Nominal Sets

Contents

Preface

Names and constructs that bind names are ubiquitous in programming languages. Nominal sets provide a mathematical theory of structures involving names that was introduced by the author and Jamie Gabbay about 10 years ago. The theory is based on some simple, but subtle ideas to do with permutations of names and the notion of 'finitely supported' mathematical structures which first arose in mathematical logic in the 1930s. The theory has turned out to have some interesting logical and computational properties, with applications to programming language semantics, machine-assisted theorem proving and the design of functional and logical metaprogramming languages.

These notes on the underlying theory of nominal sets accompany a four-lecture course on 'Nominal Sets and Their Applications' given at the Midland Graduate School 2011. The material forms part of a book on Nominal Sets that is currently in preparation. I would be very grateful if readers would notify me of any errors they detect, or of any suggestions they have for improving the presentation.

Andrew Pitts
Andrew.Pitts@cl.cam.ac.uk
University of Cambridge
April 2011

1

Permutations

The characteristic feature of the nominal sets approach to the syntax and semantics of formal languages is the use of permutations of names. As such, it is part of an important theme in mathematics with a well-developed body of work, the study of symmetry and the theory of groups. We need only a small amount of that theory, which we review in this chapter.

1.1 The category of G-sets

A *group* is a set G equipped with an element $e \in G$ (the group *unit*), a function $(g,' g') \in G \times G \mapsto g\,g' \in G$ (the group *multiplication* operation) and a function $g \in G \mapsto g^{-1} \in G$ (the group *inverse* operation). This structure is required to satisfy

$$(g\,g')\,g'' = g\,(g'\,g'') \tag{1.1}$$

$$e\,g = g = g\,e \tag{1.2}$$

$$g^{-1}\,g = e = g\,g^{-1} \tag{1.3}$$

for all $g, g', g'' \in G$. A *homomorphism* of groups is a function $\theta : G \to G'$ satisfying

$$\theta\,e = e \tag{1.4}$$

$$\theta(g\,g') = (\theta\,g)(\theta\,g') \tag{1.5}$$

$$\theta(g^{-1}) = (\theta\,g)^{-1} \tag{1.6}$$

for all $g, g' \in G$. A *subgroup* of a group G is a subset $G' \subseteq G$ that contains e and is closed under the group multiplication and inverse operations.

Notation In this book we specify functions using a variety of informal notations. If an expression $e(x)$ denotes an element of a set Y as x ranges over the elements of set X, then the function $X \to Y$ it determines will be denoted by either of the notations

$$\begin{aligned} x \in X &\mapsto e(x) \in Y \\ \lambda x \in X &\to e(x). \end{aligned} \tag{1.7}$$

When the set X has some structure, we use notations for patterns; for example if $X = X_1 \times X_2$ is a cartesian product, we write $\lambda(x_1, x_2) \in X_1 \times X_2 \to e(x_1, x_2)$.

Example 1.1 If A is a set, then a *permutation* of A is a bijection π from A to itself. The composition $\pi' \circ \pi$ of two functions that are permutations is another such, as is the inverse function π^{-1} of a permutation; and the identity function id on A is a permutation. Therefore, taking the group multiplication to be function composition, the permutations of A form a group, called the *symmetric group* on the set A and denoted $\mathrm{S}\,A$. It is classic result of group theory (*Cayley's Theorem*, a special case of the *Yoneda Lemma* in category theory) that every group is a subgroup of a symmetric group.

An *action* of a group G on a set X is a function $G \times X \to X$ assigning to each $(g, x) \in G \times X$ an element $g \cdot x$ of X satisfying

$$g \cdot (g' \cdot x) = (g\,g') \cdot x \tag{1.8}$$

$$e \cdot x = x \tag{1.9}$$

for all $g, g' \in G$ and $x \in X$. This is equivalent to specifying a homomorphism of groups $G \to \mathrm{S}\,X$ (see Exercise 1.1).

Definition 1.2 If G is a group, then a *G-set* is a set X equipped with an action of G on X. We will usually refer to a G-set by naming its underlying set X, using the same notation $\_\cdot\_$ for all group actions, whatever the set X. G-sets are the objects of a category $[G, \mathbf{Set}]$ whose morphism from X to X' are *equivariant functions*, that is, functions $F : X \to Y$ satisfying

$$F(g \cdot x) = g \cdot (F\,x) \tag{1.10}$$

for all $g \in G$ and $x \in X$. Composition and identities in the category are the same as in the category **Set** of sets and functions.

Example 1.3 If G is any subgroup of the symmetric group $\mathrm{S}\,A$, we get a G-set with underlying set A by taking the G-action to be given by function application: $\pi \cdot a = \pi\,a$.

Example 1.4 Let Σ be a (single-sorted) *algebraic signature*. Thus $\Sigma = (\Sigma_n \mid n \in \mathbb{N})$ is a countably infinite family of sets. The elements of each set Σ_n are the n-ary operations of the signature. The set $\Sigma[X]$ of *algebraic terms* over Σ with variables drawn from some set X is inductively defined by the following rules.

$$\frac{x \in X}{x \in \Sigma[X]} \qquad \frac{t_1 \in \Sigma[X] \quad \cdots \quad t_n \in \Sigma[X] \quad \mathsf{op} \in \Sigma_n}{\mathsf{op}(t_1\ ,\ \cdots\ ,\ t_n) \in \Sigma[X]}$$

There is an action of $\mathrm{S}\,X$ on $\Sigma[X]$ given by applying a finite permutation to variables where they occur in algebraic terms:

$$\begin{aligned} \pi \cdot x &= \pi\, x \\ \pi \cdot \mathsf{op}(t_1\ ,\ \cdots\ ,\ t_n) &= \mathsf{op}(\pi \cdot t_1\ ,\ \cdots\ ,\ \pi \cdot t_n). \end{aligned} \tag{1.11}$$

1.2 Products and coproducts

Given a group G and G-sets $X_1, \ldots, X_n$, we make the cartesian product

$$X_1 \times \cdots \times X_n \triangleq \{(x_1, \ldots, x_n) \mid x_1 \in X_1 \wedge \cdots \wedge x_n \in X_n\} \tag{1.12}$$

into a G-set by defining the group action coordinate-wise:

$$g \cdot (x_1, \ldots, x_n) \triangleq (g \cdot x_1, \ldots, g \cdot x_n). \tag{1.13}$$

In case $n = 0$, the cartesian product is just a singleton set $1 = \{()\}$ and the action is $g \cdot () = ()$. Definition (1.13) ensures that the projection functions from a product of G-sets to one of its components

$$\begin{aligned} &\mathrm{proj}_i : X_1 \times \cdots \times X_n \to X_i \\ &\mathrm{proj}_i \triangleq \lambda(x_1, \ldots, x_n) \in X_1 \times \cdots \times X_n \to x_i \end{aligned} \tag{1.14}$$

are all equivariant and hence give morphisms in $[G, \mathbf{Set}]$. Indeed they make $X_1 \times \cdots \times X_n$ into the categorical product of the objects X_i in $[G, \mathbf{Set}]$. For if $(F_i : X \to X_i \mid i = 1..n)$ are some equivariant functions, then the unique function $\langle F_1, \ldots, F_n\rangle : X \to X_1 \times \cdots \times X_n$ satisfying $\mathrm{proj}_i \circ \langle F_1, \ldots, F_n\rangle = F_i$ (for $i = 1..n$) is easily seen to be equivariant.

Example 1.5 Given any set X, the second projection function $\lambda(g, x) \in G \times I \to x$ is trivially a G-action. We call X equipped with this action the *discrete G-set* on X. If $F : X \to Y$ is an equivariant function whose domain X is a discrete G-set, then for each $x \in X$, $F\,x = F(g \cdot x) = g \cdot (F\,x)$. Thus F maps X into the subset

$$\Gamma\, Y \triangleq \{y \in Y \mid (\forall g \in G)\ g \cdot y = y\}. \tag{1.15}$$

(See Exercise 1.2.) The terminal object 1 is a discrete G-set and the *global sections* $1 \to X$ of any G-set X correspond to elements of $\Gamma\, X$. Note that $\Gamma\, X$ may be empty even if X as a set is non-empty. For example, when $G = \mathrm{S}\,A$ is the symmetric group on a set A, the $\mathrm{S}\,A$-set A from Example 1.3 satisfies $\Gamma\, A = \emptyset$ so long as A has at least two elements. In this case $\mathrm{proj}_1, \mathrm{proj}_2 : A \times A \to A$ are different morphisms in $[\mathrm{S}\,A, \mathbf{Set}]$ that have equal compositions with every $1 \longrightarrow A$ (since there are no such global sections). Thus $[\mathrm{S}\,A, \mathbf{Set}]$ is not a well-pointed category. (In general a category with a terminal object is *well-pointed* if any two morphisms with equal domain and codomain are equal if their compositions with any global section are equal.)

Example 1.6 The group G is itself a G-set once we endow it with the *conjugation* action:

$$g \cdot g' \triangleq g\,g'\,g^{-1}. \tag{1.16}$$

This is not the only possible action of G on itself, unless $G = \{e\}$ is trivial; see Exercise 1.3. However it has the distinction of making the action function $F \triangleq \lambda(g, x) \in G \times X \to g \cdot x$ for any G-set X into an equivariant function $F : G \times X \longrightarrow X$. For we have $F(g \cdot g', g \cdot x) = (g\,g' g^{-1}) \cdot (g \cdot x) = (g\,g' g^{-1} g) \cdot x = (g\,g') \cdot x = g \cdot F(g', x)$.

We make the disjoint union

$$X_1 + \cdots + X_n \triangleq \{(i, x) \mid i \in \{1, \ldots, n\} \wedge x \in X_i\} \tag{1.17}$$

into a G-set by defining the group action as follows:

$$g \cdot (i, x) \triangleq (i, g \cdot x). \tag{1.18}$$

It is easy to see that definitions (1.13) and (1.18) inherit the required properties (1.8) and (1.9) from the actions for each X_i. Definition (1.18) ensures that the functions injecting a G-set into a disjoint union of G-sets

$$\begin{aligned} &\mathrm{inj}_i : X_i \longrightarrow X_1 + \cdots + X_n \\ &\mathrm{inj}_i \triangleq \lambda x \in X_i \to (i, x) \end{aligned} \tag{1.19}$$

are morphisms in $[G, \mathbf{Set}]$ and make $X_1 + \cdots + X_n$ into the coproduct of the objects X_i in $[G, \mathbf{Set}]$.

These properties of cartesian product and disjoint union extend from the finite to the infinite case. Thus if $(X_i \mid i \in I)$ is a family of G-sets indexed by the elements of a set I, then the cartesian product

$$\textstyle\prod_{i \in I} X_i \triangleq \{(x_i \mid i \in I) \mid (\forall i \in I)\ x_i \in X_i\} \tag{1.20}$$

equipped with the G-action $g \cdot (x_i \mid i \in I) = (g \cdot x_i \mid i \in I)$ is the product of the objects X_i in $[G, \mathbf{Set}]$. Similarly, the disjoint union

$$\textstyle\sum_{i \in I} X_i \triangleq \{(i, x) \mid i \in I \wedge x_i \in X_i\} \tag{1.21}$$

equipped with the G-action $g \cdot (i, x) = (i, g \cdot x)$ is their coproduct in $[G, \mathbf{Set}]$.

1.3 Natural numbers

The coproduct $\sum_{i \in I} X_i$ in the case $I = \mathbb{N} = \{0, 1, 2, \dots\}$ and each X_i is the terminal 1, is necessarily a *natural number object* in $[G, \mathbf{Set}]$. In other words it is a G-set N equipped with equivariant functions

$$1 \xrightarrow{\text{zero}} N \xrightarrow{\text{suc}} N \tag{1.22}$$

with the universal property that for any other such diagram in $[G, \mathbf{Set}]$

$$1 \xrightarrow{X_0} X \xrightarrow{F} X \tag{1.23}$$

there is a unique equivariant function $\text{iter}\, X_0\, F$ making

$$\begin{tikzcd}
1 \arrow[r, "\text{zero}"] \arrow[d, equal] & N \arrow[r, "\text{suc}"] \arrow[d, dashed, "\text{iter}\, X_0\, F"] & N \arrow[d, dashed, "\text{iter}\, X_0\, F"] \\
1 \arrow[r, "X_0"'] & X \arrow[r, "F"'] & X
\end{tikzcd} \tag{1.24}$$

commute.

Note that $\sum_{i \in I} X_i$ is discrete (Example 1.5) when all the X_i are discrete G-sets.In particular we can identify N with the discrete G-set on the set $\mathbb{N}$ of natural numbers equipped with the usual zero and successor functions: $\text{zero}() = 0$, $\text{suc}\, n = n + 1$. Given (1.23), the unique function $\text{iter}\, X_0\, F : \mathbb{N} \to X$ making (1.24) commute is recursively defined by

$$\begin{aligned}
\text{iter}\, X_0\, F\, 0 &= X_0() \\
\text{iter}\, X_0\, F\, (n+1) &= F\, (\text{iter}\, X_0\, F\, n).
\end{aligned}$$

It is equivariant because $\mathbb{N}$ is discrete and X_0 and F are equivariant.

1.4 Functions

Theorem 1.7 *For any group G, the category* $[G, \mathbf{Set}]$ *is cartesian closed.*

Proof This theorem can be deduced from the more general fact that categories of **Set**-valued functors are cartesian closed (see Johnstone, 2002, Proposition 1.5.5, for instance); this is because each group G can be regarded as a category with a single object and whose morphisms are the group elements. However, unlike for functor categories in general, in this particular case the cartesian closed structure is almost as simple as that of the category of sets itself and is worth describing explicitly. We saw in the previous section that $[G, \mathbf{Set}]$ has finite products, inherited from **Set**. So we just have to describe exponentials.

If X and Y are G-sets for a group G, then we can make the set Y^X of functions with domain X and codomain Y into a G-set by defining the action of $g \in G$ on a function $F \in Y^X$ to be

$$g \cdot F \triangleq \lambda x \in X \to g \cdot (F\,(g^{-1} \cdot x)). \tag{1.25}$$

The reader should check that this does give a group action on functions (Exercise 1.4). Equivariant functions $F : X \longrightarrow Y$ are precisely the elements of Y^X that satisfy $g \cdot F = F$ for all $g \in G$; we leave this as an exercise (Exercise 1.5).

Definition (1.25) ensures that the application function

$$\begin{aligned} &\mathrm{app} : Y^X \times X \longrightarrow Y \\ &\mathrm{app} \triangleq \lambda(F, x) \in Y^X \times X \to F\,x \end{aligned} \tag{1.26}$$

is equivariant. For we have

$$\begin{aligned}
&\mathrm{app}(g \cdot F, g \cdot x) \\
=\;& \{\text{definition of app}\} \\
&(g \cdot F)(g \cdot x) \\
=\;& \{\text{definition of } g \cdot F\} \\
&g \cdot (F\,(g^{-1} \cdot (g \cdot x))) \\
=\;& \{(1.8), (1.3) \text{ and } (1.9)\} \\
&g \cdot (F\,x) \\
=\;& \{\text{definition of app}\} \\
&g \cdot \mathrm{app}(F, x).
\end{aligned}$$

The equivariance of the currying function

$$\begin{aligned} &\mathrm{curry} : Y^{Z \times X} \longrightarrow (Y^X)^Z \\ &\mathrm{curry} \triangleq \lambda F' \in Y^{Z \times X}, z \in Z, x \in X \to F'(z, x) \end{aligned} \tag{1.27}$$

is a similar calculation (Exercise 1.6).

It follows that $\mathrm{app} : Y^X \times X \longrightarrow Y$ gives the exponential of Y by X in $[G, \mathbf{Set}]$. For if we have $F' : Z \times X \longrightarrow Y$ in $[G, \mathbf{Set}]$, then from the usual properties of application and currying in **Set**, we have that $\mathrm{curry}\,F' \in (Y^X)^Z$ is the unique function satisfying $\mathrm{app} \circ (\mathrm{curry}\,F' \times \mathrm{id}_X) = F'$. So we just have to see that it is equivariant and hence a morphism $Z \longrightarrow Y^X$ in $[G, \mathbf{Set}]$; but this follows from the equivariance of application, of currying and of F', using Exercise 1.5. □

1.5 Power sets

Let

$$\mathbb{B} \triangleq \{\mathrm{true}, \mathrm{false}\} \tag{1.28}$$

denote the discrete G-set (Example 1.5) on a two element set. Given any G-set X, we can use the bijection between functions $X \to \mathbb{B}$ and subsets of X to transfer the G-action on $\mathbb{B}^X$ to one on the powerset

$$\mathrm{P}\, X \triangleq \{S \mid S \subseteq X\}. \tag{1.29}$$

Converting a subset $S \subseteq X$ to a corresponding function $\chi_S : X \to \mathbb{B}$

$$\chi_S\ x \triangleq \begin{cases} \text{true} & \text{if } x \in S \\ \text{false} & \text{if } x \notin S \end{cases} \tag{1.30}$$

and acting by $g \in G$ to get $g \cdot \chi_S$, the subset $\{x \in X \mid (g \cdot \chi_S)\, x = \text{true}\}$ gives the action of g on S. Since $\mathbb{B}$ is discrete, $(g \cdot \chi_S)\, x = g \cdot (\chi_S\ (g^{-1} \cdot x)) = \chi_S\ (g^{-1} \cdot x)$. So $(g \cdot \chi_S)\, x = \text{true}$ if and only if $g^{-1} \cdot x \in S$. Therefore

$$g \cdot S = \{x \in S \mid g^{-1} \cdot x \in S\}. \tag{1.31}$$

Note that $g^{-1} \cdot x \in S \Leftrightarrow (\exists x' \in S)\ x = g \cdot x'$ and thus

$$g \cdot S = \{g \cdot x \mid x \in S\}. \tag{1.32}$$

The associativity and unit properties of the G-action on functions transfer to give the required properties for an action on subsets: $g \cdot (g' \cdot S) = (g\, g') \cdot S$ and $e \cdot S = S$. Note also that subset inclusion is preserved by the G-action:

$$S \subseteq S' \Rightarrow g \cdot S \subseteq g \cdot S'. \tag{1.33}$$

Definition 1.8 The *equivariant subsets* of a G-set X are those $S \subseteq X$ which are closed under the G-action on X in the sense that for all $x \in X$ and $g \in G$

$$x \in S \Rightarrow g \cdot x \in S. \tag{1.34}$$

So by (1.32), $S \subseteq X$ is an equivariant subset if $g \cdot S \subseteq S$ holds for all $g \in G$. Note that in this case we have

$$S = e \cdot S = (g\, g^{-1}) \cdot S = g \cdot (g^{-1} \cdot S) \subseteq g \cdot S$$

by (1.33); and hence $g \cdot S = S$ for any $g \in G$. Thus equivariant subsets are precisely the elements of $\mathrm{P}\ X$ that are fixed by the action of any $g \in G$. Compare this with the similar characterization of equivariant functions as elements of Y^X in Exercise 1.5.

A morphism $F : Y \to X$ in a category $\mathbf{C}$ is a *monomorphism* if $F \circ F_1 = F \circ F_2$ implies $F_1 = F_2$ for any $F_1, F_2 : \cdot \to Y$. We write $F : Y \rightarrowtail X$ to indicate that F is a monomorphism. The collection of monomorphisms with codomain X is preordered by the relation

$$(F_1 : Y_1 \rightarrowtail X) \leq (F_2 : Y_2 \rightarrowtail X) \Leftrightarrow (\exists F : Y_1 \to Y_2)\ F_1 = F_2 \circ F. \tag{1.35}$$

A *subobject* of X is an equivalence class of monomorphisms with codomain X for the equivalence relation generated by $\leq$. When the category is $[G, \mathbf{Set}]$ it is not hard to see that the subobjects of a G-set X are in bijection with the equivariant subsets of X; and under this bijection the partial order on subobjects induced by (1.35) corresponds to subset inclusion. (See Exercise 1.7.)

We saw above (Theorem 1.7) that $[G, \mathbf{Set}]$ is a cartesian closed category. It also has a subobject classifier and hence is a topos (Johnstone, 2002, Example 2.1.4). This means there is an object Ω equipped with a morphism $\top : 1 \to \Omega$ so that for any monomorphism $F : Y \rightarrowtail X$ there is a unique morphism $\chi_F : X \to \Omega$ making

$$\begin{tikzcd}
Y \arrow[r, "\langle\rangle"] \arrow[d, tail, "F"'] & 1 \arrow[d, "\top"] \\
X \arrow[r, "\chi_F"'] & \Omega
\end{tikzcd}$$

a pullback square in $[G, \mathbf{Set}]$. In fact $[G, \mathbf{Set}]$ is Boolean (Johnstone, 2002, p 38) since $\Omega = 1 + 1$ is just the discrete two-element set G-set $\mathbb{B}$. For if $F : Y \rightarrowtail X$ corresponds to the equivariant subset $S \subseteq X$, then $\chi_F : X \to \Omega$ is the characteristic function (1.30) which is an equivariant function, because S is an equivariant subset.

Since $[G, \mathbf{Set}]$ is a Boolean topos, it provides a model of classical higher-order logic (Johnstone, 2002, chapter D4). Just as for the cartesian closed structure, the interpretation of higher-order logic in $[G, \mathbf{Set}]$ is the same as in $\mathbf{Set}$:

Proposition 1.9 *Let X and Y be G-sets. The following are equivariant subsets:*

1. *Truth* $X \subseteq X$.
2. *Equality* $\{(x, x') \in X \times X \mid x = x'\} \subseteq X \times X$.
2. *Membership* $\{(x, S) \in X \times \mathrm{P}(X) \mid x \in S\} \subseteq X \times \mathrm{P}(X)$.

The following are equivariant functions:

4. *Conjunction* $\_ \cap \_ : \mathrm{P}(X) \times \mathrm{P}(X) \to \mathrm{P}(X)$.
5. *Negation* $\neg : \mathrm{P}(X) \to \mathrm{P}(X)$, *where* $\neg S \triangleq \{x \in X \mid x \notin S\}$.
6. *Universal quantification* $\bigcap : \mathrm{P}(\mathrm{P}(X)) \to \mathrm{P}(X)$, *where* $\bigcap \mathcal{S} \triangleq \{x \in X \mid (\forall S \in \mathcal{S})\ x \in S\}$.
7. *Substitution* $f^* : \mathrm{P}(Y) \to \mathrm{P}(X)$, *where* $f : X \to Y$ *is an equivariant function and* $f^* S \triangleq \{x \in X \mid f(x) \in S\}$.
8. *Comprehension* $\mathrm{compr} : \mathrm{P}(X \times Y) \to \mathrm{P}(Y)^X$, *where* $\mathrm{compr}\, S \triangleq \lambda x \in X \to \{y \in Y \mid (x, y) \in S\}$.

