# Equational Reasoning in Languages with Binders via Permutation Fixed-Points

**Ali K. Caires-Santos**
University of Brasília, Brasília, Brazil

**Maribel Fernández**
King's College London, London, United Kingdom

**Murdoch James Gabbay**
Heriot-Watt University, Edinburgh, United Kingdom

**Daniele Nantes-Sobrinho**
University of Brasília, Brasília, Brazil

## Abstract

Equational reasoning with binders and structural congruence is difficult due to the interaction between name binding and algebraic laws. Equational theories such as commutativity induce forms of permutation invariance on names that are not captured by standard approaches to the formalisation of syntax with binders.

We show that in the nominal setting, this limitation can be addressed by using generalised permutation fixed-point constraints to make invariance explicit. This yields a uniform framework for reasoning about equality of nominal terms modulo $\alpha$-equivalence and arbitrary equational theories. We introduce a proof system and show that it is sound and complete with respect to a nominal-set semantics, which explains how symmetry can be internalised via fixed-point constraints viewed as И-quantified stabiliser conditions. We provide examples in Milner's $\pi$-calculus – a well-known model of concurrent computation that includes binders and non-trivial structural congruences.

**2012 ACM Subject Classification** Theory of computation → Logic and verification; Theory of computation → Lambda calculus; Theory of computation → Equational logic and rewriting

**Keywords and phrases** Binding, alpha-equivalence, nominal algebra, permutation fixed-point

**Digital Object Identifier** 10.4230/LIPIcs.FSCD.2026.10

**Funding** This paper was partially funded by Royal Society IES\R3\243187.

**Acknowledgements** We thank the anonymous referees for their valuable comments and suggestions, which have substantially improved the quality of this paper.

## 1 Introduction

Formal reasoning modulo an equivalence relation is a challenging task. Most formal developments proceed by working with chosen representatives of equivalence classes, which requires frequent – and often manual – transitions between equivalent representatives during computations. For example, in proof assistants, one typically has to explicitly apply associativity and commutativity laws to reshape expressions so that hypotheses, lemmas, or assertions in the context become applicable. To the best of our knowledge, the Maude system [8] is the only framework that addresses this difficulty by handling commutative, associative, and unit axioms transparently within the rewriting engine [10, 18].

In this work, we address an even more general problem: reasoning modulo equational theories in languages that also involve binders (e.g., separation logic [20, 24], $\pi$-calculus [19], first-order logic, etc). In this setting, one must simultaneously account for both $\alpha$-equivalence classes and the underlying equational theories. Things get more interesting because it turns


11th International Conference on Formal Structures for Computation and Deduction (FSCD 2026).
Editor: Frank Pfenning; Article No. 10; pp. 10:1–10:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

out that binding and equational theories interact in unexpected and non-obvious ways. To see why, consider trying to find solutions to the unification problem

$$\nu a.\nu b.X = \nu b.\nu a.X,$$

where $\nu$ is a binder (think: $\pi$-calculus restriction) and our syntax includes a commutative operator $+$ with equational theory $X + Y = Y + X$ (think: nondeterministic choice, or possibly parallel composition). Here $a$ and $b$ are *names* (also called *atoms*) in the sense of *nominal terms* [26]: they represent abstractable names (think: $\pi$-calculus channel names). $X$ is an *unknown* also in the sense of nominal terms: it represents an unknown term.[1]

What solutions exist to this problem? *Without* an equational theory there is just one family of solutions: instantiate $X$ to any $t$ such that $a$ and $b$ are not free in $t$. *With* an equational theory that includes commutativity of $+$, other families of solutions come into existence: we could take $X = a + b$, or $X = (a + a) + (b + b)$, and so on. We now observe that an elegant way exists to finitely describe *all possible solutions* to the problem $\nu a.\nu b.X = \nu b.\nu a.X$: take any $X$ that is *invariant* under swapping the names $a$ and $b$, i.e., $(a\ b) \cdot X = X$, where $(a\ b)$ is a *swapping* permutation, i.e., a bijection on atoms that maps $a$ to $b$, $b$ to $a$, and any other atom $d$ to itself.[2]

This motivates us to consider equational reasoning based on permutation fixed-point constraints, expressed as axioms of the form $\Phi \Rightarrow s = t$ where $\Phi$ is a set of *primitive permutation fixed-point constraints.*

For example, in the $\pi$-calculus [19], in addition to the commutativity of $+$, Milner specified a *structural congruence* including axioms such as

$$\nu x.(P|Q) = (P|\nu x.Q) \ \text{ if } x \notin fn(P)$$

where the condition can be formally expressed using swappings (intuitively, the condition in this axiom amounts to saying that swapping $x$ with a fresh name does not change $P$). In the format above, this is written:

$$\text{И}c.\ (x\ c) \cdot P = P\ \vdash\ \nu x.(P|Q) = (P|\nu x.Q),$$

where $\text{И}c.\,\varphi(c)$ informally expresses that the property $\varphi(c)$ holds for all but finitely many atoms $c$.

Our main contribution is a proof system for reasoning about equality modulo $\alpha$-equivalence and arbitrary nominal equational theories, obtained by generalising freshness constraints to И-quantified permutation fixed-points. We show that the standard nominal set semantics of nominal algebra [15] extends to this setting naturally, and we establish soundness and completeness of the system with respect to this semantics. This yields a robust algebraic foundation in which freshness is captured as invariance. We illustrate the expressive power of the system with examples inspired by the $\pi$-calculus.

## Related Work

To compute and reason about systems with binding, several approaches for the representation of languages with binders have been proposed, such as *de Bruijn indices* [9], *higher-order abstract syntax (HOAS)* [11, 21], the *locally nameless representation* [7] and *nominal abstract*

[1] Nominal terms are a formal syntax used in the theory of unification, rewriting, and algebra with $\alpha$-converting binders. Nominal terms have specific technical properties, but those properties are *not* required for this discussion. The interested reader can find details in [12, 13, 26].

[2] It is a basic observation of nominal terms that swappings are a useful primitive for handling $\alpha$-equivalence.

*syntax* [16]. These approaches offer techniques for the definition of data structures that take into account $\alpha$-equivalence. More recently, nominal techniques have been employed to formulate general criteria for inductively defined rule-based systems, enabling inductive proofs that exploit Barendregt's variable convention [27], as well as recursion principles for syntactic structures with bindings [23].

In many application domains, $\alpha$-equivalence is not the only equivalence relation that has to be considered. Amongst the above-mentioned approaches, the nominal approach stands out by its first-order flavour, which makes it a natural candidate to combine it with well-established equational reasoning techniques based on term rewriting.

*Nominal universal algebra* [15] provides a framework for equational reasoning in this setting, incorporating $\alpha$-equivalence. Its core syntactic objects are *nominal terms*, which feature a clear distinction between *atoms* $\mathbb{A} = \{a, b, c, \ldots\}$ (representing object-level variables) and *unknowns* $\mathbb{V} = \{X, Y, Z, \ldots\}$ (representing meta-level variables), along with an explicit *abstraction construct* $[a]t$ for binding. This syntax incorporates *suspensions* $\pi \cdot X$, which delay the application of permutations to meta-variables, and *freshness constraints* $a\#t$ to capture the notion of a name being "not free in" a term. Semantics is defined via *nominal sets*, where each element has a *finite support* $\mathtt{supp}(x)$ (the set of atoms $x$ depends on). In this semantics, the freshness constraint $a\#x$ holds precisely when the atom $a$ is not in the support of $x$, i.e., $a \notin \mathtt{supp}(x)$. Essentially, this can be defined positively using the И-*quantifier* (see [16, Eq. 13]): $a\#x$ if and only if $\text{И}\boldsymbol{c}.(a\ \boldsymbol{c}) \cdot x = x$. This formulation expresses freshness not as an absence, but as an invariance under swapping with almost any atom.

This perspective inspired the work of Ayala et al. [1], who replaced primitive *freshness constraints* $a\#X$ with primitive *fixed-point constraints* $\pi \cdot X \approx X$, abbreviated $\pi \curlywedge X$, or $\pi \curlywedge_{\mathtt{C}} X$ if working modulo commutativity as well as $\alpha$-equality. In the presence of commutative symbols, fixed-point constraints do not correspond to freshness constraints: for example, although $(a\ b) \cdot (a + b) \approx_{\mathtt{C}} a + b$, neither $a$ nor $b$ is fresh for $a + b$. The derivation system in [1] is sound and complete with respect to strong nominal set semantics [6]. To address this limitation, a recent approach [5] introduced *fixed-point constraints with nested* И-*quantifiers* of the form $\text{И}\bar{\boldsymbol{c}}.\pi \curlywedge_{\mathtt{C}} X$, where the atoms in $\bar{\boldsymbol{c}}$ may occur in the permutation $\pi$.

In this paper, we generalise the approach to any equational theory defined by equational axioms (which may have fixed-point permutation conditions), not just the theory of commutativity. This required a new set of derivation rules to incorporate arbitrary axioms, as well as new proofs of soundness and completeness. The expressive power of the equational theories considered here is illustrated by examples taken from the $\pi$-calculus, which includes both binders and an involved structural congruence beyond commutativity.

### Contributions and Overview of the paper

Section 2 summarises preliminary definitions and results. The main ones are: (i) classification of permutation cycles (Definition 2.3.2), (ii) generalised fixed-point properties (Lemma 2.3.4 and Theorem 2.3.5), and (iii) concrete examples showing why fixed-point constraints are more expressive than freshness alone. Section 3 introduces our core derivation system for equality in Figure 1. The main definitions include fixed-point constraints and contexts (Definition 3.1.1) that unify freshness and pure fixed-point constraints. An unexpected mathematical insight is how normalisation rules $\mathbf{R}_1 - \mathbf{R}_3$ ensure contexts form groups. We provide concrete examples throughout, including theories for commutativity, $\lambda$-calculus and $\pi$-calculus (Example 3.2.2). Section 4 establishes soundness (Theorem 4.1.3) and completeness (Theorem 4.3.1) using nominal sets. While soundness follows by induction as in freshness approaches [15], completeness requires an extension of the theory $\mathtt{T}$ to handle fixed-point constraints. Finally, Section 5 concludes and discusses future work.

## 2 Preliminaries

This section summarises the main definitions and results used throughout the text. We assume basic knowledge of group theory [4, 25].

### 2.1 Permutations and nominal terms

We fix disjoint countably infinite sets: the set of **atoms** $\mathbb{A} = \{a, b, c, \dots\}$ and **unknowns** $\mathbb{V} = \{X, Y, Z, \dots\}$. A **signature** $\Sigma$ contains function symbols $\mathtt{f}\!:\!n, \mathtt{g}\!:\!m, \dots$ with **arities** (for some $n, m, \dots$ non-negative integers), disjoint from $\mathbb{A}$ and $\mathbb{V}$.

A **(finite) permutation** on $\mathbb{A}$ is a bijection $\pi : \mathbb{A} \to \mathbb{A}$ whose set of **moved atoms** $\mathtt{mv}(\pi) = \{a \in \mathbb{A} \mid \pi(a) \neq a\}$ is finite. Write $\pi \circ \pi'$, $\mathtt{id}$ and $\pi^{-1}$ for functional composition, the identity, and the inverse of permutations, respectively.

The group of all finite permutations of atoms is denoted $\mathcal{G}(\mathbb{A})$. Every finite permutation $\pi \in \mathcal{G}(\mathbb{A})$ can be uniquely[3] decomposed (up to reordering) into pairwise disjoint cycles as $\pi = \gamma_1 \circ \cdots \circ \gamma_n$, with $\mathtt{mv}(\gamma_i) \cap \mathtt{mv}(\gamma_j) = \emptyset$ for $i \neq j$. This is called the **canonical form** of $\pi$.

▶ **Assumption 2.1.1.** Permutations are always in a canonical form.

The set of **nominal terms** $\mathsf{T}(\Sigma, \mathbb{V}, \mathbb{A})$ is defined by the grammar:

$$t ::= a \mid \pi \cdot X \mid \mathtt{f}(t_1, \dots, t_n) \mid [a]t$$

Nominal terms are built from atoms, suspensions[4] of the form $\pi \cdot X$ (also called unknowns or variables, with $\mathtt{id} \cdot X$ written simply as $X$), function applications, and abstractions $[a]t$. An abstraction $[a]t$ models the binding of the atom $a$ in $t$, as in constructs such as $\lambda a.t$ or name restriction.

Occurrences of $a$ in $t$ are said to be **abstracted** (or bound), while occurrences of other atoms are **unabstracted** (or free). For a term $t$, $\mathtt{Var}(t)$ and $\mathtt{Atm}(t)$ denote the sets[5] of unknowns and atoms occurring in $t$, respectively. The set $\mathtt{Atm}(t)$ includes all atoms, abstracted or not, so $\mathtt{Atm}([a]t) = \{a\} \cup \mathtt{Atm}(t)$. When $\mathtt{Var}(t) = \emptyset$, $t$ is called a *ground term* (i.e., a term without unknowns); we write $\mathsf{T}(\Sigma, \mathbb{A})$ for the set of ground terms.

▶ **Remark 2.1.2.** We work with raw syntax, where terms are not quotiented by $\alpha$-equivalence. In particular, syntactic identity $t \equiv u$ is purely structural, and abstractions are not identified up to renaming of bound atoms; for example, $[a]a \not\equiv [b]b$. Instead, $\alpha$-equivalence is treated as a derived relation on top of $\equiv$, defined via an appropriate notion of context introduced later.

The following examples illustrate the expressiveness of the nominal language and demonstrate how binding constructs can be represented using nominal abstraction.

▶ **Example 2.1.3** (Untyped Lambda Calculus). Consider $\Sigma_\lambda = \{\mathtt{lam}\!:\!1, \mathtt{app}\!:\!2\}$, where $\mathtt{lam}$ encodes abstraction and $\mathtt{app}$ encodes application. Object variables $(x, y, \dots)$ are identified with

[3] This is a classical result (see, for example, [25]).
[4] Also called *moderated variables*: the permutation $\pi$ is delayed until $X$ is instantiated.
[5] These operators extend naturally to more general syntactic objects such as constraints and contexts.

atoms ($a_x, a_y, \ldots$), and meta-variables ($M, N, \ldots$) with unknowns ($X_M, X_N, \ldots$). Binding is represented by nominal abstraction. The encoding $(\cdot)_\lambda$ is:

$$\begin{array}{rlrl} (x)_\lambda &= a_x & (M)_\lambda &= X_M \\ (t\,u)_\lambda &= \mathtt{app}(t_\lambda, u_\lambda) & (\lambda x.t)_\lambda &= \mathtt{lam}([a_x]t_\lambda) \end{array}$$

Example: $(\lambda x.(\lambda y.M)x)_\lambda = \mathtt{lam}([a_x]\mathtt{app}(\mathtt{lam}([a_y]X_M), a_x))$.

