\!\shortmid}c]$, then for all but finitely many c, $p[u{\leftarrow\!\shortmid}c] = p[v{\leftarrow\!\shortmid}c]$.

Conversely, suppose $p[u{\leftarrow\!\shortmid}c] = p[v{\leftarrow\!\shortmid}c]$ for all but finitely many c, and consider any atom a and element x. Now choose one c such that $c\#u, v, x$ and $p[u{\leftarrow\!\shortmid}c] = p[v{\leftarrow\!\shortmid}c]$ (this is always possible because all but finitely many c satisfy each condition). By $(\sigma\alpha)$ $x[a{\mapsto}u] = ((c\ a){\cdot}x)[c{\mapsto}u]$ and $x[a{\mapsto}v] = ((c\ a){\cdot}x)[c{\mapsto}v]$ and it follows using Proposition 3.2.2 repeatedly that $x \in p[u{\leftarrow\!\shortmid}a]$ if and only $x \in p[v{\leftarrow\!\shortmid}a]$. □

REMARK B.0.4. Given Lemma B.0.3, could Definition 6.3.1 use $\forall c$ instead of the less familiar $И c$?

Yes, but the use of the И-quantifier is preferable. It is a weaker proof-obligation; it is strictly easier to prove $И c.p[u{\leftarrow\!\shortmid}c] = p[u{\leftarrow\!\shortmid}c]$ than it is to prove $\forall c.p[u{\leftarrow\!\shortmid}c] = p[u{\leftarrow\!\shortmid}c]$, because we have to worry about 'fewer values' for c.

This is exploited immediately after Definition 6.3.1 in Lemma 6.3.4 when we use two И-quantifiers, one for a' and one for c. If we had taken Lemma B.0.3 as our definition instead, then we would have had to worry about the case $a' = c$; something similar happens in Proposition 6.3.8.

Thus, the И form used in Definition 6.3.1 is more elementary and convenient, though it is indeed logically equivalent to the $\forall$ form of Lemma B.0.3.

We mention that Definitions 3.3.2 and 6.3.1 have a kind of purely nominal precedent:

LEMMA B.0.5. *Suppose* $\mathcal{X}$ *is a nominal set and* $X, Y{\in}|NomPow(\mathcal{X})|$ *are finitely supported subsets of* $\mathcal{X}$ *(Subsection 2.4.1), and suppose* $a\#X$ *and* $a\#Y$. *Write* $X_{\#a} = \{x{\in}X \mid a\#x\}$ *and* $Y_{\#a} = \{y{\in}Y \mid a\#y\}$. *Then*

$$X = Y \quad \textit{if and only if} \quad X_{\#a} = Y_{\#a}.$$

Proof. The left-to-right implication is immediate. For the right-to-left implication it suffices to prove that $X \subseteq Y$. So suppose $x \in X$. Choose fresh b; then $(b\ a){\cdot}x \in X$ and by Proposition 2.3.4 $a\#(b\ a){\cdot}x$ so $(b\ a){\cdot}x \in Y$ so that $x \in Y$. □

For more on this, see [Gab11, Subsection 9.5].

[33] Note that the statements are slightly different; Definition 4.3.1 assumes an element $a{=^{\mathcal{L}}}b$ whereas above we assume an equivariant function $=^{\mathcal{L}}$ and in condition 1 we assume that substitution distributes over it.