Proof These equivariance properties follow easily from the definition of the action of G on subsets. For example

$$x \in g \cdot (\neg S) \Leftrightarrow g^{-1} \cdot x \in \neg S \Leftrightarrow g^{-1} \cdot x \notin S \Leftrightarrow x \notin g \cdot S \Leftrightarrow x \in \neg(g \cdot S)$$

so that $g \cdot (\neg S) = \neg(g \cdot S)$. □

The proposition gives a very rich collection of equivariant subsets. Consider the formulas of classical higher-order logic; they are built up from atomic formulas using equality, membership, the propositional connectives, and quantification over iterated product, function and power types. If $\varphi(x_1, \dots, x_n)$ is such a formula with free variables as indicated, and if each variable x_i is interpreted as ranging over a G-set X_i, then the Tarski interpretation of φ (with product, function and power types interpreted as cartesian products, exponentials and powersets) determines a subset of $X_1 \times \cdots \times X_n$ in the usual way. If the function and relation symbols in φ are all interpreted by equivariant functions and subsets, then Theorem 1.9 implies that the interpretation of $\varphi(x_1, \dots, x_n)$ is an equivariant subset of $X_1 \times \cdots \times X_n$. Thus we have:

Equivariance Principle *Any function or relation that is defined from equivariant functions and relations using classical higher-order logic is itself equivariant.*

We will use this principle to avoid proving that particular constructs are equivariant on a case-by-case basis. The next section gives a first example of this. We end this section with two warnings about the Equivariance Principle.

Note 1.10 In applying the Equivariance Principle, one must take into account *all* the parameters upon which a particular construction depends. For example, regarding G as a G-set as in Example 1.6, we saw there that for any G-set X the action

$$\begin{aligned} &F : G \times X \to X \\ &F \triangleq \lambda(g, x) \in G \times X \to g \cdot x \end{aligned}$$

is an equivariant function. However, if we fix upon a particular $g_0 \in G$, then the function $F_0 = F(g_0, -) : X \to X$ is not in general equivariant unless G is a commutative group (or $g_0 = e$), since $g \cdot F_0\, x = (g\, g_0) \cdot x$ whereas $F_0(g \cdot x) = (g_0\, g) \cdot x$.

Note 1.11 Classical higher-order logic is sometimes formulated using Hilbert's ε-operation, $\varepsilon x.\varphi(x)$, satisfying $(\forall x \in X)\ \varphi(x) \Rightarrow \varphi(\varepsilon x.\varphi(x))$. Unless G is a trivial group, such an operation cannot be equivariant—see Exercise 1.8. Thus the Equivariance Principle does not apply to constructions employing this strong form of choice. Toposes of the form $[G, \mathbf{Set}]$ do satisfy a weaker, internal version of the Axiom of Choice (Johnstone, 2002, Examples 4.5.2(b)). However, when we move from considering equivariance to finite support in the next chapter, even that form of choice fails; see Sect. 2.6.

1.6 Partial functions

Given sets X and Y, the set $X \rightharpoonup Y$ of partial functions from X to Y is the subset of $\mathrm{P}\,(X \times Y)$ consisting of all subsets $F \subseteq X \times Y$ that are *single-valued*

$$(\forall x \in X)(\forall y, y' \in Y)\,(x,y) \in F \wedge (x,y') \in F \Rightarrow y = y'. \tag{1.36}$$

We write $F\,x \equiv y$ to mean that F is defined at $x \in X$ and takes value $y \in Y$. Thus

$$F\,x \equiv y \Leftrightarrow (x,y) \in F. \tag{1.37}$$

More generally, given partial functions $F_1 \in X_1 \rightharpoonup Y$ and $F_2 \in X_2 \rightharpoonup Y$, for all $x_1 \in X_1$ and $x_2 \in X_2$ we define

$$F_1\,x_1 \equiv F_2\,x_2 \Leftrightarrow (\forall y \in Y)\;(x_1,y) \in F_1 \Leftrightarrow (x_2,y) \in F_2. \tag{1.38}$$

($F\,x \equiv y$ is the special case of this when $F_1 = F$ and $F_2 = \mathrm{id}_Y$.) More generally still, but less formally, if e and e' are expressions denoting partially defined values we write $e \equiv e'$ to mean that e is defined if and only if e' is and in that case they are equal values. This is sometimes referred to as *Kleene equivalence*.

The *domain of definition* of a partial function $F \in X \rightharpoonup Y$ is

$$\mathrm{Dom}\,F \triangleq \{x \in X \mid (\exists y \in Y)\;(x,y) \in F\}. \tag{1.39}$$

(We reserve the notation 'dom' for the domain of a morphism in a category. Thus in the category of sets and partial functions, given $F \in X \rightharpoonup Y$ we have $\mathrm{dom}\,F = X$, but $\mathrm{Dom}\,F$ may be strictly smaller than X.)

If X and Y are G-sets, then using the Equivariance Principle we have that $X \rightharpoonup Y$ is an equivariant subset of $\mathrm{P}\,(X \times Y)$ and hence a G-set. As the next result shows, the G-action on partial functions agrees with the action on total functions (1.25).

Proposition 1.12 *If X and Y are G-sets and $F \in X \rightharpoonup Y$, then for all $g \in G$ and $x \in X$*

$$(g \cdot F)\,x \equiv g \cdot (F(g^{-1} \cdot x)). \tag{1.40}$$

Proof If $(g \cdot F)\,x$ is defined, say $(x,y) \in g \cdot F$, then $(g^{-1} \cdot x, g^{-1} \cdot y) = g^{-1} \cdot (x,y) \in F$; so $F(g^{-1} \cdot x)$ is defined and equal to $g^{-1} \cdot y$ and therefore $g \cdot (F(g^{-1} \cdot x)) \equiv g \cdot (g^{-1} \cdot y) = y$.

Conversely if $g \cdot (F(g^{-1} \cdot x)) \equiv y$, then the subexpression $F(g^{-1} \cdot x)$ must also be defined, that is, $(g^{-1} \cdot x, y') \in F$ for some y' with $g \cdot y' = y$. Hence $(x,y) = (g \cdot (g^{-1} \cdot x), g \cdot y') = g \cdot (g^{-1} \cdot x, y') \in g \cdot F$ and thus $(g \cdot F)\,x \equiv y$. □

The elements of Y^X are those partial functions $F \in X \rightharpoonup Y$ that are also *total*

$$(\forall x \in X)(\exists y \in Y)\;(x,y) \in F. \tag{1.41}$$

Thus Y^X is an equivariant subset of $\mathrm{P}\,(X \times Y)$ and the proposition shows that the action of G on subsets (1.31) agrees with the action on functions (1.25) when restricted to Y^X.

1.7 Quotient sets

We write $X/\!\sim$ for the set of equivalence classes $[x]_\sim \triangleq \{x' \in X \mid x \sim x'\}$ of an equivalence relation on a set X. Given a group G, an *equivariant equivalence relation* on a G-set X is simply an equivalence relation on the underlying set of X which is equivariant as a subset of $X \times X$ (see Definition 1.8). In this case, for each $g \in G$, the function $g \cdot \_ : X \to X$ respects the equivalence relation and hence induces a function on equivalence classes, $X/\!\sim \;\to X/\!\sim$, that we also write as $g \cdot \_$. Thus

$$g \cdot [x]_\sim = [g \cdot x]_\sim \tag{1.42}$$

and we get a G-action on $X/\!\sim$, which we call a *quotient G-set*.

Note that by virtue of (1.42), the quotient function

$$\begin{aligned} &q : X \to X/\!\sim \\ &q \triangleq \lambda x \in X \to [x]_\sim \end{aligned} \tag{1.43}$$

is equivariant.

Any function $F : X \to Y$ that respects $\sim$, in the sense that

$$x \sim x' \Rightarrow F\,x = F\,x'$$

holds, induces a unique function $\overline{F}$ making

$$\begin{tikzcd} X \arrow[r, "q"] \arrow[dr, "F"'] & X/\!\sim \arrow[d, dashed, "\overline{F}"] \\ & Y \end{tikzcd}$$

commute. Note that $\overline{F}$ is definable within higher-order logic from $\sim$ and F. Thus assuming $\sim$ is an equivariant equivalence relation, by the Equivariance Principle, if F is equivariant, then so is $\overline{F}$.

1.8 Finitary permutations

So far in this chapter we have considered group actions for an arbitrary group G. Now we specialize to the case we need in the rest of the book, groups of finitary permutations and their actions.

Definition 1.13 A permutation $\pi \in \mathrm{S}\,A$ is *finitary* if $\{a \in A \mid \pi\,a \neq a\}$ is a finite subset of A. Note that $\mathrm{id} \in \mathrm{S}\,A$ is finitary and that the composition and inverse of finitary permutations are finitary. Therefore we get a subgroup of $\mathrm{S}\,A$ of finitary permutations, denoted $\mathrm{Perm}\,A$.

The *transposition* (also known as *swapping*) of a pair of elements $a_1, a_2 \in A$ is the finitary permutation $(a_1\ a_2) \in \mathrm{Perm}\,A$ given for all $a \in A$ by

$$(a_1\ a_2)\,a \triangleq \begin{cases} a_2 & \text{if } a = a_1 \\ a_1 & \text{if } a = a_2 \\ a & \text{otherwise.} \end{cases} \tag{1.44}$$

Note that this definition makes sense even if $a_1 = a_2$, in which case $(a_1\ a_2) = \mathrm{id}$. If $a_1 \neq a_2$, then $(a_1\ a_2)$ is a 2-cycle, where in general the *n-cycle* $(a_1\, a_2\, a_3\, \cdots\, a_n)$ is the element of $\mathrm{Perm}\,A$ that maps a_1 to a_2, a_2 to a_3, $\ldots$, a_{n-1} to a_n, and a_n to a_1, while leaving all other elements fixed; here $a_1, \ldots, a_n$ have to be n mutually distinct atoms with $n \geq 2$.

Transpositions play a prominent role in what follows, because they generate the group $\mathrm{Perm}\,A$.

Theorem 1.14 *Every element π of the group* $\mathrm{Perm}\,A$ *of finitary permutations on a set A is equal to the composition of a finite sequence of transpositions* $(a\ a')$ *with*

$$\pi\, a \neq a \neq a' \neq \pi\, a'. \tag{1.45}$$

(The sequence may be empty, in which case its composition is by definition the identity function.)

Proof We argue by induction on the size of the finite set $\{a \in A \mid \pi\, a \neq a\}$. In the base case when it is empty, π must be the identity function, which is the composition of the empty sequence of transpositions. For the induction step, given $\pi \in \mathrm{Perm}\,A$ with $\{a \mid \pi\, a \neq a\}$ non-empty, pick some a_0 in that set and consider $\pi' = \pi \circ (a_0\ \pi^{-1} a_0)$. It satisfies $\pi' a_0 = a_0$ and $(\forall a \neq a_0)\ \pi\, a = a \Rightarrow \pi' a = a$. Hence

$$\{a \mid \pi' a \neq a\} \subseteq \{a \mid \pi\, a \neq a\} - \{a_0\}. \tag{1.46}$$

In particular $\{a \mid \pi' a \neq a\}$ is strictly smaller than $\{a \mid \pi\, a \neq a\}$ and so by induction hypothesis π' is a finite composition of transpositions of elements satisfying (1.45) for π' and hence also for π in view of (1.46). Hence so is π, because

$$\begin{array}{cl} & \pi \\ = & \{\text{transpositions are idempotent!}\} \\ & \pi \circ (a_0\ \pi^{-1} a_0) \circ (a_0\ \pi^{-1} a_0) \\ = & \{\text{definition of } \pi'\} \\ & \pi' \circ (a_0\ \pi^{-1} a_0). \end{array}$$

and this completes the induction step. □

In view of the theorem, an action of Perm A on a set X is completely determined by the *swapping operation*

$$\begin{aligned} &swap : A \times A \times X \to X \\ &swap \triangleq \lambda(a, a', x) \in \mathbb{A} \times \mathbb{A} \times X \to (a\ a') \cdot x. \end{aligned} \tag{1.47}$$

This is because every $\pi \in \operatorname{Perm}\mathbb{A}$ is equal to a composition of transpositions

$$\pi = (a_1\ a'_1) \circ (a_2\ a'_2) \circ \cdots \circ (a_n\ a'_n) \tag{1.48}$$

and hence

$$\pi \cdot x = swap(a_1, a'_1, swap(a_2, a'_2, \ldots swap(a_n, a'_n, x) \cdots)). \tag{1.49}$$

Proposition 1.15 *For each* Perm $\mathbb{A}$*-set X, the function swap* : $A \times A \times X \to X$ *defined in* (1.47) *is equivariant (regarding* $\mathbb{A}$ *as a* Perm $\mathbb{A}$*-set as in Example 1.3).*

Proof Note that $\pi \circ (a\ a') = (\pi\, a\ \pi\, a') \circ \pi$, for any $\pi \in \operatorname{Perm}\mathbb{A}$ and $a, a' \in \mathbb{A}$. Therefore $\pi \cdot swap(a, a', x) = \pi \cdot ((a\ a') \cdot x) = (\pi\, a\ \pi\, a') \cdot (\pi \cdot x) = swap(\pi \cdot a, \pi \cdot a', \pi \cdot x)$. □

Exercises

1.1 If G is a group and X a G-set, show that $\lambda g \in G \to (\lambda x \in X \to g \cdot x)$ is a homomorphism of groups from G to the group $\mathrm{S}\,X$ of permutations of X. Show conversely that if $\theta : G \to \mathrm{S}\,X$ is a homomorphism, then $\lambda(g, x) \in G \times X \to \theta\, g\, x$ is a G-action on X.

1.2 Show that the function mapping a G-set Y to the set $\Gamma\, Y$ defined in (1.15) extends to a functor $[G, \mathbf{Set}] \to \mathbf{Set}$. Writing $\Delta\, X$ for the discrete G-set on a set X, show that Δ extends to a functor $\mathbf{Set} \to [G, \mathbf{Set}]$ that is left adjoint to Γ. In other words there is a bijection $[G, \mathbf{Set}](\Delta\, X, Y) \cong \mathbf{Set}(X, \Gamma\, Y)$ which is natural in X and Y.

1.3 If G is a group, show that its multiplication $(g, g') \mapsto g\, g'$ is an action of G on itself. If G' denotes the resulting G-set, show that there is no equivariant function $G \to G'$ (where G denotes the G-set from Example 1.6) unless $G = \{e\}$ is a trivial group.

1.4 Show that definition (1.25) has the properties (1.8) and (1.9) required of a group action.

1.5 Let X and Y be G-sets. Show that a function F from X to Y is equivariant (1.10) if and only if it satisfies $g \cdot F = F$ for all $g \in G$, with $g \cdot F$ as defined in (1.25).

1.6 Show that currying (1.27) is equivariant: $g \cdot \operatorname{curry} F' = \operatorname{curry}(g \cdot F')$.

1.7 Show that an equivariant function $F : X \to Y$ is an isomorphism in $[G, \mathbf{Set}]$ if and only if it is a bijection. Show that it is a monomorphism if and only if it is an injective function. [Hint: consider the pullback of F along itself.] Deduce that subobjects of a G-set are in bijection with equivariant subsets of X.

1.8 Let $\mathrm{P}_{\mathrm{ne}}\, X$ denote the set of non-empty subsets of a set X. Note that if X is a G-set, then the G-action (1.31) restricts to $\mathrm{P}_{\mathrm{ne}}\, X$ and makes it a G-set. Now let X be the G-set G' from Exercise 1.3 and suppose that G is non-trivial (that is, contains some element not equal to the group unit). Show that there is no equivariant function $c : \mathrm{P}_{\mathrm{ne}}\, X \to X$ satisfying $(\forall S \in \mathrm{P}_{\mathrm{ne}}\, X)\ c\,S \in S$. [Hint: consider the action of some $g \neq e$ on $c\,S$ when S is the whole of G.]

1.9 Let $\sim$, F and $\overline{F}$ be as in section 1.7. Instead of appealing to the Equivariance Principle, show by explicit calculation that if $\sim$ and F are equivariant, then $\overline{F}$ satisfies $g \cdot (\overline{F}[x]_{\sim}) = \overline{F}(g \cdot [x]_{\sim})$ for all $g \in G$ and $x \in X$.

2

Support

This chapter introduces the central concept of the theory of nominal sets, namely the *support* of an element in a set equipped with a permutation action. From now on $\mathbb{A}$ denotes a fixed, countably infinite set whose elements $a, b, c, \ldots$ we call *atomic names*.

2.1 The category of nominal sets

Let X be a set equipped with an action of the group $\mathrm{Perm}\,\mathbb{A}$ of finitary permutations of $\mathbb{A}$. A set of atomic names $A \subseteq \mathbb{A}$ is a *support* for an element $x \in X$ if for all $\pi \in \mathrm{Perm}\,\mathbb{A}$

$$((\forall a \in A)\ \pi\, a = a) \Rightarrow \pi \cdot x = x. \tag{2.1}$$

The following characterization of support in terms of transpositions is helpful.

Proposition 2.1 *Suppose X is an* $\mathrm{Perm}\,\mathbb{A}$*-set and $x \in X$. A subset $A \subseteq \mathbb{A}$ supports x if and only if*

$$(\forall a_1, a_2 \in \mathbb{A} - A)\ (a_1\ a_2) \cdot x = x. \tag{2.2}$$

Notation We denote *set subtraction* by $X - Y$. Thus $\mathbb{A} - A = \{a \in \mathbb{A} \mid a \notin A\}$.

Proof If $a_1, a_2 \in \mathbb{A} - A$, then $(a_1\ a_2)\, a = a$ for any $a \in A$. So if A supports x, it clearly satisfies (2.2). Conversely, suppose A satisfies (2.2) and that $\pi \in \mathrm{Perm}\,\mathbb{A}$ fixes each element of A. We have to show that $\pi \cdot x = x$. Recall from Theorem 1.14 that π can be written as a composition of transpositions $(a_1\ a_2)$ satisfying $\pi\, a_1 \neq a_1 \neq a_2 \neq \pi\, a_2$. Since π fixes each element of A, such a transposition satisfies $a_1, a_2 \notin A$ and hence by (2.2), $(a_1\ a_2) \cdot x = x$. Therefore, letting each transposition in the sequence whose composition is π act on x in turn, we eventually conclude that $\pi \cdot x = x$, as required. □

Clearly each element of an Perm $\mathbb{A}$-set is supported by the whole of $\mathbb{A}$, which is an infinite set. We will be interested in elements that are *finitely supported* in the sense that there is some finite set of atomic names that is a support for the element.

Definition 2.2 A *nominal set* is an Perm $\mathbb{A}$-set all of whose elements are finitely supported. Nominal sets are the objects of a category **Nom** whose morphisms, composition and identities are as in the category of Perm $\mathbb{A}$-sets [Perm $\mathbb{A}$, **Set**]. Thus **Nom** is a full subcategory of [Perm $\mathbb{A}$, **Set**]. The dependence of **Nom** upon $\mathbb{A}$ is left implicit.

Proposition 2.3 *If A_1 and A_2 are finite supports for an element x of an* Perm $\mathbb{A}$-*set, then so is $A_1 \cap A_2$.*

Proof By Proposition 2.1 we just have to show that if $a, a' \in \mathbb{A} - (A_1 \cap A_2)$, then $(a\ a') \cdot x = x$. The latter certainly holds if $a = a'$, so we may suppose $a \neq a'$. In that case, picking any element a'' of the infinite set $\mathbb{A} - (A_1 \cup A_2 \cup \{a, a'\})$, then

$$(a\ a') = (a\ a'') \circ (a'\ a'') \circ (a\ a''). \tag{2.3}$$

Note that either $a \notin A_1$, or $a \notin A_2$; and similarly for a'. So each of the transpositions on the right-hand side of (2.3) swaps atomic names that are either both not in the support set A_1, or both not in the support set A_2 (since a'' is definitely not in either of A_1 or A_2). So $(a\ a')$ is a composition of transpositions each of which fixes x and hence itself fixes x. □

Suppose X is a nominal set and $x \in X$. Since x possesses some finite support, by the proposition

$$\mathrm{supp}_X\, x \triangleq \bigcap\{A \in \mathrm{P}\,\mathbb{A} \mid A \text{ is a finite support for } x\} \tag{2.4}$$

is again a finite support for x and is the least one with respect to subset inclusion. We write it as $\mathrm{supp}\, x$ when X is clear from the context. Exercise 2.1 gives an equivalent formulation of $\mathrm{supp}\, x$ from Gabbay and Pitts (2002).

Example 2.4 The *nominal set of atomic names* is given by $\mathbb{A}$ regarded as an Perm $\mathbb{A}$-set as in Example 1.3. Clearly each $a \in \mathbb{A}$ is finitely supported by $\{a\}$ and this is the smallest support, so that $\mathrm{supp}\, a = \{a\}$.

Example 2.5 The *discrete nominal set* on a set X is given by the Perm $\mathbb{A}$-action of Example 1.5, for which we have $\mathrm{supp}\, x = \emptyset$ for each $x \in X$.

Example 2.6 We saw in Example 1.4 that the set $\Sigma(\mathbb{A})$ of terms over an algebraic signature Σ with variables from $\mathbb{A}$ is a Perm $\mathbb{A}$-set. It is in fact a nominal set with $\mathrm{supp}\, t$ equal to the finite set $\mathrm{var}\, t$ of variables occurring in t. This set is recursively defined by:

$$\begin{aligned} \operatorname{var} a &= \{a\} \\ \operatorname{var} \mathsf{op}(t_1\ ,\ \cdots\ ,\ t_n) &= \operatorname{var} t_1 \cup \cdots \cup \operatorname{var} t_n. \end{aligned}$$

That $\operatorname{var} t$ is the least support for t follows from the fact that it is a strong support in the sense of the following theorem due to Tzevelekos (2008, section 2.1.2). Property (2.5) for $A = \operatorname{var} t$ and $x = t$ can be proved by induction on the structure of the term t.

Theorem 2.7 *By definition, a set of atomic names $A \subseteq \mathbb{A}$* strongly supports *an element x of a nominal set X if and only if*

$$((\forall a \in A)\ \pi\, a = a) \Leftrightarrow \pi \cdot x = x. \tag{2.5}$$

In this case if A is finite, then $A = \operatorname{supp} x$.

Proof If A strongly supports x, then it certainly supports x and hence $\operatorname{supp} x \subseteq A$. If A is finite, then we also have $A \subseteq \operatorname{supp} x$. For then $\mathbb{A} - A$ is non-empty, say $a' \in \mathbb{A} - A$, and for any $a \in A - \operatorname{supp} x$, $(a\ a')\, a = a' \neq a \in A$. So from (2.5) we conclude that $(a\ a') \cdot x \neq x$; but this contradicts the fact that $a, a' \notin \operatorname{supp} x$. □

Since $\mathbb{A}$ is a $\operatorname{Perm}\mathbb{A}$-set, so is its powerset $\mathrm{P}\,\mathbb{A}$, as in section 1.5. It is not hard to see that it contains elements that are not finitely supported (Exercise 2.2); but as the next result shows, we can identify exactly which subsets of $\mathbb{A}$ are finitely supported.

Proposition 2.8 *A set $A \subseteq \mathbb{A}$ of atomic names is a finitely supported element of* $\mathrm{P}\,\mathbb{A}$ *if and only if either it or its complement $\mathbb{A} - A$ is finite. In the latter case one says that A is* cofinite.