▶ **Example 2.1.4** ($\pi$-Calculus)**.** Let $\Sigma_\pi = \{\mathtt{0}:0, \mathtt{tau}:1, \mathtt{par}:2, \mathtt{sum}:2, \mathtt{match}:3, \mathtt{in}:2, \mathtt{out}:3, \nu:1\}$. Atoms ($a_x, a_y, \ldots$) represent channel names, and unknowns ($X_P, X_Q, \ldots$) represent process variables. Binding (input and restriction) is encoded via nominal abstraction. The encoding $(\cdot)_\pi$ is:

$$\begin{array}{rlrlrl} \mathbf{0}_\pi &= \mathtt{0} & (P)_\pi &= X_P & (\tau.p)_\pi &= \mathtt{tau}(p_\pi) \\ (p \mid q)_\pi &= \mathtt{par}(p_\pi, q_\pi) & (p+q)_\pi &= \mathtt{sum}(p_\pi, q_\pi) & ([x=y]p)_\pi &= \mathtt{match}(a_x, a_y, p_\pi) \\ (x(y).p)_\pi &= \mathtt{in}(a_x, [a_y]p_\pi) & (\overline{x}y.p)_\pi &= \mathtt{out}(a_x, a_y, p_\pi) & (\nu x.p)_\pi &= \nu([a_x]p_\pi) \end{array}$$

Example: $(\nu x.(\overline{y}x.P))_\pi = \nu([a_x]\mathtt{out}(a_y, a_x, X_P))$.

## 2.2 Permutation actions, substitutions, and nominal sets

We now define permutation actions over terms. Intuitively, a permutation $\pi$ propagates through the structure of $t$, acting directly when atoms or suspensions are encountered.

1. Define $\pi \cdot t$ as an **object-level permutation action** on terms by:

$$\begin{array}{rlrl} \pi \cdot a &\equiv \pi(a) & \pi \cdot (\pi' \cdot X) &\equiv (\pi \circ \pi') \cdot X \\ \pi \cdot [a]t &\equiv [\pi(a)](\pi \cdot t) & \pi \cdot \mathtt{f}(t_1, \ldots, t_n) &\equiv \mathtt{f}(\pi \cdot t_1, \ldots, \pi \cdot t_n) \end{array}$$

2. Define $t^\pi$ as a **meta-level permutation action** on terms by:

$$\begin{array}{rlrl} a^\pi &\equiv \pi(a) & (\rho \cdot X)^\pi &\equiv \rho^\pi \cdot X \\ ([a]t)^\pi &\equiv [\pi(a)]t^\pi & (\mathtt{f}(t_1, \ldots, t_n))^\pi &\equiv \mathtt{f}(t_1^\pi, \ldots, t_n^\pi) \end{array}$$

   where $\rho^\pi = \pi \circ \rho \circ \pi^{-1}$ is the **conjugate** of $\rho$ by $\pi$.

A **substitution** $\sigma$ is a mapping from unknowns to terms such that the set $\{X \in \mathbb{V} \mid X\sigma \not\equiv X\}$ is finite. The **substitution action** $t\sigma$ is defined inductively:

$$\begin{array}{rlrl} a\sigma &\equiv a & (\pi \cdot X)\sigma &\equiv \pi \cdot (X\sigma) \\ ([a]t)\sigma &\equiv [a](t\sigma) & \mathtt{f}(t_1, \ldots, t_n)\sigma &\equiv \mathtt{f}(t_1\sigma, \ldots, t_n\sigma) \end{array}$$

We write $Id$ for the identity substitution and $[X \mapsto u]$ for the substitution mapping $X$ to $u$. Substitution composes left-to-right: $X(\sigma\rho) \equiv (X\sigma)\rho$. Key properties:

1. **No capture-avoidance**: substitution on unknowns is a first-order operation and does not avoid capture. For instance, $([a]X)\sigma \equiv [a]\mathtt{f}(a, b)$, for $\sigma = [X \mapsto \mathtt{f}(a, b)]$. This is intentional: substitution is purely syntactic, while $\alpha$-equivalence is handled separately (see [12, Sec. 2.2]).
2. **Commutation with permutations**: $\pi \cdot (t\sigma) \equiv (\pi \cdot t)\sigma$

The semantic counterpart of nominal syntax is given by nominal sets, which formalise the concept of "name dependence".

A **nominal set** is a pair $(\mathscr{X}, \cdot)$ consisting of a set $\mathscr{X}$ and an action $\cdot : \mathcal{G}(\mathbb{A}) \times \mathscr{X} \to \mathscr{X}$ such that every element $x \in \mathscr{X}$ has **finite support**, i.e., there exists a finite $B \subseteq \mathbb{A}$ with $\pi \cdot x = x$ for all $\pi \in \mathcal{G}(\mathbb{A})$ such that $\pi(a) = a$ for every $a \in B$. The least such $B$ is denoted $\mathtt{supp}(x)$. An atom $a$ is **fresh** for $x$ (written $a\#x$) if $a \notin \mathtt{supp}(x)$.

Sometimes, when there is no ambiguity, we refer to a nominal set $(\mathscr{X}, \cdot)$ by its carrier set $\mathscr{X}$ and leave the action $\cdot$ implicit.

▶ **Example 2.2.1.** Examples of nominal sets include:
1. The set of atoms $\mathbb{A}$, with the action $\pi \cdot a = \pi(a)$, where $\mathtt{supp}(a) = \{a\}$;
2. The set of ground terms $\mathsf{T}(\Sigma, \mathbb{A})$ with object-level permutation action, where $\mathtt{supp}(t) = \mathtt{Atm}(t)$, e.g., $\mathtt{supp}([a]\mathtt{f}(b, d)) = \{a, b, d\}$;
3. The group $\mathcal{G}(\mathbb{A})$ under conjugation, defined by $\pi \cdot \rho = \rho^{\pi}$, where $\mathtt{supp}(\pi) = \mathtt{mv}(\pi)$;
4. The set $\mathcal{P}_{\mathtt{fin}}(\mathbb{A})$ of finite subsets of atoms, with pointwise action $\pi \cdot B = \{\pi(a) \mid a \in B\}$, where $\mathtt{supp}(B) = B$.

## 2.3 The И-quantifier and fixed-point identities

The И-quantifier is one of the most powerful tools in nominal techniques. Introduced by Gabbay and Pitts [16], it provides a rigorous mathematical foundation for the informal notion of "choosing a fresh name" that appears frequently when reasoning within structures with binding constructs. Proofs of the results in this section are given in the Appendix A.

Throughout this section, let $\mathscr{X}$ denote a nominal set. We consider formulas of the form $\varphi(\boldsymbol{c}, \overline{x})$[6], where $\boldsymbol{c}$ is an atom and $\overline{x} = x_1, \ldots, x_n$ is a finite list of parameters with each $x_i \in \mathscr{X}$. Intuitively, the assertion $\text{И}\boldsymbol{c}.\varphi(\boldsymbol{c}, \overline{x})$ holds when $\varphi(\boldsymbol{c}, \overline{x})$ is satisfied for all but finitely many atoms $\boldsymbol{c} \in \mathbb{A}$. We adopt the visual convention of writing И-quantified atoms in **teal** [7] for visual distinction.

▶ **Definition 2.3.1.** The **И-quantifier** is formally defined as:

$$\text{И}\boldsymbol{c}.\varphi(\boldsymbol{c}, \bar{x}) \iff \{\boldsymbol{c} \in \mathbb{A} \mid \varphi(\boldsymbol{c}, \bar{x})\} \text{ is cofinite}^{8} \tag{1}$$

Thus, the И-quantifier captures the *generative* aspect of fresh name creation.

In cases where multiple fresh atoms are introduced at once, we consider nested И-quantification of the form $\text{И}\boldsymbol{c_1} \ldots \text{И}\boldsymbol{c_n}.\varphi(\boldsymbol{c_1}, \ldots, \boldsymbol{c_n}, \overline{x})$, abbreviated as $\text{И}\bar{\boldsymbol{c}}.\varphi(\bar{\boldsymbol{c}}, \overline{x})$. Further details on the meaning of nested И-quantification are given in the Appendix A.

A fundamental connection between the И-quantifier and freshness is given by the following equivalence [16]:

$$a\#x \iff \text{И}\boldsymbol{c}.(a\ \boldsymbol{c}) \cdot x = x. \tag{2}$$

This characterisation generalises to arbitrary permutations involving freshly generated atoms. We define a **generalised fixed-point identity** as an assertion of the form:

$$\text{И}\bar{\boldsymbol{c}}.\pi \cdot x = x,$$

[6] Typically, $\varphi$ is a predicate in higher-order logic [22] or ZFA set theory [14, 16].
[7] The distinction remains clear in black-and-white formats, as these atoms are typeset in bold.
[8] A *cofinite* subset of a set $M$ is a subset $B$ whose complement in $M$ is a finite set.

where $\pi$ is a permutation that may contain atoms bound by $\text{И}$. Intuitively, $\pi \cdot x = x$ holds for all tuples $\bar{\boldsymbol{c}}$ fresh for $x$, i.e. $\bar{\boldsymbol{c}}\#\overline{x}$[8]. In this case, $x$ is *invariant under* $\pi$.

By (2), a freshness condition can be expressed as a quantified fixed-point identity, which also captures permutation invariance. This motivates the study of generalised fixed-point identities. For $\bar{\boldsymbol{c}} = \boldsymbol{c_1}, \boldsymbol{c_2}, \boldsymbol{c_3}, \boldsymbol{c_4}$ and

$$\pi = (a_1\ \boldsymbol{c_1}\ \boldsymbol{c_2}) \circ (a_2\ \boldsymbol{c_3}) \circ (a_3\ a_4\ a_5) \circ (a_6\ \boldsymbol{c_4}) \circ (a_7\ a_8),$$

what is the meaning of $\text{И}\bar{\boldsymbol{c}}.\pi \cdot x = x$? Can it be expressed via freshness?

[8] For a finite set of atoms $A$ and a finite list of elements $\overline{x} = x_1, \ldots, x_k$ of $\mathscr{X}$, write $A\#\overline{x}$ as abbreviation for $A \cap \texttt{supp}(\overline{x}) = \emptyset$, where $\texttt{supp}(\overline{x}) = \bigcup_{j=1}^{k} \texttt{supp}(x_j)$. If $\overline{a} = a_1, \ldots, a_n$ is a list of atoms, we write $\overline{a}\#\overline{x}$ as shorthand for $\{\overline{a}\}\#\overline{x}$, where $\{\overline{a}\} = \{a_1, \ldots, a_n\}$.

We analyse identities of the form $\text{И}\bar{c}.\pi \cdot x = x$ by decomposing $\pi$ into disjoint cycles and classifying them according to their interaction with $\bar{c}$.

▶ **Definition 2.3.2** ($\bar{c}$-cycles and $\bar{c}$-permutations)**.** Let $\pi$ be a permutation written as a product of disjoint cycles and let $\bar{c}$ be a list of atoms.

1. A cycle $\eta$ of $\pi$ is called a **$\bar{c}$-cycle** if $\mathtt{mv}(\eta) \cap \{\bar{c}\} \neq \emptyset$, and a **$\neg\bar{c}$-cycle** otherwise.
2. A permutation $\pi$ is a **$\bar{c}$-permutation** iff all its cycles are $\bar{c}$-cycles, and a **$\neg\bar{c}$-permutation** iff all its cycles are $\neg\bar{c}$-cycles.

Write $\pi = \pi_{\bar{c}} \circ \pi_{\neg\bar{c}}$ where $\pi_{\bar{c}}$ is the product of all $\bar{c}$-cycles of $\pi$, and $\pi_{\neg\bar{c}}$ is the product of all $\neg\bar{c}$-cycles of $\pi$.

▶ **Example 2.3.3.** For $\bar{c} = c_1, c_2, c_3, c_4$, the permutation:

$$\pi = (a_1\ c_1\ c_2) \circ (a_2\ c_3) \circ (a_3\ a_4\ a_5) \circ (a_6\ c_4) \circ (a_7\ a_8)$$

decomposes as: $\pi = \underbrace{(a_1\ c_1\ c_2) \circ (a_2\ c_3) \circ (a_6\ c_4)}_{\pi_{\bar{c}}} \circ \underbrace{(a_3\ a_4\ a_5) \circ (a_7\ a_8)}_{\pi_{\neg\bar{c}}}$.

The following lemma shows that if a permutation fixes an element and one of its cycles moves an atom outside its support, then all atoms in that cycle must lie outside the support. It also ensures that permutation invariance is closed under composition. The latter is observed by taking permutations from the group generated[9] by a finite number of permutations that fix an element $x$.

▶ **Lemma 2.3.4.** *Let $x \in \mathscr{X}$ and $\pi, \eta_1, \ldots, \eta_n$ be permutations written as compositions of disjoint cycles.*

1. *If $\pi \cdot x = x$, $\eta$ is a cycle of $\pi$, and $\mathtt{mv}(\eta) - \mathtt{supp}(x) \neq \emptyset$, then $\mathtt{mv}(\eta) \# \mathtt{supp}(x)$.*
2. *If $\eta_i \cdot x = x$, for all $i = 1, \ldots, n$, and $\rho \in \langle \eta_1, \ldots, \eta_n \rangle$, then $\rho \cdot x = x$.*

For example, if $(a_1\ a_2\ a_3) \circ (a_4\ a_5) \cdot x = x$ and $a_2 \notin \mathtt{supp}(x)$, then $a_3, a_1 \notin \mathtt{supp}(x)$.

The next theorem provides answers to the questions induced by the definition of the generalised fixed-point identities.

▶ **Theorem 2.3.5** (Generalised Fix-Point)**.** *Let $x \in \mathscr{X}$ and let $\pi$ be a permutation.*

1. *If $\pi$ is a $\bar{c}$-permutation and $A = \mathtt{mv}(\pi) \setminus \{\bar{c}\}$, then $\text{И}\bar{c}.\pi \cdot x = x$ iff $A \# x$.*
2. *$\text{И}\bar{c}.\pi \cdot x = x$ iff $(\text{И}\bar{c}.\pi_{\bar{c}} \cdot x = x) \wedge (\pi_{\neg\bar{c}} \cdot x = x)$.*
   *Equivalently, if $A = \mathtt{mv}(\pi_{\bar{c}}) \setminus \{\bar{c}\}$, then $\text{И}\bar{c}.\pi \cdot x = x$ iff $A \# x \wedge (\pi_{\neg\bar{c}} \cdot x = x)$.*

Theorem 2.3.5 shows that a generalised fixed-point identity $\text{И}\bar{c}.\pi \cdot x = x$ exhibits two behaviours depending on the decomposition of $\pi$:

**A fresh part.** If $\pi$ is a $\bar{c}$-permutation, the identity enforces freshness: atoms modified by $\pi$ must lie outside $\mathtt{supp}(x)$.

**An algebraic part.** If $\pi$ is a $\neg\bar{c}$-permutation, the identity expresses permutation invariance, independent of $\bar{c}$ and not derivable from freshness, but from the structure of $x$.