Proof Note that $(a\ a')\cdot A = A$ if and only if $(a \in A \wedge a' \in A) \vee (a \notin A \wedge a' \notin A)$. So from Proposition 2.1 it follows that $A' \subseteq \mathbb{A}$ supports $A \in \mathrm{P}\,\mathbb{A}$ if and only if either $A \subseteq A'$, or $\mathbb{A} - A \subseteq A'$. Hence the result. □

Example 2.9 Each finite element of $\mathrm{P}\,\mathbb{A}$ is supported by itself and this is the least support. So we get a nominal set $\mathrm{P_f}\,\mathbb{A}$ of finite sets $\overline{a}$ of atomic names with $\pi \cdot \overline{a} = \{\pi\, a \mid a \in \overline{a}\}$ and $\operatorname{supp} \overline{a} = \overline{a}$. Note that $\overline{a}$ is not a strong support for $\overline{a}$ in the sense of Theorem 2.7, so long as it has at least two elements.

Proposition 2.10 *For each nominal set X the support function* $\operatorname{supp} : X \to \mathrm{P_f}\,\mathbb{A}$ *is equivariant:*

$$\pi \cdot (\operatorname{supp} x) = \operatorname{supp}(\pi \cdot x) \tag{2.6}$$

holds for all $\pi \in \operatorname{Perm}\mathbb{A}$ and $x \in X$.

Proof In view of (2.1) and (2.4), the support function is definable within classical higher-order logic from the action function $\lambda(\pi, x) \in \mathrm{Perm}\,\mathbb{A} \times X \to \pi \cdot x$; and we saw in Example 1.6 that the latter is equivariant. Therefore the theorem follows from the Equivariance Principle. □

We will revisit the various constructions on G-sets from chapter 1 and see whether they preserve the property of being a nominal set. The following results about support and equivariant functions will be useful for doing this.

Lemma 2.11 *Suppose that $f : X \to Y$ is an equivariant function between* $\mathrm{Perm}\,\mathbb{A}$*-sets.*

1. *If A is a support for $x \in X$, then it is a support for $f\,x \in Y$. So if X and Y are nominal sets, then*
$$(\forall x \in X)\ \mathrm{supp}_Y(f\,x) \subseteq \mathrm{supp}_X\,x. \tag{2.7}$$
2. *If f is injective ($(\forall x, x' \in X)\ f\,x = f\,x' \Rightarrow x = x'$), then A is a support for $x \in X$ if and only if it is a support for $f\,x \in Y$. So if Y is a nominal set, then so is X and*
$$(\forall x \in X)\ \mathrm{supp}_X\,x = \mathrm{supp}_Y(f\,x). \tag{2.8}$$
3. *If f is surjective ($(\forall y \in Y)(\exists x \in X)\ f\,x = y$) and X is a nominal set, then so is Y.*

Proof For part 1, suppose A supports $x \in X$. If $\pi \in \mathrm{Perm}\,\mathbb{A}$ satisfies $(\forall a \in A)\ \pi\,a = a$, then by (2.1) $x = \pi \cdot x$ and hence $f\,x = f(\pi \cdot x) = \pi \cdot (f\,x)$ since f is equivariant. Thus A supports $f\,x$ in Y. Therefore if X and Y are nominal sets and $x \in X$, then $\mathrm{supp}_X\,x$ supports $f\,x$ and hence contains the smallest support, $\mathrm{supp}_Y(f\,x)$.

For part 2, suppose f is injective and that A supports $f\,x$. So if $\pi \in \mathrm{Perm}\,\mathbb{A}$ satisfies $(\forall a \in A)\ \pi\,a = a$, then by (2.1) $f\,x = \pi \cdot (f\,x)$ and hence $f\,x = f(\pi \cdot x)$ since f is equivariant; as it is also injective this implies $x = \pi \cdot x$. Therefore A is a support for x. So if Y is nominal, each $x \in X$ possesses a finite support, namely $\mathrm{supp}_Y(f\,x)$; hence X is nominal. In this case $\mathrm{supp}_Y(f\,x)$ contains the smallest support for x, $\mathrm{supp}_X\,x$; and conversely, $\mathrm{supp}_Y(f\,x)$ is contained in $\mathrm{supp}_X\,x$ by part 1.

Finally, part 3 follows immediately from part 1, since each $y \in Y$ possesses a finite support, namely $\mathrm{supp}_X\,x$ where x is some element of X such that $f\,x = y$. □

2.2 Products and coproducts

Coproducts in **Nom** are constructed just as in $[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$, by taking disjoint unions (see section 1.2). This is because, for any family of nominal sets $(X_i \mid i \in I)$, each element (i, x) of $\sum_{i \in I} X_i$ is supported by $\mathrm{supp}_{X_i}\, x$. For if π fixes each $a \in \mathrm{supp}_{X_i}\, x$, then $\pi \cdot x = x$ and hence $\pi \cdot (i, x) = (i, \pi \cdot x) = (i, x)$. Indeed, since $\mathrm{inj}_i = \lambda x \in X_i \to (i, x)$ is an injective equivariant function from X_i to $\sum_{i \in I} X_i$, by (2.8) we have

$$\mathrm{supp}_{\sum_{i \in I} X_i}\, (i, x) = \mathrm{supp}_{X_i}\, x. \tag{2.9}$$

The situation for products is more complicated. Finite cartesian products of nominal sets are again nominal, but infinite products in **Nom** are not given in general by the product in $[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$.

Proposition 2.12 *The product $X_1 \times \cdots \times X_n$ in $[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$ of finitely many nominal sets is another such (and hence is their product in* **Nom***). For each $(x_1, \ldots, x_n) \in X_1 \times \cdots \times X_n$*

$$\mathrm{supp}_{X_1 \times \cdots \times X_n}(x_1, \ldots, x_n) = \mathrm{supp}_{X_1}\, x_1 \cup \cdots \cup \mathrm{supp}_{X_n}\, x_n. \tag{2.10}$$

Proof Recall from section 1.2 that the permutation action for products is given componentwise: $\pi \cdot (x_1, \ldots, x_n) = (\pi \cdot x_1, \ldots, \pi \cdot x_n)$. So if π fixes each atomic name in $\mathrm{supp}_{X_1}\, x_1 \cup \cdots \cup \mathrm{supp}_{X_n}\, x_n$, then it also fixes $(x_1, \ldots, x_n)$. Therefore each $(x_1, \ldots, x_n)$ is finitely supported by $\mathrm{supp}_{X_1}\, x_1 \cup \cdots \cup \mathrm{supp}_{X_n}\, x_n$ and $X_1 \times \cdots \times X_n$ is a nominal set. To prove (2.10) it just remains to show that $\mathrm{supp}_{X_1}\, x_1 \cup \cdots \cup \mathrm{supp}_{X_n}\, x_n$ is contained in $\mathrm{supp}(x_1, \ldots, x_n)$; but since each projection function $\mathrm{proj}_i : X_1 \times \cdots \times X_n \to X_i$ is equivariant, by (2.7) we have $\mathrm{supp}_{X_i}\, x_i = \mathrm{supp}_{X_i}(\mathrm{proj}_i(x_1, \ldots, x_n)) \subseteq \mathrm{supp}(x_1, \ldots, x_n)$ for each $i \in I$. □

Generalizing from finite to infinite cartesian products $\prod_{i \in I} X_i$, as in the above proof we have that an element $(x_i \mid i \in I)$ is supported by $\bigcup_{i \in I} \mathrm{supp}_{X_i}\, x_i$, but that set may not be finite (Exercise 2.3). Nevertheless, the category **Nom** does have infinite products, because of the following result.

Theorem 2.13 *For each $\mathrm{Perm}\,\mathbb{A}$-set X there is an equivariant function $X_{\mathrm{fs}} \to X$ from a nominal set X_{fs} to X that is universal among all such. In other words, given any equivariant function F from a nominal set Y to X*

$$\begin{tikzcd}
Y \arrow[d, dashed, "\overline{F}"'] \arrow[dr, "F"] & \\
X_{\mathrm{fs}} \arrow[r] & X
\end{tikzcd}$$

there is a unique equivariant function $\overline{F}$ making the above diagram commute. (Thus **Nom** *is a co-reflective subcategory of* $[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$*.)*

Proof First note that if $x \in X$ is supported by $A \in \mathrm{P}\,\mathbb{A}$, then for any $\pi \in \mathrm{Perm}\,\mathbb{A}$,

$\pi \cdot x$ is supported by $\pi \cdot A$. (This is a consequence of the Equivariance Principle mentioned in section 1.5; or one can prove it directly from the definition of support.) So

$$X_{\mathrm{fs}} \triangleq \{x \in X \mid x \text{ is finitely supported in } X\}. \tag{2.11}$$

is an equivariant subset of X and hence an Perm $\mathbb{A}$-set with permutation action inherited from X. The inclusion $X_{\mathrm{fs}} \subseteq X$ gives an injective equivariant function and therefore X_{fs} is a nominal set by (2.8). To see that it has the required universal property it suffices to show that any $F : Y \to X$ in [Perm $\mathbb{A}$, **Set**] with $Y \in$ **Nom** maps elements of Y into the subset $X_{\mathrm{fs}} \subseteq X$; but from part 1 of Lemma 2.11, for each $y \in Y$ we have that $\mathrm{supp}_Y\, y$ is a finite support for $F\, y$ and hence $F\, y \in X_{\mathrm{fs}}$. □

Combining the description of products in [Perm $\mathbb{A}$, **Set**] from section 1.2 with the above theorem, we get:

Corollary 2.14 *Given a family of nominal sets X_i indexed by the elements i of some set I, their product in* **Nom** *is given by $(\prod_{i \in I} X_i)_{\mathrm{fs}}$ with product projections*

$$(\textstyle\prod_{i \in I} X_i)_{\mathrm{fs}} \subseteq \prod_{i \in I} X_i \xrightarrow{\mathrm{proj}_i} X_i.$$

Thus infinite products in **Nom** may contain rather few elements; see Exercise 2.4.

2.3 Natural numbers

We saw in section 1.3 that the natural numbers object in the category of G-sets is given by $\mathbb{N}$ with the discrete G-action together with the usual zero and successor functions. As we noted in Example 2.5, every discrete Perm $\mathbb{A}$-set is a nominal set. It follows that

$$1 \xrightarrow{\mathrm{zero}} \mathbb{N} \xrightarrow{\mathrm{suc}} \mathbb{N}$$

is also the natural numbers object in the category **Nom** of nominal sets and equivariant functions.

2.4 Functions

Recall from section 1.4 the definition of the function G-set Y^X. When $G = \mathrm{Perm}\,\mathbb{A}$, the following characterization of the support of elements of Y^X is useful.

Lemma 2.15 *Given $X, Y \in$ [Perm $\mathbb{A}$, **Set**], a set of atomic names $A \subseteq \mathbb{A}$ supports $F \in Y^X$ if and only if for all $\pi \in$ Perm $\mathbb{A}$*

$$((\forall a \in A)\ \pi\, a = a) \Rightarrow (\forall x \in X)\ F(\pi \cdot x) = \pi \cdot (F\, x). \tag{2.12}$$

In particular, F has empty support if and only if it is an equivariant function (Definition 1.2).

Proof It follows from the definition of the action of permutations on functions (1.25) that $\pi \cdot F = F$ holds if and only if $(\forall x \in X)\ F(\pi \cdot x) = \pi \cdot (F\,x)$. □

Definition 2.16 If X and Y are nominal sets, we write $X \to_{\mathrm{fs}} Y$ for the nominal set $(Y^X)_{\mathrm{fs}}$ formed from Y^X as in (2.11), and call it the *nominal function set* of X and Y.

Restricting the application function $\mathrm{app} : Y^X \times X \to Y$ to finitely supported functions, we get an equivariant function

$$\begin{aligned}&\mathrm{app} : (X \to_{\mathrm{fs}} Y) \times X \to Y\\&\mathrm{app} \triangleq \lambda(F, x) \in (X \to_{\mathrm{fs}} Y) \times X \to F\,x.\end{aligned} \tag{2.13}$$

Currying an equivariant function $F : Z \times X \to Y$ to get $\mathrm{curry}\,F : Z \to Y^X$ as in section 1.4, by Theorem 2.13 if Z is nominal then $\mathrm{curry}\,F$ factors through $(Y^X)_{\mathrm{fs}} \subseteq Y^X$ to give an equivariant function

$$\begin{aligned}&\mathrm{curry}\,F : Z \to (X \to_{\mathrm{fs}} Y)\\&\mathrm{curry}\,F \triangleq \lambda z \in Z, x \in X \to F(z, x).\end{aligned} \tag{2.14}$$

Theorem 2.17 **Nom** *is a cartesian closed category.*

Proof We have already seen that **Nom** has finite products. Furthermore $(Y^X)_{\mathrm{fs}}$ gives the exponential of Y by X in **Nom** for general category-theoretic reasons: **Nom** is a co-reflective full subcategory of $[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$ (Theorem 2.13) and the inclusion of **Nom** into $[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$ preserves finite products. Thus there are bijections of hom-sets

$$\begin{aligned}&\mathbf{Nom}(Z \times X, Y)\\=\;&\{\text{binary products in } \mathbf{Nom} \text{ are as in } [\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]\}\\&[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}](Z \times X, Y)\\\cong\;&\{Y^X \text{ is the exponential in } [\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]\}\\&[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}](Z, Y^X)\\\cong\;&\{\text{Theorem 2.13}\}\\&\mathbf{Nom}(Z, (Y^X)_{\mathrm{fs}})\\=\;&\{\text{definition of } X \to_{\mathrm{fs}} Y\}\\&\mathbf{Nom}(Z, X \to_{\mathrm{fs}} Y)\end{aligned}$$

(natural in $X, Y, Z \in \mathbf{Nom}$) given by sending $F \in \mathbf{Nom}(Z \times X, Y)$ to $\mathrm{curry}\,F \in \mathbf{Nom}(Z, X \to_{\mathrm{fs}} Y)$. □

Not every function between nominal sets is finitely supported; in other words $X \to_{\mathrm{fs}} Y$ can be a proper subset of Y^X. Here is an example that shows this.

Example 2.18 Consider the function Perm $\mathbb{A}$-set Y^X where X is the set $\mathbb{N}$ of natural numbers regarded as a discrete nominal set (Example 2.5) and where Y is the nominal set $\mathrm{P_f}\,\mathbb{A}$ of finite sets of atoms, as in Example 2.9. Let $F \in Y^X$ be a function that maps each natural number $n \in \mathbb{N}$ to some finite set of atoms of cardinality n. Then there is no finite set of atoms A that supports F in Y^X. To see this we suppose A is a finite support for F and derive a contradiction. Since each $n \in \mathbb{N}$ has empty support, A is also a support for $(F, n) \in Y^X \times X$; and then since application $\mathrm{app} : Y^X \times X \to Y$ is equivariant, by part 1 of Lemma 2.11 we have that A is a support for $F\,n \in \mathrm{P_f}\,\mathbb{A}$. We noted in Example 2.9 that the least support of a finite set of atoms is the set itself. Therefore we have proved $(\forall n \in \mathbb{N})\ F\,n \subseteq A$. Taking n larger than the cardinality of the finite set A gives a contradiction, since by assumption on F, $F\,n$ is a set of cardinality n and so cannot be a subset of A.

2.5 Power sets

Lemma 2.19 *If $S \subseteq X$ is an equivariant subset of a nominal set X, then restricting the* Perm $\mathbb{A}$*-action on X to S, S is a nominal set.*

Proof By part 2 of Lemma 2.11 applied to the inclusion $S \subseteq X$, the elements of S are finitely supported because they are so in X. □

The subobjects of $X \in \mathbf{Nom}$ correspond to equivariant subsets of X. To see this we need to describe pullbacks in **Nom**. A *pullback* for a pair of morphisms F_1 and F_2 with common codomain in a category is a commutative square

$$\begin{tikzcd}
P \arrow[r, "P_2"] \arrow[d, "P_1"'] & X_2 \arrow[d, "F_2"] \\
X_1 \arrow[r, "F_1"'] & Y
\end{tikzcd}$$

with the universal property that given any $G_i : Y \to X_i$ with $F_1 \circ G_1 = F_2 \circ G_2$, there is a unique morphism from Y to P making the two triangles commute:

$$\begin{tikzcd}
Y \arrow[dr, dashed] \arrow[drr, bend left, "G_2"] \arrow[ddr, bend right, "G_1"'] & & \\
& P \arrow[r, "P_2"'] \arrow[d, "P_1"] & X_2 \arrow[d, "F_2"] \\
& X_1 \arrow[r, "F_1"'] & Y.
\end{tikzcd}$$

When the category is **Nom**, as for binary products, pullbacks are created by taking pullbacks of the underlying functions:

$$\begin{aligned}P &\triangleq \{(x_1, x_2) \in X_1 \times X_2 \mid F_1\, x_1 = F_2\, x_2\}\\ P_i &= \lambda(x_1, x_2) \in P \to x_i\end{aligned}$$

(Note that P is an equivariant subset of the product $X_1 \times X_2$ and hence by part 2 of Lemma 2.11 is a nominal set.) By considering the pullback of a monomorphism against itself, it follows that a morphism $F : X \to Y$ in **Nom** is a monomorphism if and only if F is an injective function. Combining this observation with part 2 of Lemma 2.11 we conclude that the subobjects of X in **Nom** are the same as its subobjects in $[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$ and correspond to equivariant subsets of X.

Theorem 2.20 *The category* **Nom** *of nominal sets and equivariant functions is a Boolean topos with a natural number object.*

Proof We have already seen that **Nom** is cartesian closed and possesses a natural number object. Just as in $[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$, the discrete nominal set $\mathbb{B} = \{\mathrm{true}, \mathrm{false}\}$ is a subobject classifier, because of the above observations about pullbacks and subobjects in **Nom**. □

As in any topos, the exponential $X \to_{\mathrm{fs}} \mathbb{B} = (\mathbb{B}^X)_{\mathrm{fs}}$ of the subobject classifier by an object X gives a form of powerset object in **Nom**. The isomorphism in $[\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$ between $\mathbb{B}^X$ and $\mathrm{P}\,X$ (section 1.5) restricts to an isomorphism between the finitely supported elements of $\mathbb{B}^X$ and $\mathrm{P}\,X$. However, as the next result shows, to check that $S \subseteq X$ is a finitely supported element of $\mathrm{P}\,X$ it suffices to check that containments $\pi \cdot S \subseteq S$ rather than equalities $\pi \cdot S = S$ hold for all permutations π fixing the atomic names in a support set.

Lemma 2.21 *Given* $X \in [\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$*, a set of atomic names* $A \subseteq \mathbb{A}$ *supports* $S \in \mathrm{P}\,X$ *if and only if*

$$(\forall \pi \in \mathrm{Perm}\,\mathbb{A})((\forall a \in A)\ \pi\, a = a) \Rightarrow (\forall x \in S)\ \pi \cdot x \in S. \qquad (2.15)$$

In particular, S *has empty support if and only if it is an equivariant subset of* X *(Definition 1.8).*

Proof Note that by (1.32), $(\forall x \in S)\ \pi \cdot x \in S$ is equivalent to $\pi \cdot S \subseteq S$. Note as well that any π satisfying $(\forall a \in A)\ \pi\, a = a)$ also satisfies $(\forall a \in A)\ \pi^{-1} a = a)$. So if (2.15) holds, then we have $\pi \cdot S \subseteq S$ and $\pi^{-1} \cdot S \subseteq S$ and hence $\pi \cdot S = S$. So (2.15) implies that A supports S in $\mathrm{P}\,X$; and the converse is immediate. □

The following simple corollary of this lemma will be useful.

Proposition 2.22 *Given* $X, Y \in [\mathrm{Perm}\,\mathbb{A}, \mathbf{Set}]$*, if* $A \subseteq \mathbb{A}$ *supports both* $S \in \mathrm{P}\,(X \times Y)$ *and* $x \in X$*, then it also supports* $\{y \in Y \mid (x, y) \in S\} \in \mathrm{P}\,Y$.

Proof Suppose $\pi \in \mathrm{Perm}\,\mathbb{A}$ satisfies $(\forall a \in A)\ \pi\, a = a$. Then $\pi{\cdot}x = x$ and $\pi{\cdot}S = S$. So for all y in $\{y \in Y \mid (x,y) \in S\}$ we have $(x, \pi{\cdot}y) = (\pi{\cdot}x, \pi{\cdot}y) = \pi{\cdot}(x,y) \in \pi{\cdot}S = S$ and therefore $\pi \cdot y$ is also in $\{y \in Y \mid (x,y) \in S\}$. So we can apply Lemma 2.21 to deduce that A supports $\{y \in Y \mid (x,y) \in S\}$. □

Definition 2.23 If X is a nominal set, we write $\mathrm{P}_{\mathrm{fs}}\, X$ for the nominal set $(\mathrm{P}\,X)_{\mathrm{fs}}$ formed from the power $\mathrm{Perm}\,\mathbb{A}$-set $\mathrm{P}\,X$ using (2.11). We call $\mathrm{P}_{\mathrm{fs}}\, X$ the *nominal powerset* of X.

For any nominal set X, from parts 1–3 of Proposition 1.9 we have the following equivariant and hence finitely supported subsets.

- *Truth* $X \in \mathrm{P}_{\mathrm{fs}}\, X$.
- *Equality* $\{(x, x') \in X \times X \mid x = x'\} \in \mathrm{P}_{\mathrm{fs}}\,(X \times X)$.
- *Membership* $\{(x, S) \in X \times \mathrm{P}_{\mathrm{fs}}\, X \mid x \in S\} \in \mathrm{P}_{\mathrm{fs}}\,(X \times \mathrm{P}_{\mathrm{fs}}\, X)$.

Furthermore, if $f : \mathrm{P}\,X_1 \times \cdots \times \mathrm{P}\,X_n \to \mathrm{P}\,X$ is an equivariant function and $(S_1, \ldots, S_n) \in \mathrm{P}_{\mathrm{fs}}\, X_1 \times \cdots \times \mathrm{P}_{\mathrm{fs}}\, X_n$, then by part 1 of Lemma 2.11 and (2.10), $f(S_1, \ldots, S_n)$ is supported by the union of the supports of each S_i and hence is in $\mathrm{P}_{\mathrm{fs}}\, X$. So parts 4–7 of Proposition 1.9 give us the following equivariant functions.

- *Conjunction* $\_ \cap \_ : \mathrm{P}_{\mathrm{fs}}\, X \times \mathrm{P}_{\mathrm{fs}}\, X \to \mathrm{P}_{\mathrm{fs}}\, X$.
- *Negation* $\neg : \mathrm{P}_{\mathrm{fs}}\, X \to \mathrm{P}_{\mathrm{fs}}\, X$, where $\neg S \triangleq \{x \in X \mid x \notin S\}$.
- *Universal quantification* $\bigcap : \mathrm{P}_{\mathrm{fs}}\,(\mathrm{P}_{\mathrm{fs}}\, X) \to \mathrm{P}_{\mathrm{fs}}\, X$, where $\bigcap \mathcal{S} \triangleq \{x \in X \mid (\forall S \in \mathcal{S})\ x \in S\}$.
- *Substitution* $f^* : \mathrm{P}_{\mathrm{fs}}\, Y \to \mathrm{P}_{\mathrm{fs}}\, X$, where $f : X \to Y$ is an equivariant function and $f^* S \triangleq \{x \in X \mid f\, x \in S\}$.

Finally, the analogue of part 8 of Proposition 1.9 is the following equivariant function.