▶ **Example 2.3.6** (Fixed-points in $\mathcal{P}_{\mathsf{fin}}(\mathbb{A})$)**.** Let $x \in \mathcal{P}_{\mathsf{fin}}(\mathbb{A})$ and consider the identity $\text{И}c_1, c_2.\ ((a\ c_1) \circ (b\ c_2) \circ (d\ e)) \cdot x = x$. By Theorem 2.3.5, this decomposes as

$$(\text{И}c_1, c_2.((a\ c_1) \circ (b\ c_2)) \cdot x = x) \ \wedge\ (d\ e) \cdot x = x.$$

The first conjunct enforces $a, b \# x$, while the second requires invariance under $(d\ e)$, allowing $d, e \in \mathtt{supp}(x)$. For instance, $x = \{d, e\}$ satisfies $a, b \# x$ and $(d\ e) \cdot x = x$.

[9] For a set $S \subseteq \mathcal{G}(\mathbb{A})$, we write $\langle S \rangle$ for the subgroup generated by $S$, that is, the smallest subgroup of $\mathcal{G}(\mathbb{A})$ containing $S$. In particular, if $S = \{\rho_1, \ldots, \rho_n\}$ is a finite set of permutations, we write $\langle \rho_1, \ldots, \rho_n \rangle$ for $\langle S \rangle$.

In this example, invariance under $(d\ e)$ is not induced by freshness, but by the internal structure of $x$. We call this an *algebraic symmetry*.

We now study how identities of the form $\text{И}\bar{\mathbf{c}}.(\pi \cdot x = x)$ interact under shared quantification, in particular expressions of the form $\text{И}\bar{\mathbf{c}}.(\pi \cdot x = x \wedge \rho \cdot x = x)$.

▶ **Definition 2.3.7** (Cycle Grafting)**.** Let $\pi = \gamma \circ \pi'$ and $\tau = \eta \circ \rho$ be (canonical) permutations, where $\gamma$ is a $\bar{\mathbf{c}}$-cycle of $\pi$ and $\eta$ is a $\neg\bar{\mathbf{c}}$-cycle of $\tau$ such that $\mathtt{mv}(\gamma) \cap \mathtt{mv}(\eta) \neq \emptyset$. We write $\gamma \frown \eta$ for a chosen cycle $\delta$ such that $\mathtt{mv}(\delta) = \mathtt{mv}(\gamma) \cup \mathtt{mv}(\eta)$, noting that the order of atoms in $\delta$ is arbitrary, and hence the grafting is not unique.

We write $\pi[\gamma \mapsto \gamma \frown \eta]$ for the permutation obtained by replacing the cycle $\gamma$ in $\pi$ with $\gamma \frown \eta$, leaving all other cycles unchanged.

The next proposition establishes a property of cycle grafting in fixed-point identities: freshness enforced by the $\bar{\mathbf{c}}$-cycle (say $\gamma$) propagates to the overlapping $\neg\bar{\mathbf{c}}$-cycle (say $\eta$), forcing (fragments of) its algebraic part to be transferred to the $\bar{\mathbf{c}}$-part without changing the semantics.

▶ **Proposition 2.3.8.** *Let $x \in \mathscr{X}$ and let $\pi = \gamma \circ \pi'$ and $\tau = \eta \circ \rho$ as in Definition 2.3.7. Then*

$$\text{И}\bar{\mathbf{c}}.(\pi \cdot x = x \wedge \tau \cdot x = x) \iff \text{И}\bar{\mathbf{c}}.(\pi[\gamma \mapsto \gamma \frown \eta] \cdot x = x \wedge \rho \cdot x = x).$$

▶ **Example 2.3.9.** Let $\gamma = (a\ \mathbf{c})$, $\pi' = (d\ e)$, $\eta = (a\ f)$ and $\rho = (r\ s)$. Since $\mathtt{mv}(\gamma) \cap \mathtt{mv}(\eta) = \{a\}$, a possible grafting is $\gamma \frown \eta = (a\ f\ \mathbf{c})$. Then the equivalence of Proposition 2.3.8 rewrites $\text{И}\mathbf{c}.(((a\ \mathbf{c}) \circ (d\ e)) \cdot x = x \wedge ((a\ f) \circ (r\ s)) \cdot x = x)$ as $\text{И}\mathbf{c}.(((a\ f\ \mathbf{c}) \circ (d\ e)) \cdot x = x \wedge (r\ s) \cdot x = x)$.

Multiple $\bar{\mathbf{c}}$-permutations that share some non-fresh atoms and fix the same element can be combined into a single fixed-point constraint.

▶ **Corollary 2.3.10.** *Let $\pi$ and $\rho$ be $\bar{\mathbf{c}}$-permutations such that $(\mathtt{mv}(\pi) \cap \mathtt{mv}(\rho)) \setminus \{\bar{\mathbf{c}}\} \neq \emptyset$. Then*

$$\text{И}\bar{\mathbf{c}}.(\pi \cdot x = x \wedge \rho \cdot x = x) \iff \text{И}\bar{\mathbf{c}}.(a_1 \dots a_k\ \mathbf{c}') \cdot x = x,$$

*where $\{a_1, \dots, a_k\} = (\mathtt{mv}(\pi) \cup \mathtt{mv}(\rho)) \setminus \{\bar{\mathbf{c}}\}$ (in any order), and $\mathbf{c}' \in \{\bar{\mathbf{c}}\}$.*

▶ **Example 2.3.11.** Let $\pi = (a\ \mathbf{c_1})$ and $\rho = (b\ \mathbf{c_2})$ with $\mathbf{c_1}, \mathbf{c_2} \in \{\bar{\mathbf{c}}\}$. Then $(\mathtt{mv}(\pi) \cap \mathtt{mv}(\rho)) \setminus \{\bar{\mathbf{c}}\} = \{a, b\}$ and $\text{И}\bar{\mathbf{c}}.(\pi \cdot x = x \wedge \rho \cdot x = x)$ is equivalent to $\text{И}\bar{\mathbf{c}}.(a\ b\ \mathbf{c}') \cdot x = x$ for any $\mathbf{c}' \in \{\bar{\mathbf{c}}\}$.

# 3 Derivation system

In this section we introduce a proof system for reasoning modulo $\alpha$-equivalence and an arbitrary nominal equational theory, based on generalised fixed-point constraints.

## 3.1 Constraints and Contexts

▶ **Definition 3.1.1.** An **equality constraint** is a pair $t = u$. A **generalised fixed-point constraint** is an expression $\text{И}\bar{\mathbf{c}}.(\pi \curlywedge t)$, abbreviating $\text{И}\bar{\mathbf{c}}.(\pi \cdot t = t)$. When $\bar{\mathbf{c}}$ is empty we write just $\pi \curlywedge t$. It is called **primitive** when $t \equiv X$.

A **(generalised) fixed-point context** is an expression $\text{И}\bar{\mathbf{c}}.\Upsilon$, where $\Upsilon$ is a finite set of primitive constraints $\{\pi_1 \curlywedge X_1, \dots, \pi_n \curlywedge X_n\}$, read conjunctively.

Intuitively, generalised fixed-point constraints are defined so as to mirror their semantic counterparts in nominal sets, expressing invariance under permutations.

### Context Normalisation

The decomposition of fixed-point identities established in Theorem 2.3.5 suggests that fixed-point contexts admit a more structured representation. In particular, permutations can be separated into two components with distinct roles: $\bar{c}$-permutations, which enforce freshness conditions, and $\neg\bar{c}$-permutations, which capture algebraic symmetry.

Motivated by this observation, we define a rewriting system $\mathcal{R}_{\mathrm{nf}}$ that normalises contexts by making this separation explicit. Rather than combining both behaviours within a single constraint, $\mathcal{R}_{\mathrm{nf}}$ rewrites contexts into a form where each primitive constraint involves only one kind of permutation.

▶ **Definition 3.1.2** (Normalisation process)**.** Consider the rewriting system $\mathcal{R}_{\mathrm{nf}}$ consisting of the following rules[10]:

$(\mathbf{R}_1)$ $\quad \text{И}\bar{c}.(\Upsilon, (\pi_{\bar{c}} \circ \pi_{\neg\bar{c}}) \curlywedge X) \longrightarrow \text{И}\bar{c}.(\Upsilon, \pi_{\bar{c}} \curlywedge X, \pi_{\neg\bar{c}} \curlywedge X)$
where $\pi_{\bar{c}}, \pi_{\neg\bar{c}} \neq \mathtt{id}$

$(\mathbf{R}_2)$ $\quad \text{И}\bar{c}.(\Upsilon, \pi \curlywedge X, (\eta \circ \rho) \curlywedge X) \longrightarrow \text{И}\bar{c}.(\Upsilon, (\pi[\gamma \mapsto \gamma \frown \eta]) \curlywedge X, \rho \curlywedge X)$
where
1. $\gamma$ is a $\bar{c}$-cycle in the decomposition of $\pi$
2. $\eta$ is a $\neg\bar{c}$-cycle satisfying $\mathtt{mv}(\gamma) \cap \mathtt{mv}(\eta) \neq \emptyset$

$(\mathbf{R}_3)$ $\quad \text{И}\bar{c}.(\Upsilon, \pi \curlywedge X, \rho \curlywedge X) \longrightarrow \text{И}\bar{c}.(\Upsilon, (a_1 \ \ldots \ a_k \ c') \curlywedge X)$
where
1. $\pi, \rho$ are $\bar{c}$-permutations such that $(\mathtt{mv}(\pi) \cap \mathtt{mv}(\rho)) \setminus \{\bar{c}\} \neq \emptyset$
2. $a_1, \ldots, a_k \in (\mathtt{mv}(\pi) \cup \mathtt{mv}(\rho)) \setminus \{\bar{c}\}$ (in any order), and $c' \in \{\bar{c}\}$ (arbitrary).

The system $\mathcal{R}_{\mathrm{nf}}$ is sound with respect to the nominal semantics, by the results of § 2.3. Specifically, rule $(\mathbf{R}_1)$ is justified by Theorem 2.3.5, rule $(\mathbf{R}_2)$ by Proposition 2.3.8, and rule $(\mathbf{R}_3)$ by Corollary 2.3.10. Consequently, every rewriting step preserves the semantic meaning of the context. After exhaustive application of the rules, contexts separate into a part $\Upsilon^{\#}$, containing only $\bar{c}$-permutations (encoding freshness), and a part $\Upsilon^{\curlywedge}$, containing only $\neg\bar{c}$-permutations (encoding algebraic symmetries).

▶ **Example 3.1.3.** Consider the context:

$$\text{И}c_1, c_2.\{\underline{(a\ c_1) \circ (b\ d) \curlywedge X}, (a\ e) \curlywedge X, (a\ c_2) \curlywedge X\}$$
$$\longrightarrow_{(\mathbf{R}_1)} \text{И}c_1, c_2.\{\underline{(a\ c_1) \curlywedge X}, (b\ d) \curlywedge X, \underline{(a\ e) \curlywedge X}, (a\ c_2) \curlywedge X\}$$
$$\longrightarrow_{(\mathbf{R}_2)} \text{И}c_1, c_2.\{\underline{(a\ e\ c_1) \curlywedge X}, (b\ d) \curlywedge X, \underline{(a\ c_2) \curlywedge X}\}$$
$$\longrightarrow_{(\mathbf{R}_3)} \text{И}c_1, c_2.\{(a\ e\ c_1) \curlywedge X, (b\ d) \curlywedge X\}$$

▶ **Theorem 3.1.4.** *The rewriting system $\mathcal{R}_{\mathrm{nf}}$ is terminating.*

**Proof sketch.** We prove termination by defining a measure function that maps a context $\Upsilon_{\bar{c}}$ to a triple $(m_1, m_2, m_3)$ of natural numbers where: $m_1$ counts mixed constraints, $m_2$ counts $\neg\bar{c}$-cycles in permutations, and $m_3$ counts constraints with only $\bar{c}$-permutations. Order these triples lexicographically.

Each rule strictly decreases this measure: $(\mathbf{R}_1)$ reduces $m_1$ by removing one mixed constraint; $(\mathbf{R}_2)$ preserves $m_1$ and reduces $m_2$ by eliminating one $\neg\bar{c}$-cycle; $(\mathbf{R}_3)$ preserves $m_1, m_2$ and reduces $m_3$ by merging two $\bar{c}$-constraints into one. ◀

[10] Recall that all permutations are assumed to be in canonical form (Assumption 2.1.1). For readability, we write $\text{И}\bar{c}.(\Upsilon \cup \{\pi \curlywedge X\})$ as $\text{И}\bar{c}.(\Upsilon, \pi \curlywedge X)$.

The next theorem ensures that the order of application of the rules in $\mathcal{R}_{\text{nf}}$ in a context is irrelevant.

▶ **Theorem 3.1.5.** *$\mathcal{R}_{\text{nf}}$ is locally confluent.*

**Proof sketch.** Let $C = \text{И}\bar{c}.\Upsilon$ and suppose $C \to C_1$ and $C \to C_2$. We show that this local peak is joinable, i.e., there exists $C_3$ such that $C_1 \xrightarrow{*} C_3$ and $C_2 \xrightarrow{*} C_3$. Since each rule rewrites only constraints for a single variable $X$, steps on different variables or on disjoint constraints commute, so it suffices to consider peaks where both steps act on the same variable and share a constraint.

Local peaks arise from pairs $(\mathbf{R}_i, \mathbf{R}_j)$, yielding 9 cases; by symmetry it suffices to consider $i \leq j$, giving 6. We illustrate the argument with $(\mathbf{R}_2, \mathbf{R}_2)$. Intuitively, $(\mathbf{R}_2)$ takes a $\neg\bar{c}$-cycle $\eta$ and grafts it into a $\bar{c}$-cycle of a permutation, so two applications correspond to choosing where $\eta$ is absorbed. Consider the case where both steps use $(\eta \circ \rho) \curlywedge X$ but absorb $\eta$ into $\pi_1 \curlywedge X$ or $\pi_2 \curlywedge X$. This is the only non-trivial overlap: all others involve disjoint constraints or independent absorptions and hence commute.

In either branch, $(\eta \circ \rho) \curlywedge X$ disappears and $\eta$ is incorporated into a $\bar{c}$-cycle; then $(\mathbf{R}_3)$ merges the resulting $\bar{c}$-constraints. Regardless of the order, the final constraint contains exactly the atoms moved by $\pi_1$, $\pi_2$, and $\eta$. Hence the two branches are joinable. The remaining cases are simpler variations of the same operations, so every local peak is joinable and $\mathcal{R}_{\text{nf}}$ is locally confluent. ◀

▶ **Corollary 3.1.6.** *$\mathcal{R}_{\text{nf}}$ is convergent. Every normal form is unique up to reordering of disjoint cycles, cyclic rotation within cycles, choice of the И-bound atoms, and reordering of atoms inside $\bar{c}$-cycles.*

**Proof sketch.** By Newman's Lemma [2], termination (Theorem 3.1.4) and local confluence (Theorem 3.1.5) imply that $\mathcal{R}_{\text{nf}}$ is confluent, hence convergent. Uniqueness of normal forms holds modulo harmless choices: reordering disjoint cycles and cyclic rotations do not affect the permutation semantics; the choice of И-bound atoms is immaterial up to renaming, since И quantifies over fresh atoms; and reordering atoms inside $\bar{c}$-cycles yields the same freshness information by Theorem 2.3.5. ◀

▶ **Assumption 3.1.7.** In what follows, all fixed-point contexts are assumed to be in normal form with respect to $\mathcal{R}_{\text{nf}}$.