- *Comprehension* $\mathrm{compr} : \mathrm{P}_{\mathrm{fs}}\,(X \times Y) \to (X \to_{\mathrm{fs}} \mathrm{P}_{\mathrm{fs}}\, Y)$, where $\mathrm{compr}\, S \triangleq \lambda x \in X \to \{y \in Y \mid (x,y) \in S\}$. To see that this is well-defined, note that given $S \in \mathrm{P}_{\mathrm{fs}}\,(X \times Y)$ and $x \in X$, by Proposition 2.22 $\mathrm{compr}\, S\ x = \{y \in Y \mid (x,y) \in S\}$ is finitely supported by $\mathrm{supp}\, S \cup \mathrm{supp}\, x$. Hence $\mathrm{compr}\, S \in (\mathrm{P}_{\mathrm{fs}}\, Y)^X$; but since we know from Proposition 1.9 that compr is equivariant, by part 1 of Lemma 2.11 we have $\mathrm{compr}\, S \in ((\mathrm{P}_{\mathrm{fs}}\, Y)^X)_{\mathrm{fs}} = X \to_{\mathrm{fs}} \mathrm{P}_{\mathrm{fs}}\, Y$.

From these facts we get a version for nominal sets of the Equivariance Principle from section 1.5:

Finite Support Principle *Any function or relation that is defined from finitely supported functions and subsets using classical higher-order logic is itself finitely supported, provided we restrict any quantification over functions or subsets to range over ones that are finitely supported.*

2.6 Failure of choice

Although nominal sets provide a model of classical higher-order logic, as the following theorem shows, they do not model choice principles. Indeed Fraenkel and Mostowski introduced permutations and finite supports into logic in the first place in order to construct a model of set theory with atoms not satisfying the Axiom of Choice.

Theorem 2.24 *Let* $\mathrm{P}_{\mathrm{nefs}}\mathbb{A}$ *be the nominal set of non-empty, finitely supported subsets of* $\mathbb{A}$ *(where the* $\mathrm{Perm}\,\mathbb{A}$*-action on such subsets is as for* $\mathrm{P}_{\mathrm{fs}}\,\mathbb{A}$*). No function* $C : \mathrm{P}_{\mathrm{nefs}}\mathbb{A} \to \mathbb{A}$ *satisfying*

$$(\forall S \in \mathrm{P}_{\mathrm{nefs}}\mathbb{A})\; C\,S \in S \tag{2.16}$$

can have finite support in the $\mathrm{Perm}\,\mathbb{A}$*-set* $(\mathrm{P}_{\mathrm{nefs}}\mathbb{A})^{\mathbb{A}}$.

Proof We suppose C satisfying (2.16) is supported by a finite subset $A \subseteq \mathbb{A}$ and derive a contradiction. Let S be the cofinite set $\mathbb{A} - A$. From Proposition 2.8 we have that $S \in \mathrm{P}_{\mathrm{fs}}\,\mathbb{A}$; and being cofinite, S is in particular non-empty. Therefore we can apply C to S to get an atomic name $a_0 \triangleq C\,S \in S = \mathbb{A} - A$. Since $A \cup \{a_0\}$ is finite and $\mathbb{A}$ is infinite, there is some $a_1 \in \mathbb{A} - (A \cup \{a_0\}) \subseteq S$. Since $a_0, a_1 \in S$, we have $(a_0\ a_1) \cdot S = S$; and since A supports C and $a_0, a_1 \notin A$, we also have $(a_0\ a_1) \cdot C = C$. So by definition of the $\mathrm{Perm}\,\mathbb{A}$-action on $(\mathrm{P}_{\mathrm{nefs}}\mathbb{A})^{\mathbb{A}}$, $(a_0\ a_1) \cdot (C\,S) = ((a_0\ a_1) \cdot C)((a_0\ a_1) \cdot S) = C\,S$. Therefore $a_1 = (a_0\ a_1) \cdot a_0 = (a_0\ a_1) \cdot (C\,S) = C\,S = a_0$, contradicting the fact that a_1 was chosen to be distinct from a_0. □

So in particular **Nom** does not model Hilbert's ε-operator mapping non-empty subsets at each type to elements of those subsets. We remarked in Note 1.11 that this is already the case for [Perm $\mathbb{A}$, **Set**] and equivariant properties. However, when we move to **Nom** and finitely supported properties even weaker, internal choice principles fail to hold. For example, the above theorem shows that **Nom** fails to satisfy the higher-order logic formula

$$\begin{aligned}&(\forall R \in \mathrm{P}\,(X \times Y))\\ &\qquad ((\forall x \in X)(\exists y \in Y)\;(x, y) \in R) \Rightarrow (\exists f \in Y^X)(\forall x \in X)\;(x, f\,x) \in R.\end{aligned}$$

(Interpret X as $\mathrm{P}_{\mathrm{nefs}}\mathbb{A}$, Y as $\mathbb{A}$ and R as $\{(S, a) \in \mathrm{P}_{\mathrm{nefs}}\mathbb{A} \times \mathbb{A} \mid a \in S\}$ and apply the theorem.)

Nevertheless, many uses of choice in theorem-provers based on classical higher-order logic (such as the HOL theorem prover, Gordon and Melham, 1993) are for making definitions, where the thing defined is unique. Such restricted uses of choice are consistent with the model of higher-order logic that **Nom** provides (see Pitts, 2003, section 8).

2.7 Partial functions

If X and Y are nominal sets, we saw in section 1.6 that the set $X \rightharpoonup Y$ of partial functions is an equivariant subset of $\mathrm{P}\,(X \times Y)$. Applying $\_{}_{\mathrm{fs}}$ to it we obtain the set of finitely supported partial functions

$$X \rightharpoonup_{\mathrm{fs}} Y \triangleq (X \rightharpoonup Y)_{\mathrm{fs}}. \tag{2.17}$$

Thus the elements of $X \rightharpoonup_{\mathrm{fs}} Y$ are the finitely supported subsets of $X \times Y$ that are single-valued (1.36). This gives the object of partial maps from X to Y in **Nom**; that is, there is a natural correspondence between partial maps (Johnstone, 2002, p 100)

$$\begin{tikzcd}
\cdot \arrow[r] \arrow[d, tail] & Y \\
Z \times X &
\end{tikzcd} \tag{2.18}$$

and morphisms $Z \to (X \rightharpoonup_{\mathrm{fs}} Y)$. Indeed, partial maps in **Nom** correspond to *equivariant partial functions*, that is, partial functions that are equivariant as subsets. Given such an $F \in (Z \times X) \rightharpoonup Y$, the corresponding equivariant function $\mathrm{curry}\, F : Z \to (X \rightharpoonup_{\mathrm{fs}} Y)$ maps each $z \in Z$ to

$$\mathrm{curry}\, F\, z = \{(x, y) \in X \times Y \mid ((z, x), y) \in F\}$$

This is a partial function because F is one; and it is finitely supported by $\mathrm{supp}\, z$, because of Proposition 2.22. The correspondence between F and $\mathrm{curry}\, F$ is mediated by the equivariant partial function

$$\begin{aligned}
&\mathrm{app} \in (X \rightharpoonup_{\mathrm{fs}} Y) \times X \rightharpoonup Y \\
&\mathrm{app} \triangleq \{((F, x), y) \in ((X \rightharpoonup_{\mathrm{fs}} Y) \times X) \times Y \mid (x, y) \in F\}.
\end{aligned} \tag{2.19}$$

that applies a partial function to its argument. For each F, $\mathrm{curry}\, F$ is the unique equivariant function $Z \to (X \rightharpoonup_{\mathrm{fs}} Y)$ satisfying

$$(\forall z \in Z)(\forall x \in X)\ \mathrm{app}(\mathrm{curry}\, F\, z, x) \equiv F(z, x) \tag{2.20}$$

(where $\equiv$ is Kleene equivalence—see section 1.6).

Since **Nom** is a Boolean topos (Theorem 2.20), there is a natural isomorphism between $X \rightharpoonup_{\mathrm{fs}} Y$ and the exponential $X \to_{\mathrm{fs}} (Y + 1)$. This is given by restricting to finitely supported elements the usual correspondence between partial functions to Y and total functions to Y augmented with an element for 'undefined', that is, to $Y + 1$.

2.8 Quotient sets

In section 1.7 we saw that if $\sim$ is an equivariant equivalence relation on a G-set, then the set $X/{\sim}$ of equivalence classes $[x]_\sim$ becomes a G-set once we endow it with the G-action given by (1.42). The quotient function $\lambda x \in X \to [x]_\sim$ is a surjective equivariant function $X \twoheadrightarrow X/{\sim}$. So if X is a nominal set, then by part 3 of Lemma 2.11 we have that $X/{\sim}$ is also a nominal set.

By part 1 of that lemma we have $\operatorname{supp}[x]_\sim \subseteq \operatorname{supp} x$, for all $x \in X$. In fact for an equivalence class $c \in X/{\sim}$, $\operatorname{supp} c$ is the intersection of the supports of all its representatives $x \in c$; see Exercise 2.8.

The next section gives an important example of the quotient construction for nominal sets.

2.9 α-Equivalence

Consider the set $\Lambda/{=_\alpha}$ of terms of the *untyped λ-calculus* (Barendregt, 1984), which we can take to be the quotient by α-equivalence of the set Λ of terms t given by the grammar

$$t \in \Lambda ::= a \mid \lambda a.t \mid t\,t$$

where a ranges over $\mathbb{A}$, regarded as an infinite set of variables. The elements of Λ are sometimes called *raw terms* to distinguish them from the elements of the quotient $\Lambda/{=_\alpha}$.

The equivalence relation $=_\alpha$ is *α-equivalence*, which can be inductively defined by the following rules.

$$\frac{}{a =_\alpha a} \qquad \frac{t_1 =_\alpha t_1' \qquad t_2 =_\alpha t_2'}{t_1\,t_2 =_\alpha t_1'\,t_2'} \qquad \frac{\begin{array}{c}(a_1\ a) \cdot t_1 =_\alpha (a_2\ a) \cdot t_2 \\ a \notin \operatorname{var}(a_1\,t_1\,a_2\,t_2)\end{array}}{\lambda a_1.t_1 =_\alpha \lambda a_2.t_2}. \tag{2.21}$$

Here we are using the action of $\operatorname{Perm}\mathbb{A}$ on terms defined as in Example 1.4; and as in Example 2.6, Λ is a nominal set with $\operatorname{supp} t = \operatorname{var} t$, the finite set of variables occurring in the term t:

$$\begin{aligned} \operatorname{var} a &= \{a\} \\ \operatorname{var}(\lambda a.t) &= \{a\} \cup \operatorname{var} t \\ \operatorname{var}(t\,t') &= \operatorname{var} t \cup \operatorname{var} t'. \end{aligned} \tag{2.22}$$

The relation of α-equivalence is more traditionally defined to be the congruence generated by relating $\lambda a.t$ and $\lambda a'.\{a'/a\}t$ if there are no occurrences of a' in t, where $\{a'/a\}t$ is the term obtained from t by replacing all free occurrences of a with a'. The properties of this form of renaming are rather inconvenient, because the function $\lambda t \in \Lambda \to \{a'/a\}t$ does not necessarily respect α-equivalence when applied to terms that do contain occurrences of a'. This is because of the possible 'capture' of a' by binders $\lambda a'.\_$ occurring in t. For example, if a, a_1 and a_2 are three distinct atomic names, then $\lambda a_1.a =_\alpha \lambda a_2.a$ holds, but $\{a_1/a\}(\lambda a_1.a) = \lambda a_1.a_1 \neq_\alpha \lambda a_1.a_1 = \{a_1/a\}(\lambda a_2.a)$. In the traditional development of the theory of lambda calculus (Barendregt, 1984), this inconvenient fact immediately leads to the formulation of more complicated, 'capture-avoiding' notions of renaming and substitution. However, it is possible to go in the other direction and replace $\lambda t \in \Lambda \to \{a'/a\}t$ with another, equally simple form of renaming which does respect α-equivalence, namely the action of the transposition $(a\ a')$. For if a' does not occur in t, then $\{a'/a\}t$ is α-equivalent to the term $(a\ a') \cdot t$ obtained from t by swapping all occurrences of a and a'. It is for this reason that definition (2.22) coincides with the usual definition of α-equivalence (see Gabbay and Pitts, 2002, Proposition 2.2).

We noted in Example 1.6 that group action functions are equivariant. It follows from this and the Equivariance Principle that $=_\alpha$ is equivariant:

$$t =_\alpha t' \Rightarrow \pi \cdot t =_\alpha \pi \cdot t'. \tag{2.23}$$

It is also an equivalence relation. Proofs of the reflexivity and symmetry of $=_\alpha$ are easy, whereas the proof of its transitivity is less so; we discuss it in more detail in the proof of Lemma 5.5. From section 2.8 we have that the quotient set $\Lambda/{=_\alpha}$ of untyped λ-terms is a nominal set with $\operatorname{supp}[t]_{=_\alpha} \subseteq \operatorname{var} t$. In fact the support of a λ-term is equal to the set of free variables of any representative raw term:

$$\operatorname{supp}[t]_{=_\alpha} = \operatorname{fv} t \tag{2.24}$$

where

$$\begin{aligned} \operatorname{fv} a &= \{a\} \\ \operatorname{fv}(\lambda a.t) &= (\operatorname{fv} t) - \{a\} \\ \operatorname{fv}(t\,t') &= \operatorname{fv} t \cup \operatorname{fv} t'. \end{aligned} \tag{2.25}$$

This is because for all $t \in \Lambda$

$$((\forall a \in \operatorname{fv} t)\ \pi\, a = a) \Leftrightarrow \pi \cdot t =_\alpha t \tag{2.26}$$

and hence $\operatorname{fv} t$ strongly supports $[t]_{=_\alpha}$ in $\Lambda/{=_\alpha}$; so we can apply Theorem 2.7 to conclude that $\operatorname{supp}[t]_{=_\alpha}$ is equal to $\operatorname{fv} t$. The proof of (2.26) is left as an exercise.

Exercises

2.1 If X is a nominal set and $x \in X$, show that

$$\operatorname{supp} x = \{a \in \mathbb{A} \mid \{a' \in \mathbb{A} \mid (a\ a') \cdot x \neq x\} \text{ is an infinite set}\}. \tag{2.27}$$

[Hint: see the proof of (Gabbay and Pitts, 2002, Proposition 3.4).]

2.2 Give an example of an element of the power Perm $\mathbb{A}$-set $\mathrm{P}\,\mathbb{A}$ that is not finitely supported.

2.3 Consider the product Perm $\mathbb{A}$-set $\prod_{i \in I} X_i$ for the case $I = \mathbb{N} = \{0, 1, 2, \ldots\}$ and $X_i = \mathbb{A}$ (as in Example 1.3) for each $i \in \mathbb{N}$. Let $(a_0, a_1, a_2, \ldots)$ be an element of this product that enumerates the elements of $\mathbb{A}$ in the sense that each $a \in \mathbb{A}$ is equal to a_i for some $i \in \mathbb{N}$. Show that this element is not finitely supported.

2.4 Give an example of a countably infinite family $(X_n \mid n \in \mathbb{N})$ of non-empty nominal sets whose product in the category **Nom** is empty. [Hint: for each n, consider the set of finite subsets of $\mathbb{A}$ of cardinality $n + 1$.]

2.5 Consider Perm $\mathbb{A}$ with Perm $\mathbb{A}$-action as in Example 1.6. Show that each $\pi \in$ Perm $\mathbb{A}$ is finitely supported by $\{a \in \mathbb{A} \mid \pi\, a \neq a\}$ and hence that Perm $\mathbb{A}$ is a nominal set. Show that $a \notin \operatorname{supp} \pi \Rightarrow \pi\, a = a$ and deduce that $\operatorname{supp} \pi = \{a \in \mathbb{A} \mid \pi\, a \neq a\}$.

2.6 Show that the nominal set Perm $\mathbb{A}$ is the 'object of bijections' on $\mathbb{A}$ in the topos **Nom**. In other words it is isomorphic to the subobject of the exponential $\mathbb{A} \to_{\mathrm{fs}} \mathbb{A}$ given by the equivariant subset of functions that are both injective and surjective.

2.7 Let X be a G-set for some group G. Show that for every subset $S \subseteq X$ there is a greatest equivariant subset contained in S and a least equivariant subset containing S. Is the same true for finitely supported subsets when X is a nominal set?

2.8 Suppose X is a nominal set and S is a non-empty, finitely supported subset of X. Use Proposition 2.10 to show that for any $a \in \mathbb{A} - \operatorname{supp} S$ there exists $x \in S$ with $a \notin \operatorname{supp} x$. Deduce that if $\sim$ is an equivariant equivalence relation on X and $c \in X/{\sim}$, then $\operatorname{supp} c = \bigcap\{\operatorname{supp} x \mid x \in c\}$.

2.9 Prove (2.26) by induction on the size of raw terms t.

3

Freshness

In the previous chapter we explored the notion of the support of elements in sets upon which permutations of names act. The complementary notion of an atomic name not being in the support of an element is in many ways more important for applications of nominal sets. This is the relation of *freshness* that we explore in this chapter.

3.1 Freshness relation

Given nominal sets X and Y and elements $x \in X$ and $y \in Y$, we write $x \mathrel{\#} y$ and say that x is *fresh for* y if the two elements have disjoint supports:

$$x \mathrel{\#} y \Leftrightarrow \mathrm{supp}_X\, x \cap \mathrm{supp}_Y\, y = \emptyset. \tag{3.1}$$

Most of the time we use the freshness relation when $X = \mathbb{A}$ and $x = a$ is an atomic name. In this case since $\mathrm{supp}\, a = \{a\}$ (Example 2.4), $a \mathrel{\#} y$ means that $a \notin \mathrm{supp}\, y$, or equivalently that there some finite support for y that does not contain a. The *finiteness* of supports compared with the *in*finiteness of $\mathbb{A}$ leads to the following simple principle that we will use very often in what follows.

Choose-a-Fresh-Name Principle *If $X_1, \ldots, X_n$ are finitely many nominal sets and if $x_1 \in X_1, \ldots, x_n \in X_n$ are elements of them, then there is an atomic name $a \in \mathbb{A}$ satisfying $a \mathrel{\#} x_1 \wedge \cdots \wedge a \mathrel{\#} x_n$ (indeed, there are infinitely many such names).*

Note that by Proposition 2.10 the freshness relation is equivariant:

$$x \mathrel{\#} y \Rightarrow (\pi \cdot x) \mathrel{\#} (\pi \cdot y). \tag{3.2}$$

The following results restate some of the properties of supports from the previous chapter in terms of the freshness relation.

Proposition 3.1 *Let $x \in X$ be an element of a nominal set X. For all $a, a' \in \mathbb{A}$, if $a \mathrel{\#} x$ and $a' \mathrel{\#} x$, then $(a\ a') \cdot x = x$.*

Proof Apply Proposition 2.1. □

Proposition 3.2 *For any atoms $a, a' \in \mathbb{A}$ and finite sets of atoms $\overline{a}, \overline{a}' \in \mathrm{P_f}\,\mathbb{A}$*

$$\begin{aligned} a \mathrel{\#} a' &\Leftrightarrow a \neq a' \\ a \mathrel{\#} \overline{a} &\Leftrightarrow a \notin \overline{a} \\ \overline{a} \mathrel{\#} \overline{a}' &\Leftrightarrow \overline{a} \cap \overline{a}' = \emptyset. \end{aligned}$$

Proof Use the facts established in Examples 2.4 and 2.9 that $\operatorname{supp} a = \{a\}$ and $\operatorname{supp} \overline{a} = \overline{a}$. □

Proposition 3.3 *Let $X_1, \ldots, X_n$ be nominal sets. Then for all $x_1 \in X_1, \ldots, x_n \in X_n$, and $a \in \mathbb{A}$*

$$\begin{aligned} a \mathrel{\#} (x_1, \ldots, x_n) \in X_1 \times \cdots \times X_n &\Leftrightarrow a \mathrel{\#} x_1 \wedge \cdots \wedge a \mathrel{\#} x_n \\ a \mathrel{\#} \mathrm{inj}_i(x_i) \in X_1 + \cdots + X_n &\Leftrightarrow a \mathrel{\#} x_i. \end{aligned}$$

Proof These follow immediately from (2.10) and (2.9). □

Proposition 3.4 *Let X and Y be nominal sets.*

1. *Suppose $F \in \mathrm{P_{fs}}\,(X \times Y)$ is a partial function:*

$$(\forall x \in X)(\forall y, y' \in Y)\ (x, y) \in F \wedge (x, y') \in F \Rightarrow y = y'. \tag{3.3}$$

If $(x, y) \in F$, $a \mathrel{\#} F$ and $a \mathrel{\#} x$, then $a \mathrel{\#} y$.

2. *Suppose $F \in X \to_{\mathrm{fs}} Y$ and $x \in X$. If $a \mathrel{\#} F$ and $a \mathrel{\#} x$, then $a \mathrel{\#} F\,x$.*
3. *Suppose $F : X \to Y$ in* **Nom**. *For all $x \in X$, if $a \mathrel{\#} x$, then $a \mathrel{\#} F\,x$.*

Proof Part 2 is the special case of part 1 when F is total. Part 3 follows from part 2, because when F is equivariant, then as noted in Lemma 2.15 $\operatorname{supp} F = \emptyset$ and hence $a \mathrel{\#} F$ always holds. So it just remains to prove part 1; and for this it suffices to show that if $(x, y) \in F$ and $A \subseteq \mathbb{A}$ supports F and x, then A supports y.

If $\pi \in \operatorname{Perm} \mathbb{A}$ satisfies $(\forall a \in \mathbb{A} - A)\ \pi\,a = a$, then $\pi \cdot x = x$ and $\pi \cdot F = F$. Therefore $(x, \pi \cdot y) = (\pi \cdot x, \pi \cdot y) = \pi \cdot (x, y) \in \pi \cdot F = F$ and hence by (3.3), $\pi \cdot y = y$. Thus A supports y. □

Example 3.5 Freshness is not a unary 'logical relation' for functions. Although it is the case that $a \mathrel{\#} F$ implies $(\forall x \in X)\ a \mathrel{\#} x \ \Rightarrow\ a \mathrel{\#} F\,x$, the converse is false in general. For example, given an atomic name $a \in \mathbb{A}$, consider the function $F_a : \mathrm{P_f}\,\mathbb{A} \to \mathrm{P_f}\,\mathbb{A}$ mapping a finite subset $\overline{a}$ to $\overline{a} - \{a\}$. It is not hard to see that F_a satisfies

$$(\forall a_1, a_2 \in \mathbb{A} - \{a\})\ (a_1\ a_2) \cdot F_a = F_a$$

and hence F_a is supported by $\{a\}$; and it also satisfies

$$(\forall a' \in \mathbb{A} - \{a\})\ (a\ a') \cdot F_a \neq F_a$$

and hence is not supported by $\emptyset$. Therefore $F \in \mathrm{P_f}\,\mathbb{A} \to_{\mathrm{fs}} \mathrm{P_f}\,\mathbb{A}$ and $\mathrm{supp}\,F_a = \{a\}$. So it is not the case that $a \mathrel{\#} F$ holds. However F_a does satisfy

$$(\forall \overline{a} \in \mathrm{P_f}\,\mathbb{A})\ a \mathrel{\#} \overline{a} \Rightarrow a \mathrel{\#} F_a\,\overline{a}$$

since $a \notin \overline{a} - \{a\} = \mathrm{supp}(\overline{a} - \{a\}) = \mathrm{supp}(F_a\,\overline{a})$.