▶ **Notation 3.1.8.** We use the following notation. Write $\Upsilon_{\bar{c}}$ for $\text{И}\bar{c}.\Upsilon$. For a variable $X$, let $\Upsilon|_X = \{\pi \curlywedge X \mid \pi \curlywedge X \in \Upsilon\}$ be the set of primitive constraints on $X$, and $\Upsilon_{\bar{c}}|_X = \text{И}\bar{c}.\Upsilon|_X$ their quantified form. Let $\mathtt{Atm}(\Upsilon_{\bar{c}})$ be the set of atoms in $\Upsilon_{\bar{c}}$, including those in $\bar{c}$. Also, $\mathtt{perm}(\Upsilon) = \{\pi \mid \pi \curlywedge X \in \Upsilon\}$ is the set of permutations in $\Upsilon$.

We associate to a context $\Upsilon_{\bar{c}}$ and a variable $X$ the set of permutations:

$$\mathtt{P}(\Upsilon_{\bar{c}}|_X) = \{\pi_f \circ \pi_s \mid \pi_f \in \mathcal{G}(\mathsf{Fresh}(\Upsilon_{\bar{c}}|_X)),\ \pi_s \in \mathsf{Sym}(\Upsilon|_X)\}. \tag{3}$$

Here, $\mathcal{G}(\mathsf{Fresh}(\Upsilon_{\bar{c}}|_X)) = \{\pi \in \mathcal{G}(\mathbb{A}) \mid \mathtt{mv}(\pi) \subseteq \mathtt{Atm}(\Upsilon^{\#}_{\bar{c}}|_X)\}$ is the subgroup of $\mathcal{G}(\mathbb{A})$ consisting of permutations that move only atoms that are fresh for $X$, and $\mathsf{Sym}(\Upsilon|_X) = \langle \mathtt{perm}(\Upsilon^{\curlywedge}|_X)\rangle$ is the subgroup of $\mathcal{G}(\mathbb{A})$ generated by symmetry constraints. For brevity, we write $\mathtt{P}(\Upsilon_{\bar{c}}|_X) = \mathcal{G}(\mathsf{Fresh}(\Upsilon_{\bar{c}}|_X)) \circ \mathsf{Sym}(\Upsilon|_X)$.

If $\Upsilon_{\bar{c}}$ is not in normal form, $\mathtt{P}(\Upsilon_{\bar{c}}|_X)$ may fail to be a subgroup of $\mathcal{G}(\mathbb{A})$. For example, let $\Phi_{c} = \text{И}c.\{(a\ c) \curlywedge X,\ (a\ b) \circ (d\ e) \curlywedge X\}$. Then $\mathtt{P}(\Phi_c|_X) = \mathcal{G}(\{a, c\}) \circ \langle (a\ b) \circ (d\ e)\rangle$, which is not closed under composition since $(a\ b) \circ (d\ e) \circ (a\ c) \notin \mathtt{P}(\Phi_c|_X)$.

If $\Upsilon_{\bar{c}}$ is normalised, these two subgroups are disjoint (in the group-theory sense that their intersection contains just the identity) and commute, so $\mathtt{P}(\Upsilon_{\bar{c}}|_X)$ is a group.

$$\frac{}{\Upsilon \vdash^{\bar{c}} t = t}\,(\mathit{refl}) \qquad \frac{\Upsilon \vdash^{\bar{c}} u = t}{\Upsilon \vdash^{\bar{c}} t = u}\,(\mathit{symm}) \qquad \frac{\Upsilon \vdash^{\bar{c}} t = v \qquad \Upsilon \vdash^{\bar{c}} v = u}{\Upsilon \vdash^{\bar{c}} t = u}\,(\mathit{tran})$$

$$\frac{\Upsilon \vdash^{\bar{c}} t = u}{\Upsilon \vdash^{\bar{c}} \mathtt{f}(t_1, \ldots, t, \ldots, t_n) = \mathtt{f}(t_1, \ldots, u, \ldots, t_n)}\,(\mathit{cong}\mathtt{f})$$

$$\frac{\Upsilon \vdash^{\bar{c}} t = u}{\Upsilon \vdash^{\bar{c}} [a]t = [a]u}\,(\mathit{cong}[a]) \qquad \frac{\Upsilon \vdash^{\bar{c},\bar{c}'} \Upsilon'^{\pi}\sigma}{\Upsilon \vdash^{\bar{c}} t^{\pi}\sigma = u^{\pi}\sigma}\,(\mathit{ax}_{\Upsilon' \vdash^{\bar{c}'} t = u})$$

$$\frac{\Upsilon \vdash^{\bar{c},c'} (a\ c') \cdot t = (b\ c') \cdot u}{\Upsilon \vdash^{\bar{c}} [a]t = [b]u}\,(\mathit{cong}[ab]) \qquad \frac{\pi'^{-1} \circ \pi \in \mathtt{P}(\Upsilon_{\bar{c}}|_X)}{\Upsilon \vdash^{\bar{c}} \pi \cdot X = \pi' \cdot X}\,(\mathit{var})$$

$$\frac{\Upsilon, \pi \curlywedge X \vdash^{\bar{c}} t = u}{\Upsilon \vdash^{\bar{c}} t = u}\,(\mathit{fr}) \quad ((\mathtt{mv}(\pi) \setminus \{\bar{c}\}) \cap \mathtt{Atm}(\Upsilon_{\bar{c}}, t, u) = \emptyset)$$

**Figure 1** Inference rules. In $(\mathit{fr})$, $\pi$ is a $\bar{c}$-permutation.

## 3.2 Inference rules

We now introduce the inference rules for equality derivations under fixed-point contexts, relative to a theory $\mathtt{T}$.

▶ **Definition 3.2.1.** An **equality judgement** is an expression of the form $\mathsf{И}\bar{c}.(\Upsilon \vdash t = u)$ (or simply $\Upsilon \vdash^{\bar{c}} t = u$).

A **theory** $\mathtt{T} = (\Sigma, Ax)$ consists of a signature $\Sigma$ and **axioms** $Ax$ of the form $\Upsilon \vdash^{\bar{c}} t = u$. We write $\Upsilon \vdash^{\bar{c}}_{\mathtt{T}} t = u$ for derivability using the rules in Figure 1, where $(ax)$ rules range over axioms in $\mathtt{T}$. When $\Upsilon$ or $\bar{c}$ are empty, they are omitted. A judgement $\mathsf{И}\bar{c}.(\Upsilon \vdash_{\mathtt{T}} t = u)$ is understood as universally quantified over fresh atoms $\bar{c}$, that is, it represents all instances of $\Upsilon \vdash_{\mathtt{T}} t = u$ where $\bar{c}$ are fresh for the remainder of the judgement.

▶ **Example 3.2.2.**

1. For each signature $\Sigma$, there is a theory called $\mathtt{CORE}(\Sigma)$, which has no axioms, i.e., such that $Ax = \emptyset$. We usually omit $\Sigma$ and refer to it simply as $\mathtt{CORE}$.
2. Commutativity ($\mathtt{C}$) theory (of some $\mathtt{f}$) has a signature with a binary function symbol $\{\mathtt{f}:2\}$ and only one axiom $\vdash \mathtt{f}(X, Y) = \mathtt{f}(Y, X)$.
3. The theory $\mathtt{PI}$ captures (a fragment of) the algebraic structure induced by parallel composition and restriction in the $\pi$-calculus [19]. The signature $\Sigma_\pi$ is as in Example 2.1.4. We write $X|Y$ for $\mathtt{par}(X, Y)$ and $\nu a.X$ for $\nu([a]X)$.

$$\begin{array}{llcrcl}
(\mathit{comm}_{|}) & & \vdash & X|Y & = & Y|X \\
(\mathit{assoc}_{|}) & & \vdash & (X|Y)|Z & = & X|(Y|Z) \\
(\mathit{zero}_{|}) & & \vdash & X|\mathtt{0} & = & X \\
(\mathit{comm}_{\nu}) & & \vdash & \nu a.(\nu b.X) & = & \nu b.(\nu a.X) \\
(\mathit{zero}_{\nu}) & & \vdash & \nu a.\mathtt{0} & = & \mathtt{0} \\
(\mathit{extr}) & (a\ c) \curlywedge Y & \vdash^{c} & \nu a.(X|Y) & = & (\nu a.X)|Y
\end{array}$$

### Understanding the Key Rules in Figure 1

1. $(cong[ab])$ - $\alpha$-conversion: To prove $[a]t = [b]u$, a fresh atom $\boldsymbol{c'}$ is introduced, chosen outside the atoms occurring in the judgement, which is always possible since atoms are drawn from an infinite supply, and equality is checked under the renaming that maps both $a$ and $b$ to $\boldsymbol{c'}$. For example, deriving $[a]\mathtt{f}(a) = [b]\mathtt{f}(b)$ reduces to $\mathtt{f}(\boldsymbol{c'}) = \mathtt{f}(\boldsymbol{c'})$ after the renaming $(a\ \boldsymbol{c'})$ and $(b\ \boldsymbol{c'})$.
2. $(ax)$ - Axiom instantiation: Axioms are of the form $\Upsilon' \vdash^{\bar{\mathsf{c}}'} r = s$, where $\Upsilon'$ is a (normalised) fixed-point context and the judgement is globally quantified by $\bar{\mathsf{c}}'$. To apply such an axiom under a context $\Upsilon_{\bar{\mathsf{c}}}$, we first need to check that $\Upsilon_{\bar{\mathsf{c}}}$ satisfies the constraints of the axiom. This is expressed by the premise[11] $\Upsilon \vdash^{\bar{\mathsf{c}},\bar{\mathsf{c}}'} \Upsilon'^{\rho}\sigma$, which requires that all fixed-point constraints of the axiom hold after applying the permutation $\rho$ and substitution $\sigma$, i.e., $\Upsilon \vdash^{\bar{\mathsf{c}},\bar{\mathsf{c}}'} \pi^{\rho} \cdot X\sigma = X\sigma$ for every $\pi \curlywedge X \in \Upsilon'$. Intuitively, this means that an axiom can only be instantiated when the current context already enforces the invariances it requires. For example, if $\Upsilon'$ contains $(a\ b) \curlywedge X$, then we may instantiate the axiom at $X$ only if $\Upsilon_{\bar{\mathsf{c}}}$ entails $(a\ b) \cdot X\sigma = X\sigma$. When $\sigma$ instantiates $X$ to a suspended variable, this reduces to checking equality of suspended variables via $(var)$, as explained below.
3. $(var)$ – Variable equality: The rule $(var)$ handles equality between suspended variables. The context may impose fixed-point constraints on a variable $X$, and these constraints determine which permutations are allowed to act trivially on $X$. This information is collected in the group $\mathtt{P}(\Upsilon_{\bar{\mathsf{c}}}|_X)$. Thus, to prove $\pi \cdot X = \pi' \cdot X$, the rule checks whether the composition $\pi'^{-1} \circ \pi$ is already accounted for by the constraints on $X$. Intuitively, two suspensions of the same variable are equal whenever their difference is absorbed by the invariance imposed on that variable. To make the role of $(var)$ clearer, we compare it below with the traditional freshness-based rule.
4. $(fr)$ - Freshness introduction/elimination: This rule has a dual nature. Reading bottom-up, it introduces freshness constraints by adding $\pi \curlywedge X$ to the context. Reading top-down, it eliminates redundant constraints when their freshness information is implicit. Following [15], $(fr)$ is primitive to ensure an infinite supply of fresh atoms for abstraction.

### Comparative Analysis: Freshness vs. Fixed-Point

The variable rule highlights a key difference with freshness-based nominal equational systems [13,15] originating in nominal unification [26]. In such systems, the equality $\pi \cdot X = \pi' \cdot X$ is derivable only when every atom moved by the permutation $\pi'^{-1} \circ \pi$ is fresh for $X$:

$$\frac{\forall a \in \mathtt{mv}(\pi'^{-1} \circ \pi).\ a\#X \in \Delta}{\Delta \vdash \pi \cdot X = \pi' \cdot X}$$

### Expressivity Advantages

The fixed-point approach offers several expressivity benefits:

1. **Conservative extension**: If $\Upsilon$ is in normal form and $\Upsilon^{\curlywedge}|_X = \emptyset$, then $\Upsilon \vdash^{\bar{\mathsf{c}}} \pi \cdot X = \pi' \cdot X$ iff $\mathtt{mv}(\pi'^{-1} \circ \pi) \subseteq \mathtt{Atm}(\Upsilon^{\#}_{\bar{\mathsf{c}}}|_X)$. Indeed, in this case $\mathtt{P}(\Upsilon_{\bar{\mathsf{c}}}|_X) = \mathcal{G}(\mathsf{Fresh}(\Upsilon_{\bar{\mathsf{c}}}|_X))$, since $\mathsf{Sym}(\Upsilon|_X) = \{\mathtt{id}\}$, so $(var)$ applies exactly when the relative permutation $\pi'^{-1} \circ \pi$ moves only atoms made fresh for $X$ by the context, which is precisely the traditional freshness-based condition.

[11] Here $\bar{\mathsf{c}}, \bar{\mathsf{c}}'$ denotes the concatenation of $\bar{\mathsf{c}}$ and $\bar{\mathsf{c}}'$. Since axioms are interpreted as schemata closed under permutation of quantified atoms, we may assume without loss of generality that the quantified atoms $\bar{\mathsf{c}}$ and $\bar{\mathsf{c}}'$ are disjoint, by renaming fresh atoms if necessary.

2. **Enhanced reasoning**: It can derive equalities impossible in the freshness approach: For $\Psi_{\boldsymbol{c}} = \text{И}\boldsymbol{c}.\{(a\ b\ \boldsymbol{c}) \curlywedge X, (d\ e\ f) \curlywedge X, (f\ e) \curlywedge X\}$, we have $\mathtt{P}(\Psi_{\bar{\boldsymbol{c}}}|_X) = \mathcal{G}(\{a, b, \boldsymbol{c}\}) \circ \langle (d\ e\ f), (f\ e)\rangle$, and so
$$\frac{(d\ e) \circ (a\ \boldsymbol{c}) \in \mathtt{P}(\Psi_{\bar{\boldsymbol{c}}}|_X)}{\Psi \vdash^{\boldsymbol{c}} (a\ \boldsymbol{c}) \cdot X = (d\ e) \cdot X}$$
is derivable, since $(d\ e) = (d\ e\ f) \circ (f\ e)$, while the freshness approach would require $d, e\#X$.
3. **Unified treatment**: Combines symmetries induced by freshness, via fixed-points involving И-quantified atoms, with algebraic symmetries, captured by fixed-points that do not involve И-quantified atoms, within a single framework.