3.2 Freshness quantifier

Here is a very common pattern when reasoning informally with fresh names. At some point in a proof one chooses *some* fresh name with certain properties; later on in the proof one may need to revise that choice to take account of extra names that have entered the current context and so one needs to know that *any* fresh name with the property would have done for the original choice. The following results show that for finitely supported properties this switch from ‘some fresh’ to ‘any fresh’ is always possible.

Lemma 3.6 *Let $S \in \mathrm{P_{fs}}\,\mathbb{A}$ be a finitely supported set of atomic names, supported by $\overline{a} \in \mathrm{P_f}\,\mathbb{A}$ say. The following are equivalent.*

1. $(\exists a \in \mathbb{A})\ a \notin \overline{a} \wedge a \in S$.
2. $(\forall a' \in \mathbb{A})\ a' \notin \overline{a} \Rightarrow a' \in S$.
3. *S is cofinite.*

Proof If $a \in S - \overline{a}$, then for any $a' \in \mathbb{A} - \overline{a}$ we have $(a\ a') \cdot S = S$ since $a, a' \notin \overline{a}$ and the latter supports S; so $a' = (a\ a') \cdot a \in (a\ a') \cdot S = S$. So 1 implies 2.

We know from Proposition 2.8 that $\mathrm{P_{fs}}\,\mathbb{A}$ splits up as the disjoint union of two equivariant subsets, one consisting of all the finite subsets and the other consisting of all the cofinite subsets. Condition 2 says that S contains $\mathbb{A} - \overline{a}$ and hence is not finite. Therefore 2 implies 3.

Finally, if S is cofinite then so is $(\mathbb{A} - \overline{a}) \cap S$ and hence it is in particular non-empty. Therefore 3 implies 1. □

Definition 3.7 We define $\text{И} \triangleq \{S \subseteq \mathbb{A} \mid \mathbb{A} - S \text{ is finite}\}$ to be the set of cofinite sets of atomic names. If φ is the description (in higher-order logic, say) of some property of atomic names a, the *freshness quantifier*

$$(\text{И}a)\ \varphi \tag{3.4}$$

asserts that the set $\{a \in \mathbb{A} \mid \varphi\}$ is in И.

Note that this is a monotone quantifier in the sense that

$$\{a \in \mathbb{A} \mid \varphi\} \subseteq \{a \in \mathbb{A} \mid \varphi'\} \Rightarrow (\text{И}a)\ \varphi \Rightarrow (\text{И}a)\ \varphi'. \tag{3.5}$$

By definition $(\text{И}a)\ \varphi$ says that φ holds for all but finitely many atomic names; but in view of the following theorem, we can also read it as 'for some/any fresh a, φ', so long as the subset $\{a \in \mathbb{A} \mid \varphi\}$ determined by φ is finitely supported. By the Finite Support Principle stated at the end of section 2.5, this is the case for a wide range of properties, namely those expressible in classical higher-order logic (without the axiom of choice) using finitely supported primitives.

Theorem 3.8 (**'Some/any' theorem**) *Suppose that X is a nominal set and that $R \subseteq \mathbb{A} \times X$ is an equivariant subset. For each $x \in X$, the following are equivalent.*

1. $(\exists a \in \mathbb{A})\ a \mathrel{\#} x \wedge (a, x) \in R.$
2. $(\forall a \in \mathbb{A})\ a \mathrel{\#} x \Rightarrow (a, x) \in R.$
3. $(\text{И}a)\ (a, x) \in R.$

Proof Since R has empty support (by Lemma 2.21), it follows from Proposition 2.22 that for each $x \in X$ the subset $\{a \in \mathbb{A} \mid (a, x) \in R\}$ is supported by $\operatorname{supp} x$. So we can apply Lemma 3.6 with $S = \{a \in \mathbb{A} \mid (a, x) \in R\}$ and $\overline{a} = \operatorname{supp} x$. □

For example, the third rule in (2.21) for inductively generating α-equivalence between raw λ-terms can be restated as

$$\frac{(\text{И}a'')\ (a\ a'') \cdot t =_\alpha (a'\ a'') \cdot t'}{\lambda a.t =_\alpha \lambda a'.t'}$$

because $\{(a'', (a, t, a', t')) \mid (a\ a'') \cdot t =_\alpha (a'\ a'') \cdot t'\}$ is an equivariant subset of $\mathbb{A} \times (\mathbb{A} \times \Lambda \times \mathbb{A} \times \Lambda)$ and $a'' \mathrel{\#} (a, t, a', t')$ holds if and only if $a'' \notin \operatorname{var}(a\, t\, a'\, t')$.

The theorem expresses the freshness quantifier in terms of the freshness relation $a \mathrel{\#} x$. The converse is also possible:

$$a \mathrel{\#} x \Leftrightarrow (\text{И}a')\ (a\ a') \cdot x = x. \tag{3.6}$$

To see this, we can take $X = \mathbb{A} \times X$ and $R = \{(a', (a, x)) \in \mathbb{A} \times (\mathbb{A} \times X) \mid (a\ a') \cdot x = x\}$ in Theorem 3.8.

As the next result shows, the freshness quantifier has very regular behaviour with respect to the Boolean operations. Exercise 3.1 explores its commutation with existential and universal quantification.

Proposition 3.9 *Suppose φ and φ' are properties of atomic names for which $\{a \in \mathbb{A} \mid \varphi\}$ and $\{a \in \mathbb{A} \mid \varphi'\}$ are finitely supported subsets of $\mathbb{A}$. Then the following* *properties hold.*

$$\neg(\text{И}a)\ \varphi \Leftrightarrow (\text{И}a)\ \neg\varphi \tag{3.7}$$

$$((\text{И}a)\ \varphi \wedge (\text{И}a)\ \varphi') \Leftrightarrow (\text{И}a)\ \varphi \wedge \varphi' \tag{3.8}$$

and hence also

$$((\text{И}a)\ \varphi \vee (\text{И}a)\ \varphi') \Leftrightarrow (\text{И}a)\ \varphi \vee \varphi' \tag{3.9}$$

$$((\text{И}a)\ \varphi \Rightarrow (\text{И}a)\ \varphi') \Leftrightarrow (\text{И}a)\ \varphi \Rightarrow \varphi'. \tag{3.10}$$

Proof Since $S \triangleq \{a \in \mathbb{A} \mid \varphi(a)\}$ and $S' \triangleq \{a \in \mathbb{A} \mid \varphi'(a)\}$ are elements of $\mathrm{P}_{\mathrm{fs}}\,\mathbb{A}$, by Proposition 2.8 they are either finite or cofinite. So $S \notin \text{И}$ if and only if S is finite if and only if $\mathbb{A} - S \in \text{И}$; so we have (3.7). Since the union of two sets is finite if and only if they are both finite, we also have $S \cap S' \in \text{И} \Leftrightarrow S \in \text{И} \wedge S' \in \text{И}$, which gives (3.8). □

3.3 Local fresh atomic names

Many constructions on syntax, especially ones involving binding operations, make use of fresh names and involve verifying that the construction is independent of which fresh name is chosen. As the following theorem shows, the notion of 'finite support' built in to nominal sets can be used to give a simple condition that guarantees this independence.

Theorem 3.10 (**Freshness theorem**) *Let X be a nominal set. If a finitely supported partial function $F \in \mathbb{A} \rightharpoonup_{\mathrm{fs}} X$ satisfies*

$$(\text{И}a)(\exists x \in X)\ a \mathbin{\#} x \wedge F\,a \equiv x \tag{3.11}$$

then there is a unique element $\mathrm{fresh}_X\,F \in X$ *satisfying*

$$(\text{И}a)\ F\,a \equiv \mathrm{fresh}_X\,F. \tag{3.12}$$

Furthermore, $\mathrm{supp}(\mathrm{fresh}_X\,F) \subseteq \mathrm{supp}\,F$.

Proof Consider the equivariant subset

$$\mathrm{fresh}_X \triangleq \{(F, x) \in (\mathbb{A} \rightharpoonup_{\mathrm{fs}} X) \times X \mid (\text{И}a)\ F\,a \equiv x\}. \tag{3.13}$$

To prove the first sentence of the theorem we have to show that (3.13) is a partial function whose domain of definition contains those F satisfying (3.11); and then the second sentence follows by part 1 of Proposition 3.4. Note that

$$\begin{aligned}
&(F, x) \in \mathrm{fresh}_X \wedge (F, x') \in \mathrm{fresh}_X \\
\Rightarrow\ & \{\text{by (3.8)}\} \\
&(\text{И}a)\ (a, x) \in F \wedge (a, x') \in F \\
\Rightarrow\ & \{\text{since } F \text{ is single-valued}\} \\
&x = x'.
\end{aligned}$$

So fresh_X is single-valued. If F satisfies (3.11), then by the 'some/any' theorem (Theorem 3.8) there exists $a \in \mathbb{A}$ and $x \in X$ with $a \mathrel{\#} (F, x)$ and $(a, x) \in F$; hence $(\text{И}a)\ F\,a \equiv x$ holds and therefore $F \in \mathrm{Dom}\,\mathrm{fresh}_X$. □

Definition 3.11 If $\lambda a \in \mathbb{A} \to \varphi(a)$ is the description of some finitely supported partial function in $\mathbb{A} \rightharpoonup_{\mathrm{fs}} X$ satisfying condition (3.11) in the above theorem, then we write the element $\mathrm{fresh}_X(\lambda a \in \mathbb{A} \to \varphi(a))$ of X as

$$\mathrm{fresh}\, a \text{ in } \varphi(a). \tag{3.14}$$

With this notation we can summarize the freshness theorem by the formula

$$((\text{И}a)(\exists x \in X)\ a \mathrel{\#} x \equiv \varphi(a)) \Rightarrow (\text{И}a)\ \varphi(a) \equiv (\mathrm{fresh}\, a \text{ in } \varphi(a)). \tag{3.15}$$

3.4 Separated product

Since the freshness relation (3.1) is equivariant, for any nominal sets X_1 and X_2

$$X_1 * X_2 \triangleq \{(x_1, x_2) \in X_1 \times X_2 \mid x_1 \mathrel{\#} x_2\} \tag{3.16}$$

is again a nominal set (by Lemma 2.19). We call it the *separated product* of X_1 and X_2. We get a functor $\_ * \_ : \mathbf{Nom} \times \mathbf{Nom} \to \mathbf{Nom}$ by sending equivariant functions $F_1 : X_1 \to X_1'$ and $F_2 : X_2 \to X_2'$ to the equivariant function

$$\begin{aligned}
&F_1 * F_2 : X_1 * X_2 \to X_1' * X_2' \\
&F_1 * F_2 \triangleq \lambda(x, y) \in X_1 * X_2 \to (F_1\, x, F_2\, y)
\end{aligned} \tag{3.17}$$

since by part 1 of Lemma 2.11, $x_1 \mathrel{\#} x_2 \Rightarrow F_1\, x_1 \mathrel{\#} F_2\, x_2$.

Note that if X_2 is a discrete nominal set then the support of any of its elements is empty and therefore $X_1 * X_2 = X_1 \times X_2$. In particular the terminal object 1 satisfies $X_1 * 1 = X_1 \times 1 \cong X_1$. There are also natural isomorphisms $X_1 * X_2 \cong X_2 * X_1$ and $X_1 * (X_2 * X_3) \cong (X_1 * X_2) * X_3$ inherited from those for cartesian product. Altogether $(\mathbf{Nom}, *, 1)$ is a *symmetric monoidal category* (MacLane, 1971, chapter VII) that is *affine*, in the sense that the terminal object 1 is the unit for the tensor product $*$. The following result of Schöpp (2006, section 3.3.1) says that the affine symmetric monoidal structure is *closed*.

Theorem 3.12 *For each nominal set X, the functor* $\_ * X : \mathbf{Nom} \to \mathbf{Nom}$ *has a right adjoint* $X \mathrel{-\!\!*} \_ : \mathbf{Nom} \to \mathbf{Nom}$. *In other words for each* $Y \in \mathbf{Nom}$ *there is a nominal set* $X \mathrel{-\!\!*} Y$ *and an equivariant function* $\varepsilon : (X \mathrel{-\!\!*} Y) * X \to Y$ *with the universal property that given any equivariant function F as shown*

$$\begin{tikzcd}
Z * X \arrow[d, dashed, "\hat{F} * \mathrm{id}_X"'] \arrow[dr, "F"] & \\
(X \mathrel{-\!\!*} Y) * X \arrow[r, "\varepsilon"'] & Y
\end{tikzcd} \tag{3.18}$$

there is a unique equivariant function $\hat{F} : Z \to (X \mathrel{-\!\!*} Y)$ *making the above diagram commute.*

Proof In certain cases a simple description for $X \mathrel{-\!\!*} \_$ is possible—such as when $X = \mathbb{A}$ (see Theorem 4.11). However, in general no very simple description of $X \mathrel{-\!\!*} \_$ is known. Schöpp gives both an abstract, category-theoretic construction (building on previous work of Menni, 2003) and a more concrete description in terms of partial functions. We give the latter here, but it is not particularly edifying.

Using the nominal set of finitely supported partial functions ($X \rightharpoonup_{\mathrm{fs}} Y$, section 2.7) and the associated notion of Kleene equivalence ($\equiv$, section 1.6), define

$$E \triangleq \{F \in X \rightharpoonup_{\mathrm{fs}} Y \mid (\forall x \in X)\ x \mathrel{\#} F \Leftrightarrow x \in \mathrm{Dom}\, F\} \tag{3.19}$$

$$\sim\ \triangleq \{(F, F') \in E \times E \mid (\forall x \in X)\ x \mathrel{\#} (F, F') \Rightarrow F\,x \equiv F'\,x\} \tag{3.20}$$

$$\overline{F} \triangleq \bigcup\{F' \in E \mid F' \sim F\} \tag{3.21}$$

$$X \mathrel{-\!\!*} Y \triangleq \{F \in E \mid F = \overline{F}\}. \tag{3.22}$$

The important definition is the first one, but unfortunately E may be slightly too large to be $X \mathrel{-\!\!*} Y$. The relation $\sim$ identifies partial functions in E if they agree where both are defined. It is evidently reflexive and symmetric; not so evidently, it is also transitive (Exercise 3.2). It turns out that $X \mathrel{-\!\!*} Y \cong E/{\sim}$, but we can identify $X \mathrel{-\!\!*} Y$ with a subset of E using the closure operation $F \mapsto \overline{F}$ taking the union of the partial functions in the equivalence class $[F]_\sim$. One can show that if $F \in E$, then $\overline{F} \in E$ and $F \sim \overline{F}$.

By definition of E, the subset $\{((F, x), y) \mid (x, y) \in F\} \subseteq ((X \mathrel{-\!\!*} Y) * X) \times Y$ is the graph of an equivariant function $\varepsilon : (X \mathrel{-\!\!*} Y) * X \to Y$. We claim that this inherits the required universal property from the universal property of $\mathrm{app} \in (X \rightharpoonup_{\mathrm{fs}} Y) \times X \rightharpoonup Y$ described in section 2.7.

For the existence part of the universal property, note that each $F : Z * X \to Y$ is in particular an equivariant partial function $(Z \times X) \rightharpoonup Y$ via the inclusion $Z * X \subseteq Z \times X$. So it uniquely determines an equivariant function $\mathrm{curry}\, F : Z \to (X \rightharpoonup_{\mathrm{fs}} Y)$

satisfying (2.20). We claim that curry F maps Z into E, that is

$$(\forall z \in Z)(\forall x \in X)\ x \mathrel{\#} (\text{curry}\, F\, z) \Leftrightarrow x \in \text{Dom}(\text{curry}\, F\, z). \tag{3.23}$$

To see this, first note that if $x \in \text{Dom}(\text{curry}\, F\, z)$, then $(\exists y \in Y)\ z \mathrel{\#} x \wedge ((z, x), y) \in F$. Since curry F is equivariant, Lemma 2.11 implies that $\text{supp}(\text{curry}\, F\, z) \subseteq \text{supp}\, z$; so from $z \mathrel{\#} x$ we conclude that $x \mathrel{\#} \text{curry}\, F\, z$. Conversely if $x \mathrel{\#} (\text{curry}\, F\, z)$, then as in Exercise 3.2 we can find a finitary permutation $\pi \in \text{Perm}\,\mathbb{A}$ satisfying $(\pi \cdot x) \mathrel{\#} z$ and $(\forall a \in \text{supp}(\text{curry}\, F\, z))\ \pi\, a = a$; hence $(x, \pi^{-1} \cdot (F(z, \pi \cdot x))) \in \text{curry}\, F\, z$ and therefore $x \in \text{Dom}(\text{curry}\, F\, z)$. So we get an equivariant function

$$\begin{aligned} &\hat{F} : Z \to (X \mathrel{-\!\!\ast} Y) \\ &\hat{F} \triangleq \lambda z \in Z \to \overline{\text{curry}\, F\, z} \end{aligned} \tag{3.24}$$

that satisfies (3.18) because curry F satisfies (2.20).

For the uniqueness part of the universal property, suppose $F' : Z \to (X \mathrel{-\!\!\ast} Y)$ also satisfies (3.18), that is

$$(\forall z \in Z)(\forall x \in X)\ z \mathrel{\#} x \Rightarrow F' z\, x \equiv F(z, x). \tag{3.25}$$

Given $z \in Z$, for any $x \in X$ with $x \mathrel{\#} (F'z, \text{curry}\, F\, z)$, as above we can use some $\pi \in \text{Perm}\,\mathbb{A}$ that fixes the atomic names in $\text{supp}(F'z, \text{curry}\, F\, z)$, but moves x to $\pi \cdot x \mathrel{\#} z$, to deduce from (3.25) that $F' z\, x \equiv \text{curry}\, F\, z\, x$. Therefore $F' z \sim \text{curry}\, F\, z$ and hence $F' z = \overline{F' z} = \overline{\text{curry}\, F\, z} = \hat{F}\, z$, for all $z \in Z$. □

Exercises

3.1 Show that if $X \in$ **Nom** and $R \in \text{P}_{\text{fs}}\,(\mathbb{A} \times X)$, then

$$(\exists x \in X)(\text{И}a)\ (a, x) \in R \Rightarrow (\text{И}a)(\exists x \in X)\ (a, x) \in R \tag{3.26}$$

$$(\text{И}a)(\forall x \in X)\ (a, x) \in R \Rightarrow (\forall x \in X)(\text{И}a)\ (a, x) \in R \tag{3.27}$$

but that the reverse implications do not necessarily hold. [Hint: consider $R_1 = \{(a, a) \mid a \in \mathbb{A}\}$ and $R_2 = \{(a, a') \in \mathbb{A} \times \mathbb{A} \mid a \neq a'\}$.]

3.2 Given $X \in$ **Nom**, $\overline{a} \in \text{P}_{\text{f}}\,\mathbb{A}$ and $x \in X$ satisfying $\overline{a} \mathrel{\#} x$, show that for any other finite set of atomic names $\overline{a}' \in \text{P}_{\text{f}}\,\mathbb{A}$, there exists $\pi \in \text{Perm}\,\mathbb{A}$ with $(\forall a \in \overline{a})\ \pi\, a = a$ and $\overline{a}' \mathrel{\#} \pi \cdot x$. Deduce that the relation $\sim$ in (3.20) is transitive.

4

Name Abstraction

The original motivation for developing the theory of nominal sets was to extend the range of structural induction and recursion for algebraic data types to encompass quotients associated with the use of name-binding operations. Quotients of sets of algebraic terms by α-equivalence, like $\Lambda/{=_\alpha}$ from section 2.9, are isomorphic in **Nom** to nominal sets inductively defined using products $X{\times}Y$, coproducts $X{+}Y$ and a name abstraction construct $[\mathbb{A}]X$ for representing the domains of name-binding operations, which is the subject of this chapter.

4.1 Nominal set of name abstractions

Section 2.9 gave a structurally inductive characterization of α-equivalence $(=_\alpha)$ for the untyped λ-calculus that makes use of name permutations rather than more general renaming operations on λ-terms. This suggests a generalized form of α-equivalence that applies to the elements of any nominal set X and not just to nominal sets of algebraic terms. Define the binary relation $\approx_\alpha$ on $\mathbb{A} \times X$ by

$$(a_1, x_1) \approx_\alpha (a_2, x_2) \Leftrightarrow (\text{И}a)\ (a_1\ a) \cdot x_1 = (a_2\ a) \cdot x_2. \tag{4.1}$$

Since the swapping operation is equivariant (Proposition 1.15), by Theorem 3.8 we have

$$\begin{aligned}(a_1, x_1) \approx_\alpha (a_2, x_2) &\Leftrightarrow (\exists a \mathrel{\#} (a_1, x_1, a_2, x_2))\ (a_1\ a) \cdot x_1 = (a_2\ a) \cdot x_2 \\ &\Leftrightarrow (\forall a \mathrel{\#} (a_1, x_1, a_2, x_2))\ (a_1\ a) \cdot x_1 = (a_2\ a) \cdot x_2.\end{aligned} \tag{4.2}$$

Equivariance of swapping implies that $\approx_\alpha$ is an equivariant relation:

$$(a_1, x_1) \approx_\alpha (a_2, x_2) \Rightarrow (\pi\, a_1, \pi \cdot x_1) \approx_\alpha (\pi\, a_2, \pi \cdot x_2). \tag{4.3}$$

Lemma 4.1 *$\approx_\alpha$ is an equivalence relation.*

Proof It is immediate from its definition that $\approx_\alpha$ is reflexive and symmetric. Transitivity follows from the fact that the freshness quantifier commutes with conjunction:

$$\begin{array}{ll}
& (a_1, x_1) \approx_\alpha (a_2, x_2) \wedge (a_2, x_2) \approx_\alpha (a_3, x_3) \\
\Leftrightarrow & \{\text{by definition}\} \\
& (\text{И}a)\ (a_1\ a) \cdot x_1 = (a_2\ a) \cdot x_2 \wedge (\text{И}a)\ (a_2\ a) \cdot x_2 = (a_3\ a) \cdot x_3 \\
\Leftrightarrow & \{\text{by Proposition 3.9}\} \\
& (\text{И}a)\ (a_1\ a) \cdot x_1 = (a_2\ a) \cdot x_2 \wedge (a_2\ a) \cdot x_2 = (a_3\ a) \cdot x_3 \\
\Rightarrow & \\
& (\text{И}a)\ (a_1\ a) \cdot x_1 = (a_3\ a) \cdot x_3 \\
\Leftrightarrow & \{\text{by definition}\} \\
& (a_1, x_1) \approx_\alpha (a_3, x_3).
\end{array}$$

□

Lemma 4.2 *For all $a \in \mathbb{A}$ and $x_1, x_2 \in X$, $(a, x_1) \approx_\alpha (a, x_2)$ holds if and only if $x_1 = x_2$.*

Proof Immediate from the definition of $\approx_\alpha$. □

Lemma 4.3 *$(a_1, x_1) \approx_\alpha (a_2, x_2)$ holds if and only if either $a_1 = a_2$ and $x_1 = x_2$, or $a_1 \mathbin{\#} (a_2, x_2)$ and $x_1 = (a_1\ a_2) \cdot x_2$.*

Proof We split the proof into two cases, according to whether or not a_1 and a_2 are equal. In case $a_1 = a_2$ we can just apply Lemma 4.2. So suppose $a_1 \neq a_2$. If $(a_1, x_1) \approx_\alpha (a_2, x_2)$, then by (4.2) $(a_1\ a) \cdot x_1 = (a_2\ a) \cdot x_2$ holds for some $a \mathbin{\#} (a_1, x_1, a_2, x_2)$. Therefore

$$\begin{array}{ll}
& a_1 \\
= & \{\text{since } a_1 \neq a_2, a\} \\
& (a_2\ a) \cdot (a_1\ a) \cdot a \\
\# & \{\text{by (3.2), since } a \mathbin{\#} x_1\} \\
& (a_2\ a) \cdot (a_1\ a) \cdot x_1 \\
= & \{\text{since } (a_1\ a) \cdot x_1 = (a_2\ a) \cdot x_2\} \\
& (a_2\ a) \cdot (a_2\ a) \cdot x_2 \\
= & \{\text{since } (a_2\ a) \circ (a_2\ a) = \text{id}\} \\
& x_2
\end{array}$$

and hence also

$$
\begin{aligned}
& (a_1\ a_2) \cdot x_2 \\
= \ & \{\text{by Theorem 3.1, since } a_1 \# x_2 \text{ and } a \# x_2\} \\
& (a_1\ a_2) \cdot (a_1\ a) \cdot x_2 \\
= \ & \{\text{since } (a_1\ a_2) \circ (a_1\ a) = (a_1\ a) \circ (a_2\ a)\} \\
& (a_1\ a) \cdot (a_2\ a) \cdot x_2 \\
= \ & \{\text{since } (a_1\ a) \cdot x_1 = (a_2\ a) \cdot x_2\} \\
& (a_1\ a) \cdot (a_1\ a) \cdot x_1 \\
= \ & \{\text{since } (a_1\ a) \circ (a_1\ a) = \mathrm{id}\} \\
& x_1 .
\end{aligned}
$$

Conversely, if $a_1 \# (a_2, x_2)$ and $x_1 = (a_1\ a_2) \cdot x_2$, then for any $a \# (a_1, x_1, a_2, x_2)$ we have

$$
\begin{aligned}
& (a_1\ a) \cdot x_1 \\
= \ & \{\text{since } x_1 = (a_1\ a_2) \cdot x_2\} \\
& (a_1\ a) \cdot (a_1\ a_2) \cdot x_2 \\
= \ & \{\text{since } (a_1\ a) \circ (a_1\ a_2) = (a_2\ a) \cdot (a_1\ a)\} \\
& (a_2\ a) \cdot (a_1\ a) \cdot x_2 \\
= \ & \{\text{by Theorem 3.1, since } a_1 \# x_2 \text{ and } a \# x_2\} \\
& (a_2\ a) \cdot x_2 .
\end{aligned}
$$

Therefore $(a_1, x_1) \approx_\alpha (a_2, x_2)$, by (4.2). □

So $\approx_\alpha$ is an equivariant equivalence relation and as in section 2.8, the quotient of $\mathbb{A} \times X$ by $\approx_\alpha$ is a nominal set.