▶ **Example 3.2.3** (Unification in the $\pi$-calculus)**.** We consider the theory $\mathtt{PAR}$ consisting only of the axioms $(comm_|)$ and $(assoc_|)$ from Example 3.2.2(3). Consider the unification problem modulo $\alpha$-equivalence and $\mathtt{PAR}$:

$$\nu a.(\nu b.X) \ \approx ?\ \nu b.(\nu a.X).$$

In standard freshness-based nominal unification [26], solutions are pairs $\langle \Gamma, \sigma\rangle$, where $\Gamma$ is a freshness context and $\sigma$ a substitution. A straightforward solution enforces $a$ and $b$ to be fresh for $X$, yielding the context $\Gamma' = \{a\#X, b\#X\}$ with $\sigma = Id$. This solution is overly restrictive, since $\Gamma'$ excludes processes where $a$ and $b$ occur freely in a symmetric way, such as $\bar{a}b.\mathtt{0} \mid \bar{b}a.\mathtt{0}$. This limitation motivates the use of fixed-point constraints, which allow us to describe invariance under name permutations directly, without enforcing freshness. In fixed-point-based nominal unification, solutions would be pairs of the form $\langle \Upsilon, \sigma\rangle$, where $\Upsilon$ is a fixed-point context such that $\Upsilon \vdash_{\mathtt{PAR}} (\nu a.(\nu b.X))\sigma = (\nu b.(\nu a.X))\sigma$.

Using fixed-point constraints, we instead consider the context $\Upsilon' = \{(a\ b) \curlywedge X\}$, which expresses invariance of $X$ under swapping $a$ and $b$. We show that $\langle \Upsilon', Id\rangle$ is a solution. Since $\nu a.P$ is sugar for $\nu([a]P)$, it suffices to derive $\Upsilon' \vdash_{\mathtt{PAR}} [a]([b]X) = [b]([a]X)$. Applying $(cong[ab])$ twice, with fresh atoms $\boldsymbol{c}$ and $\boldsymbol{c'}$, this reduces to

$$\Upsilon' \vdash^{\boldsymbol{c},\boldsymbol{c'}}_{\mathtt{PAR}} (b\ \boldsymbol{c'})(a\ \boldsymbol{c}) \cdot X = (a\ \boldsymbol{c'})(b\ \boldsymbol{c}) \cdot X.$$

Let $\pi = (b\ \boldsymbol{c'})(a\ \boldsymbol{c}), \rho = (a\ \boldsymbol{c'})(b\ \boldsymbol{c})$. Then $\rho^{-1} \circ \pi = (a\ b)(\boldsymbol{c}\ \boldsymbol{c'})$, and since

$$\mathtt{P}(\Upsilon'_{\boldsymbol{c},\boldsymbol{c'}}|_X) = \mathcal{G}(\{\boldsymbol{c}, \boldsymbol{c'}\}) \circ \langle (a\ b)\rangle\,,$$

we have $\rho^{-1} \circ \pi \in \mathtt{P}(\Upsilon'_{\boldsymbol{c},\boldsymbol{c'}}|_X)$. Rule $(var)$ applies, yielding $\Upsilon' \vdash^{\boldsymbol{c},\boldsymbol{c'}}_{\mathtt{PAR}} \pi \cdot X = \rho \cdot X$. Retracing the abstraction steps, we conclude $\Upsilon' \vdash_{\mathtt{PAR}} (\nu a.(\nu b.X))Id = (\nu b.(\nu a.X))Id$.

Thus $\langle \Upsilon', Id\rangle$ represents a family of solutions. For instance, let $s_0 \equiv \bar{a}b.\mathtt{0} \mid \bar{b}a.\mathtt{0}$ and $\sigma_0 = [X \mapsto s_0]$. To show that $\sigma_0$ satisfies $\Upsilon'$, we must check that the constraint $(a\ b) \curlywedge X$ holds after instantiation, that is, $\vdash_{\mathtt{PAR}} (a\ b) \cdot s_0 = s_0$. This follows using the axiom $(comm_|)$, which states $\vdash Z \mid Y = Y \mid Z$. Instantiating this axiom with $\rho = \mathtt{id}$ and $\sigma = [Z \mapsto \bar{b}a.\mathtt{0},\ Y \mapsto \bar{a}b.\mathtt{0}]$, we obtain $\vdash_{\mathtt{PAR}} \bar{b}a.\mathtt{0} \mid \bar{a}b.\mathtt{0} = \bar{a}b.\mathtt{0} \mid \bar{b}a.\mathtt{0}$. Since $\bar{b}a.\mathtt{0} \mid \bar{a}b.\mathtt{0} \equiv (a\ b) \cdot (\bar{a}b.\mathtt{0} \mid \bar{b}a.\mathtt{0})$, this yields $\vdash_{\mathtt{PAR}} (a\ b) \cdot s_0 = s_0$. Therefore, $\sigma_0$ satisfies $\Upsilon'$, and $\langle \emptyset, \sigma_0\rangle$ is a concrete solution, which is an instance of the symbolic solution $\langle \Upsilon', Id\rangle$.

▶ **Example 3.2.4.** Consider a concurrency course in which students are asked to write a $\pi$-calculus parallel composition process satisfying a certain specification. If there are no free names, the marker can check whether two submitted solutions $p_1$ and $p_2$ are the same by adding the axiom

$$(plag) \qquad \vdash \mathtt{plagiarism}(P, P) = \mathtt{true}$$

to the theory $\mathtt{PI}$ and checking whether $\vdash \mathtt{plagiarism}(p_1, p_2) = \mathtt{true}$ is derivable. However, if there are free channels and the students are allowed to choose any names for them, submitted solutions should be compared using the theory $\mathtt{T}$ defined as the theory $\mathtt{PI}$ augmented with the following axioms[12]:

$$\begin{array}{rll} (\mathit{glob\text{-}sym}) & & \vdash (a\ b) \cdot (P|Q) = (P|Q), \\ (\mathit{plag}) & (a\ b) \curlywedge P & \vdash \mathtt{plagiarism}(P, P) = \mathtt{true} \end{array}$$

Given two answers $p_1$ and $p_2$, the marker checks whether $\vdash_{\mathtt{T}} \mathtt{plagiarism}(p_1, p_2) = \mathtt{true}$ is derivable. If so, then $p_1$ and $p_2$ are equal in the theory $\mathtt{T}$ and therefore coincide modulo a permutation of free channel names.

For instance, let $p_1 \equiv \overline{a}d.\mathtt{0} \mid \overline{b}e.\mathtt{0}$ and $p_2 \equiv \overline{b}d.\mathtt{0} \mid \overline{a}e.\mathtt{0}$. These two processes are not identified in the theory $\mathtt{PI}$. However, we have $p_2 \equiv (a\ b) \cdot p_1$ syntactically. To apply $(\mathit{plag})$, we must check that its premise $(a\ b) \curlywedge P$ holds for the chosen instance $P \mapsto p_1$, that is, that $\vdash_{\mathtt{T}} (a\ b) \cdot p_1 = p_1$. This follows by an instance of the axiom $(\mathit{glob\text{-}sym})$, instantiated with $\rho = \mathtt{id}$ and $\sigma = [P \mapsto \overline{a}d.\mathtt{0},\ Q \mapsto \overline{b}e.\mathtt{0}]$, yielding $\vdash_{\mathtt{T}} (a\ b) \cdot p_1 = p_1$. Hence, the premise of $(\mathit{plag})$ is satisfied, and we obtain $\vdash_{\mathtt{T}} \mathtt{plagiarism}(p_1, p_1) = \mathtt{true}$. Since $p_2 \equiv (a\ b) \cdot p_1$ and $\vdash_{\mathtt{T}} (a\ b) \cdot p_1 = p_1$, it follows that $\vdash_{\mathtt{T}} p_1 = p_2$, and by congruence $(\mathit{cong}\mathtt{f})$ we conclude $\vdash_{\mathtt{T}} \mathtt{plagiarism}(p_1, p_2) = \mathtt{true}$.

In a freshness-based setting [15], this problem cannot be expressed directly, since enforcing freshness of all free channel names would exclude processes that depend on them in a symmetric way.

## 4 Semantics

The semantic constructions, concepts, and proof techniques in the next sections closely follow those of nominal universal algebra as developed by Gabbay and Mathijssen [13, 15]. The purpose of this section is to show that this nominal-set semantics also extends to permutation fixed-point constraints, providing an adequate semantic foundation for our framework. In particular, we show that the resulting semantics supports soundness and completeness of the proof system.

### 4.1 Interpretations, models and validity

For a signature $\Sigma$, a **(nominal) $\Sigma$-algebra** $\mathfrak{A}$ consists of:

1. A nominal set $(\mathscr{A}, \cdot)$;
2. An equivariant[13] map $\mathtt{atom}^{\mathfrak{A}} : \mathbb{A} \to \mathscr{A}$ to interpret atoms;
3. An equivariant map $\mathtt{abs}^{\mathfrak{A}} : \mathbb{A} \times \mathscr{A} \to \mathscr{A}$ such that $a \notin \mathtt{supp}(\mathtt{abs}^{\mathfrak{A}}(a, x))$, for all $a \in \mathbb{A}$ and $x \in \mathscr{A}$; we use this to interpret abstraction;
4. An equivariant map $\mathtt{f}^{\mathfrak{A}} : \mathscr{A}^n \to \mathscr{A}$, for each $\mathtt{f}{:}n$ in $\Sigma$.

▶ **Remark 4.1.1.** $\mathtt{atom}^{\mathfrak{A}}$ need not be injective, ensuring singleton interpretations model all theories (cf. Remark 4.22 in [15]).

[12] Axioms are understood as equivariant schemes. In particular, an axiom written with specific atoms, such as $(\mathit{glob\text{-}sym})$, represents all its variants obtained by renaming the atoms at the meta-level. Hence it suffices to consider invariance under a single swapping as the invariance under arbitrary finite permutations follows by equivariance at the object-level.

[13] Given $\mathcal{G}(\mathbb{A})$-sets $\mathscr{X}, \mathscr{Y}$, call a map $f : \mathscr{X} \to \mathscr{Y}$ **equivariant** when $\pi \cdot f(x) = f(\pi \cdot x)$ for all $\pi \in \mathcal{G}(\mathbb{A})$ and $x \in \mathscr{X}$.

Given a $\Sigma$-algebra $\mathfrak{A}$ and a **valuation** $\varsigma : \mathbb{V} \to \mathscr{A}$, the **interpretation** $[\![t]\!]^{\mathfrak{A}}_{\varsigma}$ is defined inductively by:

$$\begin{array}{rcl@{\qquad}rcl} [\![a]\!]^{\mathfrak{A}}_{\varsigma} & = & \mathtt{atom}^{\mathfrak{A}}(a) & [\![\pi \cdot X]\!]^{\mathfrak{A}}_{\varsigma} & = & \pi \cdot \varsigma(X) \\ [\![[a]t]\!]^{\mathfrak{A}}_{\varsigma} & = & \mathtt{abs}^{\mathfrak{A}}(a, [\![t]\!]^{\mathfrak{A}}_{\varsigma}) & [\![\mathtt{f}(t_1,\ldots,t_n)]\!]^{\mathfrak{A}}_{\varsigma} & = & \mathtt{f}^{\mathfrak{A}}([\![t_1]\!]^{\mathfrak{A}}_{\varsigma},\ldots,[\![t_n]\!]^{\mathfrak{A}}_{\varsigma}) \end{array}$$

The interpretation map $[\![\cdot]\!]^{\mathfrak{A}}_{\varsigma}$ is equivariant: $\pi \cdot [\![t]\!]^{\mathfrak{A}}_{\varsigma} = [\![\pi \cdot t]\!]^{\mathfrak{A}}_{\varsigma}$ for any permutation $\pi$ and term $t$.

▶ **Definition 4.1.2.** For $\Sigma$-algebra $\mathfrak{A}$ and theory $\mathtt{T} = (\Sigma, Ax)$:
1. $[\![\Upsilon]\!]^{\mathfrak{A}}_{\varsigma}$ is **valid** if $\pi \cdot \varsigma(X) = \varsigma(X)$ for each $\pi \curlywedge X \in \Upsilon$.
2. $[\![\Upsilon \vdash^{\bar{c}} t = u]\!]^{\mathfrak{A}}_{\varsigma}$ is **valid** if $\text{И}\bar{c}.([\![\Upsilon]\!]^{\mathfrak{A}}_{\varsigma}$ is valid $\Rightarrow [\![t]\!]^{\mathfrak{A}}_{\varsigma} = [\![u]\!]^{\mathfrak{A}}_{\varsigma})$.
3. $\mathfrak{A}$ is a **model** of $\mathtt{T}$ if all axioms in $Ax$ are valid in $\mathfrak{A}$ for any valuation.
4. $\Upsilon \vDash^{\bar{c}}_{\mathtt{T}} t = u$ if $[\![\Upsilon \vdash^{\bar{c}} t = u]\!]^{\mathfrak{A}}_{\varsigma}$ valid in all models of $\mathtt{T}$ for any valuation.

Derivability of equality judgements is sound for the nominal set semantics.

▶ **Theorem 4.1.3** (Soundness). *If* $\Upsilon \vdash^{\bar{c}}_{\mathtt{T}} t = u$ *then* $\Upsilon \vDash^{\bar{c}}_{\mathtt{T}} t = u$.

**Proof sketch.** The argument follows the same pattern as in nominal algebra [15], by induction on the derivation; the only difference is the interpretation of permutation fixed-point constraints, which is immediate via the permutation action. We illustrate the case of $(var)$.

Here $\Upsilon \vdash^{\bar{c}} \pi \cdot X = \rho \cdot X$ with $\rho^{-1} \circ \pi \in \mathtt{P}(\Upsilon_{\bar{c}}|_X)$. Let $\mathfrak{A}$ be a model and $\varsigma$ a valuation. Assume that for a given choice of fresh $\bar{c}$ we have $\pi' \cdot \varsigma(X) = \varsigma(X)$ for all $\pi' \curlywedge X \in \Upsilon$. In particular, for every $\eta \curlywedge X \in \Upsilon|_X$ we have $\eta \cdot \varsigma(X) = \varsigma(X)$. By the characterisation of generalised fixed-points (Theorem 2.3.5), this implies that (i) $\mathtt{Atm}(\Upsilon^{\#}_{\bar{c}}|_X)\#\varsigma(X)$, and (ii) every permutation in $\mathtt{perm}(\Upsilon^{\curlywedge}|_X)$ fixes $\varsigma(X)$.

Since $\rho^{-1} \circ \pi \in \mathtt{P}(\Upsilon_{\bar{c}}|_X)$, we can write $\rho^{-1} \circ \pi = \pi_1 \circ \pi_2$ with $\pi_1 \in \mathcal{G}(\mathsf{Fresh}(\Upsilon_{\bar{c}}|_X))$ and $\pi_2 \in \langle \mathtt{perm}(\Upsilon^{\curlywedge}|_X) \rangle$. By (i), $\pi_1$ fixes $\varsigma(X)$ following the definition of support, and by (ii) plus closure of generated groups (Lemma 2.3.4(2)), $\pi_2$ also fixes $\varsigma(X)$. Hence $(\rho^{-1} \circ \pi) \cdot \varsigma(X) = \varsigma(X)$, and applying $\rho$ yields $\pi \cdot \varsigma(X) = \rho \cdot \varsigma(X)$, as required. ◀

## 4.2 Free term models

We construct a canonical model for nominal theories $\mathtt{T}$, extending the classical free term model construction. Since traditional constants generate only elements with empty support, we enrich the signature with auxiliary term-formers.

Let $\Sigma$ be a signature and $\mathcal{D}$ a disjoint set of **auxiliary fresh term-formers**. The set of **free terms** $\mathsf{Free}(\Sigma, \mathcal{D})$ is defined by:

$$g ::= a \mid [a]g \mid \mathtt{f}(g_1, \ldots, g_n) \mid \mathtt{d}(a_1, \ldots, a_k)$$

Here $\mathtt{f}{:}n$ ranges over $\Sigma$ and $\mathtt{d}{:}k$ range over $\mathcal{D}$.

As expected, to obtain nominal models for a theory $\mathtt{T}$, we work with the set of free terms quotiented by provable equality modulo $\mathtt{T}$. This construction incorporates $\alpha$-equivalence modulo $\mathtt{T}$, allowing us to build a model that satisfies the intended properties in the nominal set semantics.

▶ **Definition 4.2.1** (Free term equivalence modulo $\mathtt{T}$). Free terms $g$ and $g'$ are equivalent if $\vdash^{\bar{c}}_{\mathtt{T}} g = g'$ (for some $\bar{c}$) is derivable without using $(cong\mathtt{d})$ for any $\mathtt{d} \in \mathcal{D}$. Let $\mathtt{cl}_{\mathtt{T}}(g)$ denote the equivalence class and $\mathsf{Free}(\mathtt{T}, \mathcal{D})$ the quotient set.

$\mathsf{Free}(\mathtt{T},\mathcal{D})$ forms a nominal set under $\pi\cdot\mathtt{cl}_\mathtt{T}(g)=\mathtt{cl}_\mathtt{T}(\pi\cdot g)$ with support $\mathtt{supp}(\mathtt{cl}_\mathtt{T}(g))=\bigcap_{g'\in\mathtt{cl}_\mathtt{T}(g)}\mathtt{supp}(g')$, where $\mathtt{supp}(g')=\mathtt{Atm}(g')$.

▶ **Remark 4.2.2.** Following [15], we exclude the $(cong\mathtt{d})$ rule to avoid situations where axioms like $\vdash a=b$ would cause $\mathtt{cl}_\mathtt{T}(\mathtt{d}(a_1,\dots,a_n))$ to have empty support. For example, if the theory proves $\vdash a=b$, then by equivariance it may identify arbitrary atoms. If congruence for $\mathtt{d}$ were allowed, we could derive equalities such as $\mathtt{d}(a,b)=\mathtt{d}(e,f)$ for arbitrary atoms $e,f$. Consequently, the class $\mathtt{cl}_\mathtt{T}(\mathtt{d}(a,b))$ would have empty support, defeating the intended role of $\mathtt{d}(a,b)$ as a representative whose support records the atoms $a,b$.

We define the **free nominal $\Sigma$-algebra** as follows. Its carrier is the nominal set $\mathsf{Free}(\mathtt{T},\mathcal{D})$, and the algebraic operations are given by:

1. $\mathtt{atom}^{\mathsf{Free}(\mathtt{T},\mathcal{D})}(a)=\mathtt{cl}_\mathtt{T}(a)$, for all $a\in\mathbb{A}$;
2. $\mathtt{abs}^{\mathsf{Free}(\mathtt{T},\mathcal{D})}(a,\mathtt{cl}_\mathtt{T}(g))=\mathtt{cl}_\mathtt{T}([a]g)$ for all $a\in\mathbb{A}$ and all $g\in\mathsf{Free}(\Sigma,\mathcal{D})$;
3. $\mathtt{f}^{\mathsf{Free}(\mathtt{T},\mathcal{D})}(\mathtt{cl}_\mathtt{T}(g_1)\dots,\mathtt{cl}_\mathtt{T}(g_n))=\mathtt{cl}_\mathtt{T}(\mathtt{f}(g_1,\dots,g_n))$, for all $g_1,\dots,g_n\in\mathsf{Free}(\Sigma,\mathcal{D})$ and all $\mathtt{f}:n$ in $\Sigma$.

▶ **Theorem 4.2.3.** *The algebra* $\mathsf{Free}(\mathtt{T},\mathcal{D})$ *forms a model of* $\mathtt{T}$

The proof follows the same argument as Theorem 6.13 in [13].

## 4.3 Completeness

Fix a signature $\Sigma$, a theory $\mathtt{T}=(\Sigma,Ax)$, and terms $t,u$ together with a context $\Upsilon_{\bar{c}}$ in signature $\Sigma$.

We prove completeness: semantic equality implies derivability.

▶ **Theorem 4.3.1** (Completeness). *If* $\Upsilon\vDash^{\bar{c}}_{\mathtt{T}}t=u$ *then* $\Upsilon\vdash^{\bar{c}}_{\mathtt{T}}t=u$.

The proof constructs a free term model $\mathsf{Free}(\mathtt{T},\mathcal{D})$ with a canonical valuation that "realizes" unknowns as concrete terms while preserving fixed-point constraints in $\Upsilon_{\bar{c}}$.

▶ **Definition 4.3.2.** Let $\mathcal{A}=\mathtt{Atm}(\Upsilon,t,u)\setminus\{\bar{c}\}$ and $\mathcal{X}=\mathtt{Var}(\Upsilon,t,u)$.

1. For each $X\in\mathcal{X}$, choose the following data:
   a. Define $\mathcal{A}_X$ as the set of atoms $a\in\mathcal{A}$ such that there is no $\pi\curlywedge X\in\Upsilon^{\#}|_X$ with $a\in\mathtt{mv}(\pi)$;
   b. Let $a_{X_1},\dots,a_{X_{k_X}}$ be the atoms in $\mathcal{A}_X$, listed in an arbitrary but fixed order.
   c. Introduce a fresh term-former $\mathtt{d}_X:k_X$.
2. For each $X\notin\mathcal{X}$, let $\mathtt{d}_X:0$ be a term-former. Denote by $\mathcal{D}$ the set of all such $\mathtt{d}$'s (so that for every $X\in\mathbb{V}$ we have a corresponding $\mathtt{d}_X\in\mathcal{D}$). Moreover, let $\Sigma^+=\Sigma\cup\mathcal{D}$.

▶ **Example 4.3.3.** For $\Upsilon_{c}=\text{И}c.\{(a\ b\ c)\curlywedge X,(d\ e)\curlywedge X\}$ and $t,u$ mentioning $a',b'$. For $X$, we fix the order $d,e,a',b'$ and assign a term-former $\mathtt{d}_X:4$. Thus, in the free term model, $X$ will be realized as $\mathtt{d}_X(d,e,a',b')$.

The **canonical substitution** $\sigma$ embeds unknowns as concrete terms in the free model as follows:

$$X\sigma\equiv\begin{cases}\mathtt{d}_X(a_{X_1},\dots,a_{X_{k_X}}) & X\in\mathcal{X},\\ \mathtt{d}_X() & X\notin\mathcal{X}.\end{cases}$$

In the standard nominal framework [15], a term $\mathtt{d}(a_1,\dots,a_n)$ represents an unknown element with support $\{a_1,\dots,a_n\}$, which suffices for freshness-based theories. With fixed-point constraints, however, terms must also reflect permutation invariance. For instance, if

$(a\ b) \curlywedge X \in \Upsilon$, we would like a term $\mathtt{d}(\dots)$ such that $(a\ b)\cdot g =_{\mathtt{T}} g$. The standard construction fails, since $\mathtt{d}$-term formers alone do not encode the algebraic symmetries required by the context.

▶ **Remark 4.3.4.** The canonical free term model $\mathsf{Free}(\mathtt{T}, \mathcal{D})$ with valuation $\varsigma^*(X) = \mathtt{cl}_{\mathtt{T}}(X\sigma)$ does not, in general, satisfy pure fixed-point constraints $\pi \curlywedge X$. Consider Example 4.3.3. Freshness-related constraints such as $(a\ b\ \boldsymbol{c}) \curlywedge X$ are handled by construction. However, for a pure fixed-point constraint $(d\ e) \curlywedge X$, there is no guarantee that $\mathtt{T}$ proves $\mathtt{d}_X(e, d, a', b') =_{\mathtt{T}} \mathtt{d}_X(d, e, a', b')$. Hence, $(d\ e)$ may fail to fix $\mathtt{cl}_{\mathtt{T}}(\mathtt{d}_X(d, e, a', b'))$ in $\mathsf{Free}(\mathtt{T}, \mathcal{D})$.

This motivates the following extension of the theory.

▶ **Definition 4.3.5.** Let $\mathtt{T}^+$ be the theory obtained from $\mathtt{T}$ by adding, for each $\pi \curlywedge X \in \Upsilon^{\curlywedge}|_X$, the axiom $\vdash \pi \cdot \mathtt{d}_X(a_{X_1}, \dots, a_{X_{k_X}}) = \mathtt{d}_X(a_{X_1}, \dots, a_{X_{k_X}})$.

▶ **Remark 4.3.6.** The extension $\mathtt{T}^+$ equips the $\mathtt{d}$-symbols with the algebraic symmetries required by the context. In $\mathsf{Free}(\mathtt{T}^+, \mathcal{D})$ with $\varsigma^*(X) = \mathtt{cl}_{\mathtt{T}^+}(X\sigma)$, all fixed-point constraints in $\Upsilon$ are satisfied. For Example 4.3.3, $\mathtt{T}^+$ contains the axiom $\vdash \mathtt{d}_X(e, d, a', b') = \mathtt{d}_X(d, e, a', b')$, ensuring that $(d\ e)$ fixes $\mathtt{cl}_{\mathtt{T}^+}(\mathtt{d}_X(d, e, a', b'))$.

▶ **Lemma 4.3.7.** *The following hold:*
1. $[\![t]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*} = \mathtt{cl}_{\mathtt{T}^+}(t\sigma)$ *and* $[\![u]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*} = \mathtt{cl}_{\mathtt{T}^+}(u\sigma)$.
2. $\text{И}\bar{\mathsf{c}}.(\forall \pi \curlywedge X \in \Upsilon.\ \pi \cdot [\![X]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*} = [\![X]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*})$.

**Proof sketch.** Item 1 is immediate by induction on the structure of the terms. For item 2, fix a fresh choice of $\bar{\mathsf{c}}$ and let $\pi \curlywedge X \in \Upsilon$. If $\pi \curlywedge X \in \Upsilon^{\#}$, then $\pi$ moves only atoms outside $\mathtt{supp}([\![X]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*})$, by construction; hence it fixes $[\![X]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*}$. If $\pi \curlywedge X \in \Upsilon^{\curlywedge}$, then $\pi$ fixes $[\![X]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*}$ by construction of $\mathtt{T}^+$. Thus, for this fresh choice of $\bar{\mathsf{c}}$, every constraint in $\Upsilon$ is satisfied, i.e. $\forall \pi \curlywedge X \in \Upsilon.\ \pi \cdot [\![X]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*} = [\![X]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*}$. Hence $\text{И}\bar{\mathsf{c}}.(\forall \pi \curlywedge X \in \Upsilon.\ \pi \cdot [\![X]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*} = [\![X]\!]^{\mathsf{Free}(\mathtt{T}^+,\mathcal{D})}_{\varsigma^*})$. ◀

We now invert the embedding $\sigma$: from concrete terms $\mathtt{d}_X(a_{X_1}, \dots, a_{X_{k_X}})$ we reconstruct the original unknown $X$.

▶ **Definition 4.3.8.** Let $\Pi$ be a derivation of $\vdash^{\bar{\mathsf{c}}}_{\mathtt{T}^+} t\sigma = u\sigma$ (without using $(cong\mathtt{d})$), and extend the set of atoms $\mathcal{A}$ to a larger set $\mathcal{A}^+$ by including:
1. a set $\mathcal{C}$ containing any atoms appearing in $\Pi$ that are not already in $\mathcal{A}$,
2. a set $\mathcal{B}$ of fresh atoms, in bijection with $\mathcal{A}$ – for convenience, we write $b_{X_i}$ for the atom corresponding under that bijection with $a_{X_i}$ – and
3. one additional fresh atom $e^*$.

We define an **inverse mapping** $(-)^{-1}$ from free terms in $\Sigma^+$ mentioning *only* atoms from $\mathcal{A}^+ \setminus (\mathcal{B} \cup \{e^*\})$ to nominal terms in $\Sigma$ inductively:

$$\begin{array}{rcl} a^{-1} & \equiv & a, \\ ([a]g)^{-1} & \equiv & [a]g^{-1}, \\ \mathtt{f}(g_1, \dots, g_n)^{-1} & \equiv & \mathtt{f}(g_1^{-1}, \dots, g_n^{-1}), \quad \mathtt{f} \in \Sigma, \\ \mathtt{d}_X()^{-1} & \equiv & e^*, \quad X \notin \mathcal{X}, \\ \mathtt{d}_X(a'_{X_1}, \dots, a'_{X_{k_X}})^{-1} & \equiv & \pi_X(a'_{X_1}, \dots, a'_{X_{k_X}}) \cdot X, \quad X \in \mathcal{X}, \end{array}$$

where $\pi_X(a'_{X_1}, \dots, a'_{X_{k_X}}) = (a'_{X_1}\ b_{X_1}) \circ \dots \circ (a'_{X_{k_X}}\ b_{X_{k_X}}) \circ (b_{X_1}\ a_{X_1}) \circ \dots \circ (b_{X_{k_X}}\ a_{X_{k_X}})$.

The permutation $\pi_X$ uses the intermediate atoms $b_{X_i}$ as fresh markers to restore the original atom ordering. We restrict to ground terms $g$ with $\mathtt{Atm}(g) \subseteq \mathcal{A}^+ \setminus (\mathcal{B} \cup \{e^*\})$.