Definition 4.4 Given a nominal set X, let $\approx_\alpha$ be as in (4.1). We denote the nominal quotient set $(\mathbb{A} \times X)/{\approx_\alpha}$ by $[\mathbb{A}]X$ and call it the *nominal set of name abstractions* of elements of X. The equivalence class of $(a, x) \in \mathbb{A} \times X$ is denoted $\langle a \rangle x$ and called a *name abstraction*. Thus the $\mathrm{Perm}\,\mathbb{A}$-action on $[\mathbb{A}]X$ is well-defined by

$$\pi \cdot \langle a \rangle x \triangleq \langle \pi\, a \rangle (\pi \cdot x). \tag{4.4}$$

We noted in section 2.8 that an equivalence class is supported by any set of atomic names that supports a representative of the class. Therefore in $[\mathbb{A}]X$ we have $\mathrm{supp}\,\langle a \rangle x \subseteq \mathrm{supp}(a, x) = \{a\} \cup \mathrm{supp}\, x$. However, we can do better than this.

Proposition 4.5 *Given any nominal set X, for all $a \in \mathbb{A}$ and $x \in X$,* $\mathrm{supp}\,\langle a \rangle x = \mathrm{supp}\, x - \{a\}$. *Hence for all $a' \in \mathbb{A}$*

$$a' \# \langle a \rangle x \Leftrightarrow a' = a \vee a' \# x. \tag{4.5}$$

Proof First note that if $\pi \in \mathrm{Perm}\,\mathbb{A}$ satisfies $\pi \cdot \langle a \rangle x = \langle a \rangle x$ and $\pi\, a = a$, then by Lemma 4.2 and (4.4) it also satisfies $\pi \cdot x = x$. Therefore if $A \subseteq \mathbb{A}$ supports

$\langle a \rangle x$ in $[\mathbb{A}]X$, then $A \cup \{a\}$ supports x in X. Hence $\operatorname{supp} x \subseteq \{a\} \cup \operatorname{supp} \langle a \rangle x$. Since we noted above that $\operatorname{supp} \langle a \rangle x \subseteq \{a\} \cup \operatorname{supp} x$, it just remains to show that $a \notin \operatorname{supp} \langle a \rangle x$, that is, $a \mathrel{\#} \langle a \rangle x$. Using the Choose-a-Fresh-Name Principle to pick some $a' \mathrel{\#} (a, x)$, from Lemma 4.3 we have $(a', (a' \ a) \cdot x) \approx_\alpha (a, x)$ and hence $(a' \ a) \cdot \langle a \rangle x = \langle a' \rangle ((a' \ a) \cdot x) = \langle a \rangle x$. Since $a' \mathrel{\#} (a, x)$, we have $a' \mathrel{\#} \langle a \rangle x$; so by equivariance of # we get $a = (a' \ a) \cdot a' \mathrel{\#} (a' \ a) \cdot \langle a \rangle x = \langle a \rangle x$, as required. □

Example 4.6 If X is a discrete nominal set, then $a \mathrel{\#} x$ holds for all $a \in \mathbb{A}$ and $x \in X$. So in this case $(a_1, x_1) \approx_\alpha (a_2, x_2) \Leftrightarrow x_1 = x_2$. Thus $\langle a \rangle x \mapsto x$ is a well-defined equivariant function witnessing the isomorphism

$$[\mathbb{A}]X \cong X \quad (X \text{ discrete}). \tag{4.6}$$

4.2 Concretion

In this section we compare name abstraction with the more familiar notion of function abstraction. Recall that each $\langle a \rangle x \in [\mathbb{A}]X$ is an equivalence class for the relation $\approx_\alpha$ and hence in particular is a subset of $\mathbb{A} \times X$. Lemma 4.2 implies that it is single-valued; and of course it is finitely supported. So we have an inclusion

$$[\mathbb{A}]X \subseteq (\mathbb{A} \rightharpoonup_{\mathrm{fs}} X). \tag{4.7}$$

Lemma 4.7 *The domain of definition of each $F \in [\mathbb{A}]X$ regarded as a partial function from $\mathbb{A}$ to X is the cofinite set of atomic names that are fresh for it:*

$$\operatorname{Dom} F = \{a \in \mathbb{A} \mid a \mathrel{\#} F\} = \mathbb{A} - \operatorname{supp} F.$$

Proof Suppose $F = \langle a \rangle x$. Then $\operatorname{Dom} F = \{a' \mid (\exists x')\ (a', x') \approx_\alpha (a, x)\}$. Applying Lemma 4.3, we get $\operatorname{Dom} F = \{a' \mid a' = a \vee a' \mathrel{\#} x\}$, from which the result follows by Proposition 4.5. □

Definition 4.8 Given a nominal set X, the result of applying a name abstraction $F \in [\mathbb{A}]X$, regarded as a partial function from $\mathbb{A}$ to X, to an atomic name $a \in \operatorname{Dom} F = \mathbb{A} - \operatorname{supp} F$ will be denoted $F \mathbin{@} a$ and called the *concretion* of F at a.

Thus from Lemma 4.3 we have for all $a, a' \in \mathbb{A}$ and $x \in X$

$$(\langle a \rangle x) \mathbin{@} a' \equiv \begin{cases} x & \text{if } a' = a \\ (a \ a') \cdot x & \text{if } a' \neq a \text{ and } a' \mathrel{\#} x \\ \text{undefined} & \text{otherwise.} \end{cases} \tag{4.8}$$

Property (4.8) is the analogue for name abstraction/concretion of β-equivalence for function abstraction/application. The analogue of η-equivalence is given by the following result.

Proposition 4.9 *Given a nominal set X and a name abstraction $F \in [\mathbb{A}]X$, for some/any $a \# F$ it is the case that $F = \langle a \rangle (F \mathbin{@} a)$.*

Proof Suppose $F = \langle a' \rangle x'$. If $a \# F$ then by Proposition 4.5 either $a = a'$, or $a \# (a', x')$. In the first case $F \mathbin{@} a = x'$ and so $\langle a \rangle (F \mathbin{@} a) = \langle a' \rangle x' = F$. In the second case $F \mathbin{@} a = (a'\ a) \cdot x'$ and so $\langle a \rangle (F \mathbin{@} a) = \langle a \rangle ((a'\ a) \cdot x') = \langle a' \rangle x' = F$. □

We can sum up the theorem by the formula

$$(\forall F \in [\mathbb{A}]X)(\text{И} a)\ \langle a \rangle (F \mathbin{@} a) \equiv F. \tag{4.9}$$

Just as η-equivalence is connected to function extensionality, so here we have an extensionality principle for name abstractions

$$(\forall F, F' \in [\mathbb{A}]X)\ F = F' \Leftrightarrow (\text{И} a)\ F \mathbin{@} a \equiv F' \mathbin{@} a. \tag{4.10}$$

For if the concretions of F and F' at $a \# (F, F')$ are equal, to $x \in X$ say, then by the theorem $F = \langle a \rangle x = F'$.

4.3 Functoriality

If $X, Y \in \mathbf{Nom}$ and $F \in X \to_{\text{fs}} Y$, we can use the freshness theorem (Theorem 3.10) to get a finitely supported function

$$\begin{aligned} &[\mathbb{A}]F \in [\mathbb{A}]X \to_{\text{fs}} [\mathbb{A}]Y \\ &[\mathbb{A}]F \triangleq \lambda z \in [\mathbb{A}]X \to \text{fresh}\, a \text{ in } \langle a \rangle (F(z \mathbin{@} a)). \end{aligned} \tag{4.11}$$

satisfying

$$(\text{И} a)\ [\mathbb{A}]F\, z = \langle a \rangle (F(z \mathbin{@} a)) \tag{4.12}$$

for all $z \in \langle \mathbb{A} \rangle X$.

Lemma 4.10 $[\mathbb{A}]\text{id}_X = \text{id}_{[\mathbb{A}]X}$ *and* $[\mathbb{A}](F' \circ F) = ([\mathbb{A}]F') \circ ([\mathbb{A}]F)$.

Proof For each $z \in [\mathbb{A}]X$, pick some $a \# (z, F, F')$. Then $[\mathbb{A}]\text{id}_X\, z = \langle a \rangle (id_X(z \mathbin{@} a)) = \langle a \rangle (z \mathbin{@} a) = z$, by Proposition 4.9. Also $[\mathbb{A}]F\, z = \langle a \rangle (F(z \mathbin{@} a))$ and so $a \# [\mathbb{A}]F\, z$ by Proposition 4.5; therefore $[\mathbb{A}]F'([\mathbb{A}]F\, z) = \langle a \rangle (F'(([\mathbb{A}]F\, z) \mathbin{@} a)) = \langle a \rangle (F'(F(z \mathbin{@} a))) = [\mathbb{A}](F' \circ F)z$. □

Note that if $F : X \to Y$ is equivariant and hence an element of $X \to_{\text{fs}} Y$ with empty support, then $[\mathbb{A}]F$ is an equivariant function $[\mathbb{A}]X \to [\mathbb{A}]Y$; and in this case from (4.12) we have for all $a \in \mathbb{A}$ and $x \in X$

$$[\mathbb{A}]F(\langle a \rangle x) = \langle a \rangle (F\, x) \quad (F \text{ equivariant}). \tag{4.13}$$

Remark The lemma implies that $[\mathbb{A}]\_$ is not just a functor $\mathbf{Nom} \to \mathbf{Nom}$, but is a 'strong' one. For being cartesian closed, the category $\mathbf{Nom}$ is enriched over itself (see Johnstone, 2002, section B2.1) and in view of the lemma, the equivariant functions

$$\begin{array}{rcl} (X \to_{\mathrm{fs}} Y) & \to & ([\mathbb{A}]X \to_{\mathrm{fs}} [\mathbb{A}]Y) \\ F \in (X \to_{\mathrm{fs}} Y) & \mapsto & [\mathbb{A}]F \end{array} \qquad (X, Y \in \mathbf{Nom})$$

make $[\mathbb{A}]\_$ into a $\mathbf{Nom}$-enriched functor. (Such functors are sometimes called *strong*.)

Theorem 4.11 *The functor* $[\mathbb{A}]\_ : \mathbf{Nom} \to \mathbf{Nom}$ *is right adjoint to the functor* $\_ * \mathbb{A} : \mathbf{Nom} \to \mathbf{Nom}$ *given by taking the separated product with* $\mathbb{A}$*. Thus*

$$[\mathbb{A}]\_ \cong \mathbb{A} \mathbin{-\!\!*} \_$$

where $\mathbin{-\!\!*}$ *is as in Theorem 3.12.*

Proof Given $X \in \mathbf{Nom}$, by Lemma 4.7 the partial operation of concretion has $([\mathbb{A}]X) * \mathbb{A}$ for its domain of definition and so gives an equivariant function

$$\begin{aligned} &\mathrm{conc}_X : ([\mathbb{A}]X) * \mathbb{A} \to X \\ &\mathrm{conc}_x \triangleq \lambda(z, a) \in ([\mathbb{A}]X) * \mathbb{A} \to z \mathbin{@} a. \end{aligned} \tag{4.14}$$

We claim this has the universal property needed for $[\mathbb{A}]X$ to be the value of the right adjoint to $\_ * \mathbb{A}$ at X, namely that given any equivariant function F as shown

$$\begin{tikzcd} Y * \mathbb{A} \arrow[d, dashed, "\hat{F} * \mathrm{id}_{\mathbb{A}}"'] \arrow[dr, "F"] & \\ ([\mathbb{A}]X) * \mathbb{A} \arrow[r, "\mathrm{conc}_X"'] & X \end{tikzcd} \tag{4.15}$$

there is a unique equivariant function $\hat{F} : Y \to [\mathbb{A}]X$ making the above diagram commute. For using the freshness theorem (Theorem 3.10) we can define

$$\hat{F} \triangleq \lambda y \in Y \to \mathrm{fresh}\, a \text{ in } \langle a \rangle (F(y, a)) \tag{4.16}$$

to get such an equivariant function. It is the unique such, since if F' is any other, then for any $y \in Y$, picking some $a \in \mathbb{A}$ with $a \# y$, we also have $a \# F'y$ (by part 3 of Proposition 3.4) and hence

$$\begin{aligned}
& F'\,y \\
= \quad & \{\text{by Proposition 4.9}\} \\
& \langle a\rangle((F'y) \mathbin{@} a) \\
= \quad & \{\text{since } \mathrm{conc}_X \circ (F' * \mathrm{id}_{\mathbb{A}}) = F\} \\
& \langle a\rangle(F(y, a)) \\
= \quad & \{\text{by definition of } \hat{F}\text{, since } a \mathrel{\#} y\} \\
& \hat{F}\,y.
\end{aligned}$$

□

Theorem 4.12 *The functor* $[\mathbb{A}]\_ : \mathbf{Nom} \to \mathbf{Nom}$ *has a right adjoint.*

Proof For each $X \in \mathbf{Nom}$

$$R\,X \triangleq \{F \in \mathbb{A} \to_{\mathrm{fs}} X \mid (\forall a \in \mathbb{A})\ a \mathrel{\#} F\,a\} \tag{4.17}$$

is an equivariant subset of $\mathbb{A} \to_{\mathrm{fs}} X$ and hence a nominal set by Lemma 2.19. Because of the way $R\,X$ is defined, we can use the partial operation of concretion from section 4.2 and the freshness theorem (Theorem 3.10) to obtain an equivariant function

$$\begin{aligned}
&\varepsilon_X : [\mathbb{A}](R\,X) \to X \\
&\varepsilon_X \triangleq \lambda z \in [\mathbb{A}](R\,X) \to \mathrm{fresh}\, a \text{ in } (z \mathbin{@} a)\, a.
\end{aligned} \tag{4.18}$$

We will show that this has the universal property required to make $R\,X$ the value of the right adjoint to $[\mathbb{A}]\_$ at X, namely that given any equivariant function F as shown

$$\begin{tikzcd}
{[\mathbb{A}]Y} \arrow[d, dashed, "{[\mathbb{A}]\hat{F}}"'] \arrow[dr, "F"] & \\
{[\mathbb{A}](R\,X)} \arrow[r, "\varepsilon_X"'] & X
\end{tikzcd} \tag{4.19}$$

there is a unique equivariant function $\hat{F} : Y \to R\,X$ making the above diagram commute. First note that by (4.13) and (4.18), commutation of (4.19) is equivalent to

$$(\forall a \in \mathbb{A})(\forall y \in Y)\ \hat{F}\,y\,a = F(\langle a\rangle y). \tag{4.20}$$

So there is at most one such $\hat{F}$. If $y \in Y$ and $a \in \mathbb{A}$, then $a \mathrel{\#} F(\langle a\rangle y)$ (by Propositions 3.4 and 4.5) and hence $\hat{F} \triangleq \lambda y \in Y \to (\lambda a \in \mathbb{A} \to F(\langle a\rangle y))$ does indeed give an equivariant function $Y \to R\,X$ satisfying (4.20). □

This theorem shows that $[\mathbb{A}]\_$ gives a notion of abstraction rather different from the function abstraction $\mathbb{A} \to_{\mathrm{fs}} \_$, which most certainly does not have a right adjoint. For note that functors with right adjoints preserve colimits. In particular we have

$$[\mathbb{A}](X_1 + X_2) \cong ([\mathbb{A}]X_1) + ([\mathbb{A}]X_2) \tag{4.21}$$

whereas $\mathbb{A} \to_{\mathrm{fs}} (X_1 + X_2)$ is not in general isomorphic to the disjoint union of $\mathbb{A} \to_{\mathrm{fs}} X_1$ and $\mathbb{A} \to_{\mathrm{fs}} X_2$. (Just consider the case $X_1 = X_2 = 1$ to see this.)

Since by Theorem 4.11 $[\mathbb{A}]_-$ has a left adjoint, it also preserves limits in **Nom**. In particular

$$[\mathbb{A}](X_1 \times X_2) \cong ([\mathbb{A}]X_1) \times ([\mathbb{A}]X_2). \tag{4.22}$$

It is a useful exercise to construct the isomorphisms in (4.21) and (4.22) explicitly (Exercises 4.3 and 4.4).

The adjoint properties of the name abstraction functor do not explain the following somewhat surprising preservation property discovered by Gabbay (2000, Corollary 9.6.9).

Proposition 4.13 *The functor* $[\mathbb{A}]_- : \mathbf{Nom} \to \mathbf{Nom}$ *preserves exponentials:*

$$[\mathbb{A}](X \to_{\mathrm{fs}} Y) \cong ([\mathbb{A}]X) \to_{\mathrm{fs}} ([\mathbb{A}]Y).$$

Proof Using the partial operation of concretion and the freshness theorem (Theorem 3.10) we get equivariant functions

$$\begin{aligned}
&I : [\mathbb{A}](X \to_{\mathrm{fs}} Y) \longrightarrow ([\mathbb{A}]X) \to_{\mathrm{fs}} ([\mathbb{A}]Y) && (4.23)\\
&I \triangleq \lambda u \in [\mathbb{A}](X \to_{\mathrm{fs}} Y) \to \lambda z \in [\mathbb{A}]X \to \text{fresh}\, a \text{ in } \langle a\rangle((u \mathbin{@} a)(z \mathbin{@} a))\\
&J : ([\mathbb{A}]X) \to_{\mathrm{fs}} ([\mathbb{A}]Y) \longrightarrow [\mathbb{A}](X \to_{\mathrm{fs}} Y) && (4.24)\\
&J \triangleq \lambda F \in ([\mathbb{A}]X) \to_{\mathrm{fs}} ([\mathbb{A}]Y) \to \text{fresh}\, a \text{ in } \langle a\rangle(\lambda x \in X \to F(\langle a\rangle x) \mathbin{@} a).
\end{aligned}$$

that we show that they are mutually inverse.

First note that for all $a \in \mathbb{A}$, $F \in X \to_{\mathrm{fs}} Y$ and $x \in X$, the definition of @ and I give $I(\langle a\rangle F)(\langle a\rangle x) = \langle a\rangle(F\,x)$. Hence by definition of J we have

$$J(I(\langle a\rangle F)) = \langle a\rangle(\lambda x \in X \to (\langle a\rangle(F\,x)) \mathbin{@} a) = \langle a\rangle(\lambda x \in X \to F\,x) = \langle a\rangle F.$$

Thus $J \circ I = \mathrm{id}$. Conversely, given any $F \in ([\mathbb{A}]X) \to_{\mathrm{fs}} ([\mathbb{A}]Y)$ and $z \in [\mathbb{A}]X$, pick some $a \in \mathbb{A}$ with $a \mathbin{\#} (F, z)$. Then by definition of J

$$(J\,F) \mathbin{@} a = \lambda x \in X \to F(\langle a\rangle x) \mathbin{@} a \tag{4.25}$$

and hence

$$
\begin{aligned}
& I(J\,F)z \\
= \quad & \{\text{by definition of } I\} \\
& \langle a\rangle(((J\,F) \mathbin{@} a)(z \mathbin{@} a)) \\
= \quad & \{\text{by (4.25)}\} \\
& \langle a\rangle(F(\langle a\rangle(z \mathbin{@} a)) \mathbin{@} a)) \\
= \quad & \{\text{by Proposition 4.9}\} \\
& \langle a\rangle((F\,z) \mathbin{@} a) \\
= \quad & \{\text{by Proposition 4.9}\} \\
& F\,z.
\end{aligned}
$$

Thus $I \circ J = \mathrm{id}$. □

4.4 Freshness condition for binders

In this section we give analysis of what is needed to specify functions of name abstractions somewhat different from the one in Theorem 4.12. By construction, functions with domain $[\mathbb{A}]X$ correspond to functions with domain $\mathbb{A} \times X$ that respect the generalized α-equivalence relation $\approx_\alpha$. The next theorem shows that the requirement that a function respects $\approx_\alpha$ is equivalent to a simpler condition involving freshness, called the *freshness condition for binders* in (Pitts, 2006). At the same time the theorem embodies another common pattern when defining functions on α-equivalence classes, namely that one only specifies the function for bound names avoiding some finite set of 'bad' names. (The support of F plays this role in the theorem below.) For example, when defining capture-avoiding substitution for a λ-term $\lambda a.t$ one can just say what to do when a avoids the finite set of free variables of the term to be substituted.