▶ **Example 4.3.9.** For $\mathcal{A}_X = \{a, b, d\}$ with order $a, b, d$:
1. $\mathtt{d}_X(a, b, d)^{-1} = X$
2. $\mathtt{d}_X(b, d, a)^{-1} = (b\ b_{X_1}) \circ (d\ b_{X_2}) \circ (a\ b_{X_3}) \circ (b_{X_1}\ a) \circ (b_{X_2}\ b) \circ (b_{X_3}\ d) \cdot X$

Observe that applying $\sigma$ to this suspension recovers $\mathtt{d}_X(b, d, a)$.

▶ **Definition 4.3.10.** Define $\Upsilon^+_{\bar{c}'}$ as $\Upsilon_{\bar{c}}$ extended with:
1. $\bar{c}' = \bar{c}, c^*$ where $c^*$ is a fresh name
2. $(b_1\ \ldots\ b_n\ c_1^*\ \ldots\ c_n^*\ e^*\ c^*) \curlywedge X$ for all $X \in \mathcal{X}$ where $\mathcal{B} = \{b_1, \ldots, b_n\}$ and $\mathcal{C} = \{c_1^*\ \ldots\ c_n^*\}$.

This extension ensures the technical atoms $b_i$, $e^*$, and $c_i^*$ are treated as fresh for each unknown $X$, preventing interference with the original term structure during inversion.

The inversion mapping respects permutation structure:

▶ **Lemma 4.3.11.** $\Upsilon^+ \vdash^{\bar{c}'}_{\mathtt{CORE}} (\pi \cdot g)^{-1} = \pi \cdot g^{-1}$ *for all* $\pi$ *such that* $\mathtt{mv}(\pi) \subseteq \mathcal{A}^+ \setminus (\mathcal{B} \cup \{e^*\})$.

**Proof sketch.** By induction on the structure of $g$. The only non-trivial case is $g \equiv \mathtt{d}_X(\overline{a'_X})$, with $\overline{a'_X} = a'_{X_1}, \ldots, a'_{X_{k_X}}$. We must show $\Upsilon^+ \vdash^{\bar{c}'}_{\mathtt{CORE}} (\pi \cdot \mathtt{d}_X(\overline{a'_X}))^{-1} = \pi \cdot \mathtt{d}_X(\overline{a'_X})^{-1}$.

Unfolding the definitions, we have $(\pi \cdot \mathtt{d}_X(\overline{a'_X}))^{-1} \equiv \pi_X(\overline{\pi(a'_X)}) \cdot X$ and $\pi \cdot \mathtt{d}_X(\overline{a'_X})^{-1} \equiv (\pi \circ \pi_X(\overline{a'_X})) \cdot X$. Thus, it suffices to prove $\Upsilon^+ \vdash^{\bar{c}'}_{\mathtt{CORE}} \pi_X(\overline{\pi(a'_X)}) \cdot X = (\pi \circ \pi_X(\overline{a'_X})) \cdot X$.

By rule $(var)$, this holds if $(\pi \circ \pi_X(\overline{a'_X}))^{-1} \circ \pi_X(\overline{\pi(a'_X)}) \in \mathtt{P}(\Upsilon^+_{\bar{c}'}|_X)$. A direct calculation yields $(\pi \circ \pi_X(\overline{a'_X}))^{-1} \circ \pi_X(\overline{\pi(a'_X)}) = (\pi^{-1})^{(\mathcal{A}_X\ \mathcal{B}_X)}$ where $(\mathcal{A}_X\ \mathcal{B}_X) = (a_{X_1}\ b_{X_1}) \circ \cdots \circ (a_{X_{k_X}}\ b_{X_{k_X}})$. Its support is contained in $\mathtt{Atm}(\Upsilon^{\#}_{\bar{c}}|_X) \cup \mathcal{B}$, which is included in $\mathtt{Atm}((\Upsilon^+)^{\#}_{\bar{c}'}|_X)$. Hence $(\pi^{-1})^{(\mathcal{A}_X\ \mathcal{B}_X)} \in \mathcal{G}(\mathsf{Fresh}(\Upsilon^+_{\bar{c}'}|_X)) \subseteq \mathtt{P}(\Upsilon^+_{\bar{c}'}|_X)$, and the result follows. ◀

▶ **Lemma 4.3.12.** *The following hold:*
1. $\Upsilon^+ \vdash^{\bar{c}'}_{\mathtt{CORE}} (t\sigma)^{-1} = t$ *and* $\Upsilon^+ \vdash^{\bar{c}'}_{\mathtt{CORE}} (u\sigma)^{-1} = u$.
2. *If* $\vdash^{\bar{c}}_{\mathtt{T}^+} t\sigma = u\sigma$, *then* $\Upsilon \vdash^{\bar{c}}_{\mathtt{T}} t = u$.

**Proof sketch.** For part 1, the result follows by a straightforward induction on the structure of the terms. For part 2, the result follows by induction on the given derivation $\vdash^{\bar{c}}_{\mathtt{T}^+} t\sigma = u\sigma$. ◀

**Proof of Theorem 4.3.1 (Completeness).** Assume $\Upsilon \vDash^{\bar{c}}_{\mathtt{T}} t = u$. We interpret terms in the free model $\mathsf{Free}(\mathtt{T}^+, \mathcal{D})$ under $\varsigma^*(X) = \mathtt{cl}_{\mathtt{T}^+}(X\sigma)$. Fix a fresh choice of $\bar{c}$. By Lemma 4.3.7(2), all constraints in $\Upsilon$ are satisfied under this choice. Hence, by the semantic assumption $\Upsilon \vDash^{\bar{c}}_{\mathtt{T}} t = u$, we obtain $[\![t]\!]^{\mathsf{Free}(\mathtt{T}^+, \mathcal{D})}_{\varsigma^*} = [\![u]\!]^{\mathsf{Free}(\mathtt{T}^+, \mathcal{D})}_{\varsigma^*}$. By Lemma 4.3.7(1), this gives $\mathtt{cl}_{\mathtt{T}^+}(t\sigma) = \mathtt{cl}_{\mathtt{T}^+}(u\sigma)$, hence $\vdash^{\bar{c}}_{\mathtt{T}^+} t\sigma = u\sigma$. By Lemma 4.3.12(2), this lifts to $\Upsilon \vdash^{\bar{c}}_{\mathtt{T}} t = u$. ◀

## 5 Conclusion and Future work

We have introduced a new framework for equational reasoning with binding and equational theories, extending freshness constraints to И-quantified permutation fixed-points. The resulting system is expressive and equipped with a sound and complete derivation system with respect to the nominal set semantics.

One natural next step is to investigate whether our setting admits algebraic closure properties similar to those for freshness in nominal algebra [13], where Birkhoff's HSP (Homomorphic images, Subalgebras, Products) theorem has been successfully adapted (an alternative approach can be found in [17]).

Another next step is to study connections to *nominal automata* [3], where orbit-finite sets and group-theoretic methods play a central role.

## References


1 Mauricio Ayala-Rincón, Maribel Fernández, and Daniele Nantes-Sobrinho. On Nominal Syntax and Permutation Fixed Points. *Log. Methods Comput. Sci.*, 16(1):19:1–19:36, 2020. doi:10.23638/LMCS-16(1:19)2020.

2 Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998. doi:10.1017/CBO9781139172752.

3 Mikolaj Bojanczyk, Bartek Klin, and Slawomir Lasota. Automata theory in nominal sets. *Log. Methods Comput. Sci.*, 10(3), 2014. doi:10.2168/LMCS-10(3:4)2014.

4 Stanley Burris and Hanamantagouda P. Sankappanavar. *A Course in Universal Algebra*, volume 78 of *Graduate Texts in Mathematics.* Springer, 1981. URL: https://link.springer.com/book/9781461381327.

5 Ali K. Caires-Santos, Maribel Fernández, and Daniele Nantes-Sobrinho. Equational Reasoning Modulo Commutativity in Languages with Binders. In *Automated Deduction – CADE 30: 30th International Conference on Automated Deduction, Stuttgart, Germany, July 28-31, 2025, Proceedings*, pages 632–652, Berlin, Heidelberg, 2025. Springer-Verlag. doi:10.1007/978-3-031-99984-0_33.

6 Ali K. Caires-Santos, Maribel Fernández, and Daniele Nantes-Sobrinho. Strong Nominal Semantics for Fixed-Point Constraints. *Electronic Notes in Theoretical Informatics and Computer Science, Proceedings of MFPS XL.*, 4, December 2024. doi:10.46298/entics.14777.

7 Arthur Charguéraud. The locally nameless representation. *J. Autom. Reason.*, 49(3):363–408, 2012. doi:10.1007/s10817-011-9225-2.

8 Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn L. Talcott, editors. *All About Maude - A High-Performance Logical Framework*, volume 4350 of *LNCS.* Springer, 2007. doi:10.1007/978-3-540-71999-1.

9 N.G de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. *Indagationes Mathematicae (Proceedings)*, 75(5):381–392, 1972. doi:10.1016/1385-7258(72)90034-0.

10 Francisco Durán, Santiago Escobar, Jose Meseguer, and Julia Sapiña. Nuitp: An inductive theorem prover for equational program verification. In *Proceedings of the 26th International Symposium on Principles and Practice of Declarative Programming*, PPDP '24, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3678232.3678236.

11 Amy P. Felty, Alberto Momigliano, and Brigitte Pientka. The next 700 challenge problems for reasoning with higher-order abstract syntax representations - part 2 - A survey. *J. Autom. Reason.*, 55(4):307–372, 2015. doi:10.1007/s10817-015-9327-3.

12 Maribel Fernández and Murdoch James Gabbay. Nominal Rewriting. *Inf. Comput.*, 205(6):917–965, 2007. doi:10.1016/j.ic.2006.12.002.

13 Murdoch James Gabbay. Nominal Algebra and the HSP Theorem. *J. Log. Comput.*, 19(2):341–367, 2009. doi:10.1093/logcom/exn055.

14 Murdoch James Gabbay. Foundations of Nominal Techniques: Logic and Semantics of Variables in Abstract Syntax. *Bull. Symb. Log.*, 17(2):161–229, 2011. doi:10.2178/bsl/1305810911.

15 Murdoch James Gabbay and Aad Mathijssen. Nominal (Universal) Algebra: Equational Logic with Names and Binding. *J. Log. Comput.*, 19(6):1455–1508, 2009. doi:10.1093/logcom/exp033.

16 Murdoch James Gabbay and Andrew M. Pitts. A New Approach to Abstract Syntax with Variable Binding. *Formal Aspects Comput.*, 13(3-5):341–363, 2002. doi:10.1007/s001650200016.

17 Alexander Kurz and Daniela Petrişan. On universal algebra over nominal sets. *Mathematical Structures in Computer Science*, 20(2):285–318, 2010. doi:10.1017/S0960129509990399.

18 José Meseguer. Inductive reasoning with equality predicates, contextual rewriting and variant-based simplification. *J. Log. Algebraic Methods Program.*, 144:101036, 2025. doi:10.1016/j.jlamp.2025.101036.

19 Robin Milner, Joachim Parrow, and David Walker. A Calculus of Mobile Processes, I. *Information and Computation*, 100(1):1–40, 1992. doi:10.1016/0890-5401(92)90008-4.

20 Peter W. O'Hearn, John C. Reynolds, and Hongseok Yang. Local reasoning about programs that alter data structures. In Laurent Fribourg, editor, *Computer Science Logic (CSL 2001)*, volume 2142 of *Lecture Notes in Computer Science*, pages 1–19, Berlin, Heidelberg, 2001. Springer. doi:10.1007/3-540-44802-0_1.

21 F. Pfenning and C. Elliott. Higher-Order Abstract Syntax. *SIGPLAN Not.*, 23(7):199–208, 1988. doi:10.1145/960116.54010.

22 Andrew M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2013. doi:10.1017/CBO9781139084673.

23 Andrei Popescu. Nominal recursors as epi-recursors. *Proc. ACM Program. Lang.*, 8(POPL):425–456, 2024. doi:10.1145/3632857.

24 John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS 2002)*, pages 55–74, Washington, DC, USA, 2002. IEEE Computer Society. doi:10.1109/LICS.2002.1029817.

25 Joseph J. Rotman. *An Introduction to the Theory of Groups*, volume 148 of *Graduate Texts in Mathematics*. Springer, New York, 4 edition, 1995. URL: https://link.springer.com/book/10.1007/978-1-4612-4176-8.

26 Christian Urban, Andrew M. Pitts, and Murdoch James Gabbay. Nominal Unification. *Theor. Comput. Sci.*, 323(1-3):473–497, 2004. doi:10.1016/j.tcs.2004.06.016.

27 Jan van Brügge, James McKinna, Andrei Popescu, and Dmitriy Traytel. Barendregt convenes with Knaster and Tarski: Strong rule induction for syntax with bindings. *Proc. ACM Program. Lang.*, 9(POPL):1687–1718, 2025. doi:10.1145/3704893.


## A Proofs of Section 2.3

▶ **Lemma 2.3.4.** *Let $x \in \mathscr{X}$ and $\pi, \eta_1, \dots, \eta_n$ be permutations written as compositions of disjoint cycles.*

1. *If $\pi \cdot x = x$, $\eta$ is a cycle of $\pi$, and $\mathtt{mv}(\eta) - \mathtt{supp}(x) \neq \emptyset$, then $\mathtt{mv}(\eta) \# \mathtt{supp}(x)$.*
2. *If $\eta_i \cdot x = x$, for all $i = 1, \dots, n$, and $\rho \in \langle \eta_1, \dots, \eta_n \rangle$, then $\rho \cdot x = x$.*

**Proof.**

1. Write $\pi = \mu_1 \circ \cdots \circ \mu_n$ with disjoint cycles $\mu_1, \dots, \mu_n$, and let $\eta = \mu_j$ for some $1 \leq j \leq n$. Choose $a \in \mathtt{mv}(\eta) \setminus \mathtt{supp}(x)$.
   Suppose, by contradiction, there exists $b \in \mathtt{mv}(\eta) \cap \mathtt{supp}(x)$. Since the cycles are disjoint, $\mu_i(b) = b$ for all $i \neq j$. Hence $\pi(b) = \eta(b)$. Since $b \in \mathtt{supp}(x)$ and $\pi \cdot x = x$, it follows that $\pi(b) \in \mathtt{supp}(\pi \cdot x) = \mathtt{supp}(x)$, and therefore $\eta(b) \in \mathtt{supp}(x)$.
   Because $a, b \in \mathtt{mv}(\eta)$, there exists $l > 0$ such that $\eta^l(b) = a$. Applying $\pi$ repeatedly to $\eta(b) \in \mathtt{supp}(x)$ and using that $\pi \cdot x = x$ yields $\eta^l(b) \in \mathtt{supp}(x)$, hence $a \in \mathtt{supp}(x)$, contradicting the choice of $a$.
2. Assume that $\eta_i \cdot x = x$ for every $i = 1, \dots, n$, and let $\rho \in \langle \eta_1, \dots, \eta_n \rangle$. By definition of the subgroup generated by $\eta_1, \dots, \eta_n$, the permutation $\rho$ can be written as a finite composition $\rho = \zeta_1 \circ \cdots \circ \zeta_k$, where each $\zeta_r$ belongs to $\{\eta_1^{\pm 1}, \dots, \eta_n^{\pm 1}\}$.
   We first note that each inverse $\eta_i^{-1}$ also fixes $x$. Indeed, from $\eta_i \cdot x = x$, applying $\eta_i^{-1}$ to both sides gives $\eta_i^{-1} \cdot (\eta_i \cdot x) = \eta_i^{-1} \cdot x$, and therefore $x = \eta_i^{-1} \cdot x$. So every $\zeta_r$ in the above decomposition fixes $x$. We now prove that $\rho$ fixes $x$ by repeated application of these equalities:

$$\rho \cdot x = (\zeta_1 \circ \cdots \circ \zeta_k) \cdot x = \zeta_1 \cdot (\zeta_2 \cdot (\cdots (\zeta_k \cdot x) \cdots )).$$

   Since $\zeta_k \cdot x = x$, the innermost term simplifies to $x$; then $\zeta_{k-1} \cdot x = x$, and continuing in this way we obtain $\rho \cdot x = x$. ◀

A key property of the $\mathsf{N}$-quantifier is the *Some/Any principle* [16, Prop. 4.10]:

▶ **Proposition A.1** (Some/Any Principle). *The following are equivalent:*

$$\mathsf{N}c.\varphi(c,\overline{x}) \iff \forall c.(c\#\overline{x} \Rightarrow \varphi(c,\overline{x})) \iff \exists c.(c\#\overline{x} \wedge \varphi(c,\overline{x})) \tag{4}$$

For multiple atoms, nested $\mathsf{N}$-quantifiers introduce distinctness conditions. We avoid writing these explicitly by fixing the following convention.

▶ **Notation A.2.** Let $\bar{c} = c_1, \dots, c_n$ be a list of pairwise distinct atoms. We abbreviate $\forall c_1 \dots \forall c_n.(\bigwedge_{i<j} c_i \neq c_j \Rightarrow \varphi)$ as $\forall \bar{c}.\,\varphi$, and similarly for $\exists \bar{c}.\,\varphi$.

▶ **Lemma A.3** (Generalised Some/Any Principle). *Let $\bar{c}$ be a list of atoms. Then:*

$$\mathsf{N}\bar{c}.\varphi(\bar{c},\overline{x}) \iff \forall \bar{c}.(\bar{c}\#\overline{x} \Rightarrow \varphi(\bar{c},\overline{x})) \iff \exists \bar{c}.(\bar{c}\#\overline{x} \wedge \varphi(\bar{c},\overline{x})) \tag{5}$$

**Proof.** By induction on the length of $\bar{c}$. The base case is (4). For the inductive step, write $\bar{c}' = \bar{c}, c_{n+1}$ and define $\psi(\bar{c},\overline{x}) := \mathsf{N}c_{n+1}.\varphi(\bar{c}, c_{n+1}, \overline{x})$. Then $\mathsf{N}\bar{c}'.\varphi \equiv \mathsf{N}\bar{c}.\psi$ and by the induction hypothesis and (4) this is equivalent to $\forall \bar{c} \forall c_{n+1}.(\bar{c}\#\overline{x} \wedge c_{n+1}\#\overline{x}, \bar{c} \Rightarrow \varphi)$. By Notation A.2, this is exactly $\forall \bar{c}'.(\bar{c}'\#\overline{x} \Rightarrow \varphi)$, as required. The existential case is analogous. ◀

▶ **Theorem 2.3.5** (Generalised Fix-Point). *Let $x \in \mathscr{X}$ and let $\pi$ be a permutation.*

1. *If $\pi$ is a $\bar{c}$-permutation and $A = \mathtt{mv}(\pi) \setminus \{\bar{c}\}$, then $\mathsf{N}\bar{c}.\pi \cdot x = x$ iff $A\#x$.*
2. *$\mathsf{N}\bar{c}.\pi \cdot x = x$ iff $(\mathsf{N}\bar{c}.\pi_{\bar{c}} \cdot x = x) \wedge (\pi_{\neg\bar{c}} \cdot x = x)$.*
   *Equivalently, if $A = \mathtt{mv}(\pi_{\bar{c}}) \setminus \{\bar{c}\}$, then $\mathsf{N}\bar{c}.\pi \cdot x = x$ iff $A\#x \wedge (\pi_{\neg\bar{c}} \cdot x = x)$.*

**Proof.**

1. ($\Rightarrow$) Assume $\mathsf{N}\bar{c}.\pi \cdot x = x$. By Lemma A.3, for some (equivalently, any) choice of $\bar{c}$ such that $\bar{c}\#x$, we have $\pi \cdot x = x$. Let $\eta$ be a cycle of $\pi$. Since $\pi$ is a $\bar{c}$-permutation, $\eta$ contains some atom $a \in \bar{c}$. As $a\#x$, Lemma 1 yields $\mathtt{mv}(\eta)\#x$. Hence $\mathtt{mv}(\pi)\#x$, and therefore $A\#x$.
   ($\Leftarrow$) Assume $A\#x$. Let $\bar{c}$ be such that $\bar{c}\#x$. Since $\pi$ is a $\bar{c}$-permutation, every cycle of $\pi$ contains an atom from $\bar{c}$ and all remaining atoms lie in $A$. Thus every cycle is disjoint from $\mathtt{supp}(x)$, so $\pi \cdot x = x$. By Lemma A.3, it follows that $\mathsf{N}\bar{c}.\pi \cdot x = x$.
2. If $\pi_{\bar{c}} = \mathtt{id}$ or $\pi_{\neg\bar{c}} = \mathtt{id}$, the result is immediate. Assume both are non-identity.
   We first show that $\mathsf{N}\bar{c}.\pi \cdot x = x$ iff $\mathsf{N}\bar{c}.(\pi_{\bar{c}} \cdot x = x \wedge \pi_{\neg\bar{c}} \cdot x = x)$.
   By Lemma A.3, both sides are equivalent to universal statements over $\bar{c}$ with the freshness condition $\bar{c}\#x$, $\mathtt{mv}(\pi) \setminus \{\bar{c}\}$. Hence it suffices to show that, for such $\bar{c}$, one has $\pi \cdot x = x$ iff both $\pi_{\bar{c}} \cdot x = x$ and $\pi_{\neg\bar{c}} \cdot x = x$.
   Assume $\pi \cdot x = x$. Let $\eta$ be a cycle of $\pi_{\bar{c}}$. Since $\pi_{\bar{c}}$ is a $\bar{c}$-permutation, $\eta$ contains some atom $a \in \bar{c}$. As $a\#x$, Lemma 1 yields $\mathtt{mv}(\eta)\#x$. Since this holds for every cycle $\eta$ of $\pi_{\bar{c}}$, and the cycles are disjoint, we obtain $\mathtt{mv}(\pi_{\bar{c}})\#x$, and therefore $\pi_{\bar{c}} \cdot x = x$ (equivalently, $\pi_{\bar{c}}^{-1} \cdot x = x$).
   From $\pi = \pi_{\bar{c}} \circ \pi_{\neg\bar{c}}$ and $\pi \cdot x = x$, we have $\pi_{\bar{c}} \cdot (\pi_{\neg\bar{c}} \cdot x) = x$. Applying $\pi_{\bar{c}}^{-1}$ to both sides and using that $\pi_{\bar{c}}^{-1} \cdot x = x$, it follows that $\pi_{\neg\bar{c}} \cdot x = x$.
   Conversely, if both $\pi_{\bar{c}}$ and $\pi_{\neg\bar{c}}$ fix $x$, then $\pi \cdot x = (\pi_{\bar{c}} \circ \pi_{\neg\bar{c}}) \cdot x = \pi_{\bar{c}} \cdot (\pi_{\neg\bar{c}} \cdot x) = x$.
   From the fact that $\mathsf{N}$ distributes over conjunction, we obtain $\mathsf{N}\bar{c}.(\pi_{\bar{c}} \cdot x = x \wedge \pi_{\neg\bar{c}} \cdot x = x)$ is equivalent to $\mathsf{N}\bar{c}.(\pi_{\bar{c}} \cdot x = x) \wedge \mathsf{N}\bar{c}.(\pi_{\neg\bar{c}} \cdot x = x)$. Since $\pi_{\neg\bar{c}}$ does not involve atoms from $\bar{c}$, the formula $\pi_{\neg\bar{c}} \cdot x = x$ is independent of $\bar{c}$, and hence, $\mathsf{N}\bar{c}.(\pi_{\neg\bar{c}} \cdot x = x)$ is equivalent to $\pi_{\neg\bar{c}} \cdot x = x$.
   Therefore, $\mathsf{N}\bar{c}.\pi \cdot x = x$ iff $\mathsf{N}\bar{c}.(\pi_{\bar{c}} \cdot x = x) \wedge (\pi_{\neg\bar{c}} \cdot x = x)$, and the final equivalence follows from item (1), applied to $\pi_{\bar{c}}$ and $B$. ◀

▶ **Proposition 2.3.8.** *Let $x \in \mathscr{X}$ and let $\pi = \gamma \circ \pi'$ and $\tau = \eta \circ \rho$ as in Definition 2.3.7. Then*

$$\text{И}\bar{\mathbf{c}}.(\pi \cdot x = x \wedge \tau \cdot x = x) \iff \text{И}\bar{\mathbf{c}}.(\pi[\gamma \mapsto \gamma \frown \eta] \cdot x = x \wedge \rho \cdot x = x).$$

**Proof.** ($\Rightarrow$) Assume $\text{И}\bar{\mathbf{c}}.(\pi \cdot x = x \wedge \tau \cdot x = x)$. By Lemma A.3, it suffices to fix $\bar{\mathbf{c}}$ such that $\bar{\mathbf{c}}\#x$ and $\bar{\mathbf{c}}\#((\mathtt{mv}(\pi) \cup \mathtt{mv}(\rho)) \setminus \{\bar{\mathbf{c}}\})$ and prove that $\pi[\gamma \mapsto \gamma \frown \eta] \cdot x = x$ and $\rho \cdot x = x$.

Since $\gamma$ is a $\bar{\mathbf{c}}$-cycle, it contains some atom $\boldsymbol{c}_i \in \{\bar{\mathbf{c}}\}$. As $\boldsymbol{c}_i\#x$ and $\pi \cdot x = x$, Lemma 2.3.4(1) applied to $\gamma$ implies that $\mathtt{mv}(\gamma)\#x$. Choose $a \in \mathtt{mv}(\gamma) \cap \mathtt{mv}(\eta)$, which exists by hypothesis. Then, $a\#x$. Since $\tau \cdot x = x$ and $\eta$ is a cycle of $\tau$ with $a \in \mathtt{mv}(\eta)$, another application of Lemma 2.3.4(1) gives $\mathtt{mv}(\eta)\#x$. Thus, $((\mathtt{mv}(\gamma) \cup \mathtt{mv}(\eta))\#x$, and therefore $(\gamma \frown \eta) \cdot x = x$ by the definition of support.

By definion, $\pi[\gamma \mapsto \gamma \frown \eta] = (\gamma \frown \eta) \circ \pi'$. Since $\pi \cdot x = x$ and $\gamma \cdot x = x$, it follows that $\pi' \cdot x = x$. Hence $\pi[\gamma \mapsto \gamma \frown \eta] \cdot x = x$. Finally, from $\tau \cdot x = x$ and $\eta \cdot x = x$, we conclude that $\rho \cdot x = x$.

($\Leftarrow$) Assume $\text{И}\bar{\mathbf{c}}.(\pi[\gamma \mapsto \gamma \frown \eta] \cdot x = x \wedge \rho \cdot x = x)$. Fix $\bar{\mathbf{c}}$ such that $\bar{\mathbf{c}}\#x$ and $\bar{\mathbf{c}}\#((\mathtt{mv}(\pi) \cup \mathtt{mv}(\rho)) \setminus \{\bar{\mathbf{c}}\})$.

Since $\gamma \frown \eta$ contains all atoms of $\gamma$, it contains at least one atom from $\bar{\mathbf{c}}$. Thus, from $\pi[\gamma \mapsto \gamma \frown \eta] \cdot x = x$, Lemma 2(1) yields $((\mathtt{mv}(\gamma) \cup \mathtt{mv}(\eta))\#x$. In particular, by the definition of support, $\gamma \cdot x = x, \eta \cdot x = x$, and $(\gamma \frown \eta) \cdot x = x$. By definition, $\pi[\gamma \mapsto \gamma \frown \eta] = (\gamma \frown \eta) \circ \pi'$. Since $(\gamma \frown \eta) \cdot x = x$ and $\pi[\gamma \mapsto \gamma \frown \eta] \cdot x = x$, we obtain $\pi' \cdot x = x$. Therefore, $\pi \cdot x = x$. Moreover, since $\rho \cdot x = x$ and $\eta \cdot x = x$, we conclude that $\tau \cdot x = x$. ◀

▶ **Corollary 2.3.10.** *Let $\pi$ and $\rho$ be $\bar{\mathbf{c}}$-permutations such that $(\mathtt{mv}(\pi) \cap \mathtt{mv}(\rho)) \setminus \{\bar{\mathbf{c}}\} \neq \emptyset$. Then*

$$\text{И}\bar{\mathbf{c}}.(\pi \cdot x = x \wedge \rho \cdot x = x) \iff \text{И}\bar{\mathbf{c}}.(a_1 \ldots a_k\ \mathbf{c}') \cdot x = x,$$

*where $\{a_1, \ldots, a_k\} = (\mathtt{mv}(\pi) \cup \mathtt{mv}(\rho)) \setminus \{\bar{\mathbf{c}}\}$ (in any order), and $\mathbf{c}' \in \{\bar{\mathbf{c}}\}$.*

**Proof.** Define $S = (\mathtt{mv}(\pi) \cup \mathtt{mv}(\rho)) \setminus \{\bar{\mathbf{c}}\} = \{a_1, \ldots, a_k\}$. Since $\pi$ and $\rho$ are $\bar{\mathbf{c}}$-permutations, by Theorem 2.3.5(1), we have:

$$\begin{aligned}
\text{И}\bar{\mathbf{c}}.(\pi \cdot x = x \wedge \rho \cdot x = x) &\iff (\mathtt{mv}(\pi) \setminus \{\bar{\mathbf{c}}\})\#x \wedge (\mathtt{mv}(\rho) \setminus \{\bar{\mathbf{c}}\})\#x \\
&\iff ((\mathtt{mv}(\pi) \cup \mathtt{mv}(\rho)) \setminus \{\bar{\mathbf{c}}\})\#x \\
&\iff S\#x \\
&\iff \text{И}\bar{\mathbf{c}}.(a_1 \ldots a_k\ \mathbf{c}') \cdot x = x
\end{aligned}$$

and the result follows. ◀