Theorem 4.14 *Given $X, Y \in$ **Nom** and a finitely supported partial function $F \in (\mathbb{A} \times X) \rightharpoonup_{\mathrm{fs}} Y$ satisfying*

$$(\text{И}a)(\forall x \in X)(\exists y \in Y)\ a \mathrel{\#} y \wedge F(a, x) \equiv y \tag{4.26}$$

there is a unique finitely supported total function $\overline{F} \in ([\mathbb{A}]X) \to_{\mathrm{fs}} Y$ satisfying

$$(\text{И}a)(\forall x \in X)\ F(a, x) \equiv \overline{F}(\langle a\rangle x). \tag{4.27}$$

In this case $\operatorname{supp} \overline{F} \subseteq \operatorname{supp} F$.

Proof If F satisfies (4.26), then for each $z \in [\mathbb{A}]X$ we can apply the freshness theorem to the partial function $\{(a, y) \mid a \mathrel{\#} (F, z) \wedge F(a, z \mathbin{@} a) \equiv y\}$ to get

$$\overline{F}\,z \triangleq \text{fresh}\,a \text{ in } F(a, z \mathbin{@} a). \tag{4.28}$$

Thus for all $z \in [\mathbb{A}]X$ and $y \in Y$

$$\overline{F}\,z = y \Leftrightarrow (\text{И}a)(\exists x \in X)\ z = \langle a\rangle x \wedge y = F(a, x). \tag{4.29}$$

By Proposition 2.22, the right-hand side in (4.29) defines a subset of $([\mathbb{A}]X) \times Y$ that is supported by $\operatorname{supp} F$; hence $\overline{F} \in ([\mathbb{A}]X) \to_{\text{fs}} Y$ and $\operatorname{supp} \overline{F} \subseteq \operatorname{supp} F$. This function satisfies property (4.27), because given any $a \mathrel{\#} (F, \overline{F})$ and $x \in X$, since $a \mathrel{\#} (F, \langle a\rangle x)$, definition (4.28) gives us $\overline{F}(\langle a\rangle x) \equiv F(a, (\langle a\rangle x) \mathbin{@} a) \equiv F(a, x)$; thus

$$(\forall a \in \mathbb{A})\ a \mathrel{\#} (F, \overline{F}) \Rightarrow (\forall x \in X)\ F(a, x) \equiv \overline{F}(\langle a\rangle x) \tag{4.30}$$

and so (4.27) holds by Theorem 3.8. Finally, the uniqueness of $\overline{F}$ is immediate from property (4.27), since if F' is any other such function, then by the Choose-a-Fresh-Name Principle and Lemma 4.3 for every $z \in [\mathbb{A}]X$ we can find $a \mathrel{\#} (F, \overline{F}, F')$ and $x \in X$ with $z = \langle a\rangle x$ and hence with $F'\,z \equiv F'(\langle a\rangle x) \equiv F(a, x) \equiv \overline{F}(\langle a\rangle x) \equiv \overline{F}\,z$. □

Notation The theorem justifies the following use of name abstraction patterns in notation for functions with domain $[\mathbb{A}]X$. If $\lambda(a, x) \in \mathbb{A} \times X \to \varphi(a, x)$ is the description of some finitely supported partial function $F \in (\mathbb{A} \times X) \rightharpoonup_{\text{fs}} Y$ satisfying condition (4.26), then we write the function $\overline{F} \in ([\mathbb{A}]X) \to_{\text{fs}} Y$ from Theorem 4.14 as

$$\lambda\langle a\rangle x \in [\mathbb{A}]X \to \varphi(a, x). \tag{4.31}$$

The following corollary of the theorem gives a simple criterion for defining equivariant functions with parameters on nominal sets of name abstractions.

Corollary 4.15 *Given $X, Y, Z \in$ **Nom**, suppose the equivariant function $F : X \times \mathbb{A} \times Y \to Z$ satisfies*

$$(\forall x \in X)(\text{И}a)(\forall y \in Y)\ a \mathrel{\#} F(x, a, y). \tag{4.32}$$

Then there is a unique equivariant function $\overline{F} : X \times [\mathbb{A}]Y \to Z$ satisfying

$$(\forall x \in X)(\text{И}a)(\forall y \in Y)\ \overline{F}(x, \langle a\rangle y) = F(x, a, y). \tag{4.33}$$

Proof If F satisfies (4.32), then for each $x \in X$

$$\begin{aligned}&\operatorname{curry} F\,x \in (\mathbb{A} \times Y) \to_{\text{fs}} Z\\ &\operatorname{curry} F\,x = \lambda(a, y) \in \mathbb{A} \times Y \to F(x, a, y)\end{aligned}$$

satisfies property (4.26) in Theorem 4.14. So by that theorem there is a unique

$$\overline{\operatorname{curry} F\,x} \in ([\mathbb{A}]Y) \to_{\text{fs}} Z$$

satisfying (4.27) and with $\operatorname{supp}(\overline{\operatorname{curry} F\,x}) \subseteq \operatorname{supp}(\operatorname{curry} F\,x)$. Since $\operatorname{curry} F\,x$ is supported by $\mathrm{supp}\, x$ (because $\mathrm{supp}\, F = \emptyset$), it follows that the function $X \times [\mathbb{A}]Y \to Z$ defined by

$$\overline{F} \triangleq \lambda(x, u) \in X \times [\mathbb{A}]Y \to \overline{\mathrm{curry}\, F\, x}\, u$$

satisfies (4.33). The uniqueness property of $\overline{\mathrm{curry}\, F\, x}$ implies that

$$\pi \cdot \overline{\mathrm{curry}\, F\, x} = \overline{\mathrm{curry}\, F\, (\pi \cdot x)}$$

and hence that $\overline{F}$ is equivariant. It is clearly the only function satisfying (4.33), because by the Choose-a-Fresh-Name Principle and Lemma 4.3, given any $(x, u) \in X \times [\mathbb{A}]Y$ there is some $a \in \mathbb{A}$ with $a \mathrel{\#} x$ and $u = \langle a \rangle y$ for some $y \in Y$. □

Example 4.16 In the corollary take $X = 1 = ()$, $Y = \mathbb{A}$, $Z = \mathbb{A} + 1 = \mathbb{A} \cup \{()\}$ and F to be the equivariant function

$$F((), a, a') = \begin{cases} () & \text{if } a = a' \\ a' & \text{if } a \neq a'. \end{cases}$$

In this case condition (4.32) is equivalent to $(\forall a \in \mathbb{A})(\forall a' \in \mathbb{A})\ a \mathrel{\#} F((), a, a')$, which is true. So applying the corollary we get an equivariant function $\overline{F} : [\mathbb{A}]\mathbb{A} \to \mathbb{A} + 1$ satisfying $\overline{F}(\langle a \rangle a') = F(a, a')$ for all $a, a' \in \mathbb{A}$. Since F is surjective, so is $\overline{F}$. It is also injective since if $\overline{F}(\langle a_1 \rangle a'_1) = \overline{F}(\langle a_2 \rangle a'_2)$, then $F(a_1, a'_1) = F(a_2, a'_2)$ and hence either $a_1 = a'_1$ and $a_2 = a'_2$, or $a_1 \neq a'_1 = a'_2 \neq a_2$; and hence by Lemma 4.3, $\langle a_1 \rangle a'_1 = \langle a_2 \rangle a'_2$. Thus $\overline{F}$ gives an isomorphism in **Nom**:

$$[\mathbb{A}]\mathbb{A} \cong \mathbb{A} + 1 \tag{4.34}$$

which using the notation as in (4.31), we can write as

$$\lambda \langle a \rangle a' \in [\mathbb{A}]\mathbb{A} \to \text{if } a = a' \text{ then } () \text{ else } a'.$$

Exercises

4.1 For each nominal set X show that there is an isomorphism making the following diagram commute.

$$\begin{tikzcd} \mathbb{A} * ([\mathbb{A}]X) \arrow[rr, phantom, "\cong"] \arrow[dr, "\mathrm{proj}_1"'] & & \mathbb{A} \times X \arrow[dl, "\mathrm{proj}_1"] \\ & \mathbb{A} & \end{tikzcd}$$

4.2 Given $F : X \to Y$ in **Nom**, regarding each $z \in [\mathbb{A}]X$ as a finitely supported partial function $\mathbb{A} \rightharpoonup_{\mathrm{fs}} X$ as in section 4.2, show that $[\mathbb{A}]F\, z \in [\mathbb{A}]Y$ is equal to the composition $F \circ z = \{(a, y) \in \mathbb{A} \times Y \mid (\exists x \in X)\ (a, x) \in z \wedge F\, x = y\}$. What happens if F is finitely supported, but not equivariant? [Hint: consider the effect of $[\mathbb{A}]\_$ on the finitely supported function $1 \to_{\mathrm{fs}} \mathbb{A}$ corresponding to an element $a \in \mathbb{A}$.]

4.3 Given $X_1, X_2 \in \mathbf{Nom}$, show that there is an isomorphism $I : ([\mathbb{A}]X_1) \times ([\mathbb{A}]X_2) \cong [\mathbb{A}](X_1 \times X_2)$ satisfying

$$I(z_1, z_2) = \mathrm{fresh}\, a \text{ in } \langle a \rangle (z_1 \mathbin{@} a, z_2 \mathbin{@} a)$$

for all $(z_1, z_2) \in ([\mathbb{A}]X_1) \times ([\mathbb{A}]X_2)$.

4.4 Given $X_1, X_2 \in \mathbf{Nom}$, show that there is an isomorphism $J : ([\mathbb{A}]X_1) + ([\mathbb{A}]X_2) \cong [\mathbb{A}](X_1 + X_2)$ satisfying

$$J(\mathrm{inj}_i\, z) = \mathrm{fresh}\, a \text{ in } \langle a \rangle (\mathrm{inj}_1(z \mathbin{@} a))$$

for all $z \in [\mathbb{A}]X_i$ $(i = 1, 2)$.

4.5 Show that the functors $\_ * \mathbb{A}$ and $R\,\_$ from Theorems 4.11 and 4.12 do not give **Nom**-enriched adjoints for $[\mathbb{A}]\_$, because in general

$$(X * \mathbb{A}) \to_{\mathrm{fs}} Y \ncong X \to_{\mathrm{fs}} ([\mathbb{A}]Y)$$
$$([\mathbb{A}]X) \to_{\mathrm{fs}} Y \ncong X \to_{\mathrm{fs}} R\,Y.$$

[Hint: consider $X = 1$ and $Y = \mathbb{B}$.]

5
Nominal Algebraic Data Types

In this chapter we consider initial algebras for functorial constructions on nominal sets that combine products, coproducts and name-abstraction. They give a semantics for languages involving names and name-binding operations where term equality is α-equivalence. This generalizes the classic initial algebra semantics of algebraic data types (Goguen et al., 1977) and associated principles of structural recursion and induction.

5.1 Signatures

In the literature of programming language theory, the syntax of a language is usually specified by an *algebraic signature*. This gives a number of sorts, or 'syntactic categories', into which the terms of the language are divided, together with a number of operations for constructing terms of the various sorts. As well as giving the sorts and the operations, the signature specifies the type of each operation, namely how many arguments each operation takes, what the sorts of those arguments are, and what sort of term the operation constructs. We will use the notation

$$\mathsf{op} : \mathsf{S}_1 \ , \ \cdots \ , \ \mathsf{S}_n \to \mathsf{S} \tag{5.1}$$

to indicate that an operation op constructs a term $\mathsf{op}(t_1 \ , \ \cdots \ , \ t_n)$ of sort S from terms $t_1, \ldots, t_n$ of sorts $\mathsf{S}_1, \ldots, \mathsf{S}_n$. (When $n = 0$ such an operation amounts to a *constant* of sort S.)

The following two related features of languages that occur in practice are not covered by this notion of algebraic signature, but are so common that it is worth enhancing the notion to formalize them.

- *Some syntactic categories are 'sorts of name'.* Language terms of such a sort form an infinite collection of elements whose only attribute, from the point of view of the semantics of the language, is their identity. For example there might be sorts of ‘identifier’ and ‘type variable’ whose terms are represented concretely by strings of alphanumeric characters formed according to specific grammatical rules; but such concrete details are usually irrelevant to the semantics of the language, so we abstract away from such details and just assume there are disjoint infinite sets of names of sort ‘identifier’ and ‘type variable’.

- *Some operations are binders.* That is, the sorts of one or more of their arguments are sorts of name in the above sense; and when the operation is applied to form a term, such an argument (‘binding name’) is linked, in some way that has to be specified for the binding operation, to occurrences of the same name (‘bound names’) elsewhere in the term. The concrete detail of which particular name is used for a binding-bound linkage is usually irrelevant to the semantics of the language. In other words, whatever the meaning of a term, it should be invariant under ‘α-converting’ such a linkage to use another, so far unused name.

To take account of these two features, following (Urban et al., 2004) we will use sorts that are built up from *name-sorts* $\mathtt{N}$ and *data-sorts* $\mathtt{D}$ according to the following grammar.

$$\mathtt{S} ::= \mathtt{N} \mid \mathtt{D} \mid \mathtt{1} \mid \mathtt{S}\ \mathtt{,}\ \mathtt{S} \mid \mathtt{N}\ \mathtt{.}\ \mathtt{S} \tag{5.2}$$

The compound sort $\mathtt{N}\ \mathtt{.}\ \mathtt{S}$ classifies terms that bind a name of sort $\mathtt{N}$ in a scope given by a term of sort $\mathtt{S}$. The compound sort $\mathtt{S}_1\ \mathtt{,}\ \mathtt{S}_2$ classifies terms that are pairs of terms of the indicated sorts. Iterating, we get sorts of the form $(\mathtt{S}_1\ \mathtt{,}\ \cdots)\ \mathtt{,}\ \mathtt{S}_n$ classifying tuples, with the $n = 0$ case given by the unit sort $\mathtt{1}$.

Notation To reduce parentheses in sort expressions, we take $\_\ \mathtt{.}\ \_$ to bind more tightly than $\_\ \mathtt{,}\ \_$ and make the latter associate to the left. Thus for example, $\mathtt{S}_1\ \mathtt{,}\ \mathtt{N}\ \mathtt{.}\ \mathtt{S}_2\ \mathtt{,}\ \mathtt{S}_3$ stands for $(\mathtt{S}_1\ \mathtt{,}\ (\mathtt{N}\ \mathtt{.}\ \mathtt{S}_2))\ \mathtt{,}\ \mathtt{S}_3$.

Definition 5.1 A *nominal algebraic signature* is specified by a set of name-sorts, a set of data-sorts and a set of operations, each of which comes with typing information of the form $\mathtt{op} : \mathtt{S} \to \mathtt{D}$, where $\mathtt{D}$ is a data-sort and $\mathtt{S}$ is a compound sort given by the grammar in (5.2). Given such a signature Σ, fixing disjoint countably infinite sets $\mathbb{A}_{\mathtt{N}}$ of *atomic names of sort* $\mathtt{N}$ (as $\mathtt{N}$ ranges over the signature's name-sorts), the sets $\Sigma(\mathtt{S})$ of *raw terms* of sort S are inductively defined by the following rules.

$$\frac{a \in \mathbb{A}_{\mathtt{N}}}{a \in \Sigma(\mathtt{N})} \qquad \frac{t \in \Sigma(\mathtt{S}) \qquad \mathtt{op} : \mathtt{S} \to \mathtt{D}}{\mathtt{op}\, t \in \Sigma(\mathtt{D})} \qquad \frac{}{() \in \Sigma(\mathtt{1})}$$

$$\frac{t_1 \in \Sigma(\mathtt{S}_1) \qquad t_2 \in \Sigma(\mathtt{S}_2)}{t_1\ \mathtt{,}\ t_2 \in \Sigma(\mathtt{S}_1\ \mathtt{,}\ \mathtt{S}_2)} \qquad \frac{a \in \mathbb{A}_{\mathtt{N}} \qquad t \in \Sigma(\mathtt{S})}{a\ \mathtt{.}\ t \in \Sigma(\mathtt{N}\ \mathtt{.}\ \mathtt{S})}$$

Note that operations of a nominal algebraic signature construct terms of data-sort, but not of name-sort; this is because we wish each name-sort to classify terms that are atomic names without any compound structure. Note also that the usual notion of (many-sorted) algebraic signature can be regarded as the special case of Definition 5.1 in which the set of name-sorts is empty.

Remark We call the elements generated by the above rules *raw* terms to distinguish them from the objects of primary interest, nominal algebraic terms, which are quotients of raw terms by the relation of α-equivalence to be introduced in the next section. Pitts (2006) uses the phrase 'term' for what we call a raw term and 'α-term' for its α-equivalence class.

Example 5.2 A nominal algebraic signature for the untyped λ-calculus (Barendregt, 1984) has a name-sort `Var` for variables, a data-sort `Term` for terms, and operations

$$\begin{aligned}\mathtt{V} &: \mathtt{Var} \to \mathtt{Term}\\ \mathtt{L} &: \mathtt{Var}\ .\ \mathtt{Term} \to \mathtt{Term}\\ \mathtt{A} &: \mathtt{Term}\ ,\ \mathtt{Term} \to \mathtt{Term}.\end{aligned}$$

For example, if x and y are atomic names of sort `Var`, then the raw term $\mathtt{L}(x\,.\,\mathtt{A}(\mathtt{V}\,x\,,\ \mathtt{V}\,y))$ represents the λ-term $\lambda x.x\,y$.

Example 5.3 A nominal algebraic signature for the π-calculus (Sangiorgi and Walker, 2001, Definition 1.1.1) has a name-sort `Chan` for channel names, data-sorts `Proc`, `Sum` and `Pre` for processes, summations and prefixed processes, and operations

$$\begin{aligned}\mathtt{S} &: \mathtt{Sum} \to \mathtt{Proc}\\ \mathtt{Comp} &: \mathtt{Proc}\ ,\ \mathtt{Proc} \to \mathtt{Proc}\\ \mathtt{Nu} &: \mathtt{Chan}\ .\ \mathtt{Proc} \to \mathtt{Proc}\\ \mathtt{!} &: \mathtt{Proc} \to \mathtt{Proc}\\ \mathtt{Out} &: \mathtt{Chan}\ ,\ \mathtt{Chan}\ ,\ \mathtt{Proc} \to \mathtt{Proc}\\ \mathtt{In} &: \mathtt{Chan}\ ,\ \mathtt{Chan}\ .\ \mathtt{Proc} \to \mathtt{Proc}\\ \mathtt{P} &: \mathtt{Pre} \to \mathtt{Sum}\\ \mathtt{O} &: 1 \to \mathtt{Sum}\\ \mathtt{Plus} &: \mathtt{Sum}\ ,\ \mathtt{Sum} \to \mathtt{Sum}\\ \mathtt{Tau} &: \mathtt{Proc} \to \mathtt{Pre}\\ \mathtt{Match} &: \mathtt{Chan}\ ,\ \mathtt{Chan}\ ,\ \mathtt{Pre} \to \mathtt{Pre}.\end{aligned}$$

Thus there are two operations involving binding, `Nu` and `In`. Assuming x and z are atomic names of sort Chan and that P is a raw term of sort Proc, then the raw terms $\mathtt{Nu}(x \;\mathtt{.}\; P)$ and $\mathtt{Out}(x \;\mathtt{,}\; (z \;\mathtt{.}\; P))$ of sort Proc represent restricted and input-prefixed processes that are written in the π-calculus as $\nu x\, P$ and $x(z).P$ respectively.

5.2 α-Equivalence

In a nominal algebraic signature, sorts of the form N . S are used in the type of an operation to indicate where it binds names. From this typing information one can generate a version of α-equivalence that identifies raw terms up to renaming such bound names. We use the theory of nominal sets to define this equivalence relation and develop it properties. In general this requires a 'many-sorted' extension of the theory of nominal sets to cope with the fact that a nominal algebraic signature may have many name-sorts; see Pitts (2006). For simplicity's sake, we will restrict attention to signatures with at most one name-sort; the general case is treated in (Pitts, 2006). We will also only consider nominal algebraic signatures that are *finite*, in the sense of having only finitely many data-sorts and operations.

Let Σ be such a finite nominal algebraic signature with a single name-sort N. We use elements of the fixed set $\mathbb{A}$ as the atomic names of sort N. For each sort S, the set $\Sigma(\mathtt{S})$ of raw terms of sort S possesses a Perm $\mathbb{A}$-action, extending the definition of the Perm $\mathbb{A}$-action for ordinary algebraic terms given in Example 1.4. It is defined for all sorts simultaneously as follows.

$$\begin{aligned}
\pi \cdot a &= \pi\, a \\
\pi \cdot \mathsf{op}\, t &= \mathsf{op}(\pi \cdot t) \\
\pi \cdot () &= () \\
\pi \cdot (t_1 \;\mathtt{,}\; t_2) &= \pi \cdot t_1 \;\mathtt{,}\; \pi \cdot t_2 \\
\pi \cdot (a \;\mathtt{.}\; t) &= (\pi\, a) \;\mathtt{.}\; (\pi \cdot t).
\end{aligned} \tag{5.3}$$

One can prove by induction on the structure of t that

- $t \in \Sigma(\mathtt{S})$ implies $\pi \cdot t \in \Sigma(\mathtt{S})$;
- $(\pi, t) \mapsto \pi \cdot t$ has the properties (1.8) and (1.9) required of an action;
- with respect to this action, each $t \in \Sigma(\mathtt{S})$ is strongly supported (*cf.* Theorem 2.7) by the finite set of atomic names occurring in any position in t.

Thus each $\Sigma(\mathsf{S})$ is a nominal set with:

$$\begin{aligned} \operatorname{supp} a &= \{a\} \\ \operatorname{supp}(\mathsf{op}\, t) &= \operatorname{supp} t \\ \operatorname{supp}() &= \emptyset \\ \operatorname{supp}(t_1 \mathbin{,} t_2) &= \operatorname{supp} t_1 \cup \operatorname{supp} t_2 \\ \operatorname{supp}(a \mathbin{.} t) &= \{a\} \cup \operatorname{supp} t. \end{aligned} \tag{5.4}$$

The way we defined α-equivalence for untyped λ-terms in section 2.9 extends in a straightforward way from the signature in Example 5.2 to arbitrary nominal algebraic signatures.

Definition 5.4 Given a nominal algebraic signature Σ (with single name sort), the binary relations of *α-equivalence* $=_\alpha \subseteq \Sigma(\mathsf{S}) \times \Sigma(\mathsf{S})$ for each sort S, are simultaneously inductively defined by the following rules.

$$\frac{a \in \mathbb{A}}{a =_\alpha a} \qquad \frac{t =_\alpha t'}{\mathsf{op}\, t =_\alpha \mathsf{op}\, t'} \qquad \frac{}{() =_\alpha ()}$$

$$\frac{t_1 =_\alpha t_1' \qquad t_2 =_\alpha t_2'}{t_1 \mathbin{,} t_2 =_\alpha t_1' \mathbin{,} t_2'} \qquad \frac{(a_1\ a) \cdot t_1 =_\alpha (a_2\ a) \cdot t_2 \qquad a \mathbin{\#} (a_1, t_1, a_2, t_2)}{a_1 \mathbin{.} t_1 =_\alpha a_2 \mathbin{.} t_2}$$

The first four of these rules express congruence properties of $=_\alpha$, while the fifth one combines a congruence property

$$t_1 =_\alpha t_2 \Rightarrow a \mathbin{.} t_1 =_\alpha a \mathbin{.} t_2 \tag{5.5}$$

with a renaming property (*cf.* Lemma 4.3)

$$a \mathbin{\#} (a_1, t_1) \Rightarrow a_1 \mathbin{.} t_1 =_\alpha a \mathbin{.} (a_1\ a) \cdot t_1. \tag{5.6}$$

Both (5.5) and (5.6) are simple consequences of the definition of $=_\alpha$ and the fact that it is equivariant. Indeed we have the following result.

Lemma 5.5 *The relations $=_\alpha$ are equivariant equivalence relations.*

Proof Equivariance of $=_\alpha$ is a consequence of the Equivariance Principle. The fact that $=_\alpha$ is reflexive and symmetric is immediate from its definition. To prove transitivity it suffices to show that the sets

$$H_{\mathsf{S}} \triangleq \{(t_1, t_2) \in \Sigma(\mathsf{S}) \times \Sigma(\mathsf{S}) \mid (\forall t \in \Sigma(\mathsf{S}))\ t_2 =_\alpha t \Rightarrow t_1 =_\alpha t\} \quad (\mathsf{S} \text{ a sort of } \Sigma)$$

are closed under the rules in Definition 5.4 inductively defining $=_\alpha$. For then $=_\alpha$ is contained in H_{S} and hence is transitive. Closure of H_{S} under the first four rules is straightforward. For the fifth rule, suppose

$$((a_1\ a) \cdot t_1, (a_2\ a) \cdot t_2) \in H_{\mathsf{S}} \tag{5.7}$$

with $a \# (a_1, t_1, a_2, t_2)$. We have to show that $(a_1 \,.\, t_1, a_2 \,.\, t_2) \in H_{\mathtt{N.S}}$. If $a_2 \,.\, t_2 =_\alpha t$, then the syntax-directed nature of the rules defining $=_\alpha$ implies that this instance of α-equivalence must have been deduced by an application of the fifth rule; in other words $t = a' \,.\, t'$ for some $a' \in \mathbb{A}$ and $t' \in \Sigma(\mathsf{S})$, and there is some $a'' \# (a_2, t_2, a', t')$ with

$$(a_2\ a'') \cdot t_2 =_\alpha (a'\ a'') \cdot t'. \tag{5.8}$$

Use the Choose-a-Fresh-Name Principle to pick some $a''' \in \mathbb{A}$ with

$$a''' \# (a_1, t_1, a_2, t_2, a, a', t', a'').$$

Note that H_{S} is equivariant, because $=_\alpha$ is. So applying $(a\ a''') \cdot \_$ to (5.7) we get $((a_1\ a''')\cdot t_1, (a_2\ a''')\cdot t_2) \in H_{\mathsf{S}}$; and applying $(a''\ a''')\cdot\_$ to (5.8) we get $(a_2\ a''')\cdot t_2 =_\alpha (a'\ a''') \cdot t'$. So by definition of H_{S}, we have $(a_1\ a''') \cdot t_1 =_\alpha (a'\ a''') \cdot t'$; and since $a''' \# (a_1, t_1, a', t')$, we can apply the fifth rule in Definition 5.4 to conclude that $a_1 \,.\, t_1 =_\alpha a' \,.\, t' = t$. Therefore we do indeed have $(a_1 \,.\, t_1, a_2 \,.\, t_2) \in H_{\mathtt{N.S}}$. □

Remark Note that $=_\alpha$ is a decidable relation (given a suitable Gödel numbering of raw terms). A decision procedure is implicit in the inductive definition, because of the syntax-directed nature of the rules in Definition 5.4 and the fact that the size (number of symbols) of raw terms goes down reading the rules for compound terms bottom-up. (Note that the size function is equivariant: the size of $\pi \cdot t$ is equal to the size of t for any $\pi \in \operatorname{Perm} \mathbb{A}$.)

Definition 5.6 For each sort S of a nominal algebraic signature Σ, the elements of the quotient nominal set

$$\Sigma_\alpha(\mathsf{S}) \triangleq \Sigma(\mathsf{S})/{=_\alpha} \tag{5.9}$$

are called *nominal algebraic terms* of sort S. We call nominal sets of the form $\Sigma_\alpha(\mathsf{S})$ *nominal algebraic data types.*

Proposition 5.7 *For each sort* S *of a nominal algebraic signature* Σ*, the least support* $\operatorname{supp} e$ *of a nominal algebraic term* $e \in \Sigma_\alpha(\mathsf{S})$ *is equal to the finite set* $\operatorname{fn} t$ *of* free atomic names *of any representative raw term* t*:*

$$e = [t]_{=_\alpha} \Rightarrow \operatorname{supp} e = \operatorname{fn} t \tag{5.10}$$

where

$$\begin{aligned} \mathrm{fn}\, a &= \{a\} \\ \mathrm{fn}(\mathtt{op}\, t) &= \mathrm{fn}\, t \\ \mathrm{fn}() &= \emptyset \\ \mathrm{fn}(t_1 \mathbin{\texttt{,}} t_2) &= \mathrm{fn}\, t_1 \cup \mathrm{fn}\, t_2 \\ \mathrm{fn}(a \mathbin{\texttt{.}} t) &= (\mathrm{fn}\, t) - \{a\}. \end{aligned} \tag{5.11}$$

Proof One can prove that $\mathrm{fn}\, t$ strongly supports $[t]_{=_\alpha}$ in exactly the same way as for the special case of λ-terms in section 2.9. Hence by Theorem 2.7, $\mathrm{supp}([t]_{=_\alpha}) = \mathrm{fn}\, t$. □

When working with languages involving binders, it is common to use notation that blurs the distinction between a raw term t and the α-equivalence class $[t]_{=_\alpha}$ it determines. Accordingly, we use the following notation for nominal algebraic terms.

Definition 5.8

$$\mathtt{op}\, e \triangleq [\mathtt{op}\, t]_{=_\alpha} \quad \text{where } e = [t]_{=_\alpha} \tag{5.12}$$

$$e_1 \mathbin{\texttt{,}} e_2 \triangleq [t_1 \mathbin{\texttt{,}} t_2]_{=_\alpha} \quad \text{where } e_1 = [t_1]_{=_\alpha} \text{ and } e_2 = [t_2]_{=_\alpha} \tag{5.13}$$

$$a \mathbin{\texttt{.}} e \triangleq [a \mathbin{\texttt{.}} t] \quad \text{where } e = [t]_{=_\alpha}. \tag{5.14}$$

We will also write $[a]_{=_\alpha}$ just as a and $[()]_{=_\alpha}$ just as $()$ when it is clear from the context that we are referring to nominal algebraic terms.

These notational conventions are justified by the following properties of $=_\alpha$, which are simple consequences of its definition and of Lemma 5.5.

- For each operation $op : \mathtt{S} \to \mathtt{D}$ there is a function $\Sigma_\alpha(\mathtt{S}) \to \Sigma_\alpha(\mathtt{D})$, well-defined by $[t]_{=_\alpha} \mapsto [\mathtt{op}\, t]_{=_\alpha}$.
- Given sorts $\mathtt{S}_1$ and $\mathtt{S}_2$, there is a function $\Sigma_\alpha(\mathtt{S}_1) \times \Sigma_\alpha(\mathtt{S}_2) \to \Sigma_\alpha(\mathtt{S}_1 \mathbin{\texttt{,}} \mathtt{S}_2)$, well-defined by $([t_1]_{=_\alpha}, [t_2]_{=_\alpha}) \mapsto [t_1 \mathbin{\texttt{,}} t_2]_{=_\alpha}$.
- For each sort $\mathtt{S}$ there is a function $\mathbb{A} \times \Sigma_\alpha(\mathtt{S}) \to \Sigma_\alpha(\mathtt{N} \mathbin{\texttt{.}} \mathtt{S})$, well-defined by $(a, [t]_{=_\alpha}) \mapsto [a \mathbin{\texttt{.}} t]_{=_\alpha}$.
- For each $a \in \mathbb{A}$, the equivalence class $[a]_{=_\alpha}$ is the singleton $\{a\}$.
- The equivalence class $[()]_{=_\alpha}$ is the singleton $\{()\}$.

5.3 Algebraic functors

To each finite nominal algebraic signature Σ, with a single name-sort $\mathtt{N}$ and n data-sorts $\mathtt{D}_1, \ldots, \mathtt{D}_n$, we associate a functor $\mathrm{T} : \mathbf{Nom}^n \to \mathbf{Nom}^n$. To do so we first re-organize the typing information for a signature's operations so as to present, for each data-sort, the different ways of constructing terms of that sort. Thus

$$\begin{aligned} \mathtt{D}_1 &= \mathtt{op}_{1,1}(\mathtt{S}_{1,1}) \mid \cdots \mid \mathtt{op}_{1,m_1}(\mathtt{S}_{1,m_1}) \\ &\vdots \\ \mathtt{D}_n &= \mathtt{op}_{n,1}(\mathtt{S}_{n,1}) \mid \cdots \mid \mathtt{op}_{n,m_n}(\mathtt{S}_{n,m_n}). \end{aligned} \tag{5.15}$$

is the signature with operations $\mathtt{op}_{i,j} : \mathtt{S}_{i,j} \to D_i$ ($i = 1..n$, $j = 1..m_i$). For example, the signature in Example 5.3 written in this way looks like this:

$$\begin{aligned} \mathtt{Proc} &= \mathtt{S(Sum)} \mid \mathtt{Comp(Proc\ ,\ Proc)} \mid \mathtt{Nu(Chan\ .\ Proc)} \mid \mathtt{!(Proc)} \mid \\ &\quad\ \mathtt{Out(Chan\ ,\ Chan\ ,\ Proc)} \mid \mathtt{In(Chan\ ,\ Chan\ .\ Proc)} \\ \mathtt{Sum} &= \mathtt{P(Pre)} \mid \mathtt{O(1)} \mid \mathtt{Plus(Sum\ ,\ Sum)} \\ \mathtt{Pre} &= \mathtt{Tau(Proc)} \mid \mathtt{Match(Chan\ ,\ Chan\ ,\ Pre)}. \end{aligned}$$

The sorts $\mathtt{S}_{i,j}$ in (5.15) are built up from $\mathtt{D}_1, \ldots, \mathtt{D}_n$ and the name-sort $\mathtt{N}$ as in (5.2). Given an n-tuple of nominal sets $X = (X_1, \ldots, X_n) \in \mathbf{Nom}^n$, each such sort $\mathtt{S}$ gives rise to a nominal set $[\![\mathtt{S}]\!]X$, defined by recursion on the structure of S as follows.

$$\begin{aligned} [\![\mathtt{N}]\!]X &= \mathbb{A} \\ [\![\mathtt{D}_i]\!]X &= X_i \\ [\![\mathtt{1}]\!]X &= 1 \\ [\![\mathtt{S}_1\ ,\ \mathtt{S}_2]\!]X &= [\![\mathtt{S}_1]\!]X \times [\![\mathtt{S}_2]\!]X \\ [\![\mathtt{N}\ .\ \mathtt{S}]\!]X &= [\mathbb{A}]([\![\mathtt{S}]\!]X). \end{aligned} \tag{5.16}$$

The mapping $X \mapsto [\![\mathtt{S}]\!]X$ extends to a functor $[\![\mathtt{S}]\!] : \mathbf{Nom}^n \to \mathbf{Nom}$ using the functoriality of products and of name abstraction (section 4.3).

Definition 5.9 If Σ is the signature given by (5.15), then the associated functor $\mathrm{T} : \mathbf{Nom}^n \to \mathbf{Nom}^n$ has components $\mathrm{T}_i : \mathbf{Nom}^n \to \mathbf{Nom}$ (for $i = 1..n$) given by mapping each $X = (X_1, \ldots, X_n) \in \mathbf{Nom}^n$ to

$$\mathrm{T}_i\, X \triangleq [\![\mathtt{S}_{i,1}]\!]X + \cdots + [\![\mathtt{S}_{i,m_i}]\!]X \tag{5.17}$$

and similarly for n-tuples of equivariant functions. We call functors that arise in this way *nominal algebraic functors*.

Example 5.10 If Σ is the signature from Example 5.2, then the associated functor $\mathbf{Nom} \to \mathbf{Nom}$ maps a nominal set X to $\mathbb{A} + ([\mathbb{A}]X) + (X \times X)$. Whereas for the signature in Example 5.3, the associated functor $\mathbf{Nom}^3 \to \mathbf{Nom}^3$ maps a triple of nominal sets (X_1, X_2, X_3) to the triple whose components are

$$\begin{aligned} & X_2 + (X_1 \times X_1) + ([\mathbb{A}]X_1) + (\mathbb{A} \times \mathbb{A} \times X_1) + (\mathbb{A} \times ([\mathbb{A}]X_1)),\\ & X_3 + 1 + (X_2 \times X_2)\\ \text{and}\quad & X_1 + (\mathbb{A} \times \mathbb{A} \times X_3). \end{aligned}$$

In the next section we will need to use the fact that the action of nominal algebraic functors on equivariant functions extends to one on all finitely supported functions. Given $X = (X_1 \ldots, X_n)$ and $Y = (Y_1 \ldots, Y_n)$ in $\mathbf{Nom}^n$, define

$$X \rightarrow^n_{\mathrm{fs}} Y \triangleq (X_1 \rightarrow_{\mathrm{fs}} Y_1) \times \cdots \times (X_n \rightarrow_{\mathrm{fs}} Y_n). \tag{5.18}$$

This is the hom-object in **Nom** for the the **Nom**-enriched category $\mathbf{Nom}^n$. We noted in section 2.4 that the functor $[\mathbb{A}]\_ : \mathbf{Nom} \rightarrow \mathbf{Nom}$ is **Nom**-enriched; and the same is true for the product and coproduct functors for more standard reasons. Using these we get a **Nom**-enrichment for each T. Concretely this means that there are equivariant functions

$$(X \rightarrow^n_{\mathrm{fs}} Y) \rightarrow (\mathrm{T}X \rightarrow^n_{\mathrm{fs}} \mathrm{T}Y) \quad (X, Y \in \mathbf{Nom}^n) \tag{5.19}$$

that preserve identity and composition and agree with the application of T to equivariant functions (recalling that these are the elements of $X \rightarrow^n_{\mathrm{fs}} Y$ with empty support). Following (5.17), the functions in (5.19) are defined using coproduct (disjoint union) from **Nom**-enrichments

$$(X \rightarrow^n_{\mathrm{fs}} Y) \rightarrow ([\![\mathtt{S}]\!]X \rightarrow_{\mathrm{fs}} [\![\mathtt{S}]\!]Y) \quad (X, Y \in \mathbf{Nom}^n) \tag{5.20}$$

for the functors $[\![\mathtt{S}]\!] : \mathbf{Nom}^n \rightarrow \mathbf{Nom}$; and following (5.16), these are defined by recursion on the structure of the sort S using the enrichments for product and name abstraction. Thus given $F \in X \rightarrow^n_{\mathrm{fs}} Y$, the function $[\![\mathtt{S}]\!]\, F \in [\![\mathtt{S}]\!]X \rightarrow_{\mathrm{fs}} [\![\mathtt{S}]\!]Y$ has the following properties according to the structure of S:

$$\begin{aligned} [\![\mathtt{N}]\!]\, F\, a &= a\\ [\![\mathtt{S}_i]\!]\, F\, d &= F_i\, d\\ [\![\mathtt{1}]\!]\, F\, () &= ()\\ [\![\mathtt{S}_1\ \mathtt{,}\ \mathtt{S}_2]\!]\, F\, (d_1, d_2) &= ([\![\mathtt{S}_1]\!]\, F\, d_1, [\![\mathtt{S}_2]\!]\, F\, d_2)\\ a \mathrel{\#} F \Rightarrow [\![\mathtt{N}\ .\ \mathtt{S}]\!]\, F\, (\langle a \rangle d) &= \langle a \rangle ([\![\mathtt{S}]\!]\, F\, d) \end{aligned} \tag{5.21}$$

where the last, conditional equation uses the functorial action of $[\mathbb{A}]\_$ on finitely supported functions (4.12).

5.4 Initial algebra semantics

Let Σ be the signature given by (5.15) and define $D \in \mathbf{Nom}^n$ to be

$$D \triangleq (\Sigma_\alpha(\mathtt{D}_1), \ldots, \Sigma_\alpha(\mathtt{D}_1)). \tag{5.22}$$

We can define equivariant functions $I_{\mathtt{S}} : \llbracket \mathtt{S} \rrbracket D \to \Sigma_\alpha(\mathtt{S})$ by recursion on the structure of sorts S as follows (using the notational conventions of Definition 5.8 and, in the last clause, the definition (4.13) of the action of $[\mathbb{A}]\_$ on equivariant functions).

$$\begin{aligned} I_{\mathtt{N}}\, a &= a \\ I_{\mathtt{D}_i}\, e &= e \\ I_{\mathtt{1}}\, () &= () \\ I_{\mathtt{S}_1\,,\,\mathtt{S}_2}(d_1, d_2) &= I_{\mathtt{S}_1}\, d_1 \;\mathbf{,}\; I_{\mathtt{S}_2}\, d_2 \\ I_{\mathtt{N}\,.\,\mathtt{S}}(\langle a \rangle d) &= a \;\mathbf{.}\; I_{\mathtt{S}}\, d \end{aligned} \tag{5.23}$$

If $\mathrm{T} : \mathbf{Nom}^n \to \mathbf{Nom}^n$ is the functor associated with Σ as in Definition 5.9, then using these functions for each $i = 1..n$ we get an equivariant function $I_i : \mathrm{T}_i\, D = \llbracket \mathtt{S}_{i,1} \rrbracket D + \cdots + \llbracket \mathtt{S}_{i,m_i} \rrbracket D \to \Sigma_\alpha(\mathtt{D}_i)$, given by

$$I_i(\mathrm{inj}_j\, d) = \mathtt{op}_{i,j}(I_{\mathtt{S}_{i,j}}\, d) \quad (j = 1..m_i,\ d \in \llbracket \mathtt{S}_{i,j} \rrbracket D). \tag{5.24}$$

So altogether we get a morphism in $\mathbf{Nom}^n$:

$$I \triangleq (I_1, \ldots, I_n) : \mathrm{T}\, D \to D. \tag{5.25}$$

We will show that this gives an *initial* T*-algebra* for the functor $\mathrm{T} : \mathbf{Nom}^n \to \mathbf{Nom}^n$. Thus given any T-algebra, that is, any morphism $F : \mathrm{T}\, X \to X$, there is a unique morphism $\hat{F} : D \to X$ making the following diagram commute.

$$\begin{tikzcd} \mathrm{T}\, D \arrow[r, dashed, "\mathrm{T}\hat{F}"] \arrow[d, "I"'] & \mathrm{T}\, X \arrow[d, "F"] \\ D \arrow[r, dashed, "\hat{F}"'] & X \end{tikzcd} \tag{5.26}$$

(In particular, I is an isomorphism: see Exercise 5.2.)

In general, such a universal property is of interest because it gives rise to recursion principles for the initial algebra. In this case, in order to capture some common informal uses of recursion in the presence of α-equivalence, we need to establish a stronger version of (5.26), one in which equivariant functions are generalized to finitely supported functions. The following familiar example illustrates the need for this.

Example 5.11 Let Σ be the signature for untyped λ-calculus from Example 5.2. In this case $D = \Sigma_\alpha(\mathtt{Term})$ is the nominal set of λ-terms modulo α-equivalence from section 2.9; the functor $\mathrm{T} : \mathbf{Nom} \to \mathbf{Nom}$ is $\mathbb{A} + ([\mathbb{A}]\_) + (\_ \times \_)$; and the morphism $I : \mathbb{A} + [\mathbb{A}]D + D \times D \to D$ satisfies

$$\begin{aligned} I(\mathrm{inj}_1\, a) &= \mathtt{V}\, a \\ I(\mathrm{inj}_2(\langle a\rangle e)) &= \mathtt{L}\, a \mathrel{\mathtt{.}} e \\ I(\mathrm{inj}_3(e_1, e_2)) &= \mathtt{A}(e_1 \mathrel{\mathtt{,}} e_2). \end{aligned} \tag{5.27}$$

A T-algebra is given by equivariant functions $F_1 : \mathbb{A} \to X$, $F_2 : [\mathbb{A}]X \to X$ and $F_3 : X \times X \to X$. The unique $\hat{F} : D \to X$ making (5.26) commute satisfies the following recursion equations.

$$\begin{aligned} \hat{F}(\mathtt{V}\, a_1) &= F_1\, a_1 \\ \hat{F}(\mathtt{L}\, a_1 \mathrel{\mathtt{.}} e_1) &= F_2(\langle a_1\rangle(\hat{F}\, e_1)) \\ \hat{F}(\mathtt{A}(e_1 \mathrel{\mathtt{,}} e_2)) &= F_3(\hat{F}\, e_1, \hat{F}\, e_2). \end{aligned} \tag{5.28}$$

Contrast this with the operation $(a := e) : D \to D$ of *capture-avoiding substitution* of a λ-term e for all free occurrences of a variable $\mathtt{V}\, a$ in a λ-term. Being finitely supported by $\{a\} \cup \mathrm{fv}\, e$, this operation cannot be an instance of the above initial T-algebra property (with $X = \Sigma_\alpha(\mathtt{Term})$), since that produces an emptily supported function $\hat{F}$ from emptily supported functions (F_1, F_2, F_3). Nevertheless, it has a recursive specification that is quite similar to (5.28):

$$\begin{aligned} (a := e)(\mathtt{V}\, a_1) &= F_1\, a_1 \\ a_1 \notin \{a\} \cup \mathrm{fv}\, e \Rightarrow (a := e)(\mathtt{L}\, a_1 \mathrel{\mathtt{.}} e_1) &= F_2(\langle a_1\rangle((a := e)e_1)) \\ (a := e)(\mathtt{A}(e_1 \mathrel{\mathtt{,}} e_2)) &= F_3((a := e)e_1, (a := e)e_2) \end{aligned} \tag{5.29}$$

where $F_1 \in \mathbb{A} \to_{\mathrm{fs}} D$ is $\lambda a_1 \in \mathbb{A} \to \text{if}\, a_1 = a \text{ then } e \text{ else } \mathtt{V}\, a_1$ (which has support $\{a\} \cup \mathrm{fv}\, e$), and where F_2 and F_3 are the equivariant functions $I \circ \mathrm{inj}_2$ and $I \circ \mathrm{inj}_3$ from (5.27).

The middle clause in (5.29) is the essence of the capture-avoiding aspect of this form of substitution: it is only necessary to say how to unwind the recursive definition for sufficiently fresh bound variables a_1. Since $(a := e)$ is uniquely determined by (F_1, F_2, F_3) and the latter has support $\{a\} \cup \mathrm{fv}\, e$, this middle clause can be rephrased using a freshness quantifier

$$(\text{И} a_1)\ (a := e)(\mathtt{L}\, a_1 \mathrel{\mathtt{.}} e_1) = F_2(\langle a_1\rangle((a := e)e_1)).$$

It turns out that this use of the freshness quantifier corresponds exactly to the way it occurs in the definition of the action of the functor $[\mathbb{A}]\_$ on finitely supported functions, as in (4.12). Thus the recursive definition of $(a := e)$ is an instance of the following initial algebra property of nominal data types.

Theorem 5.12 (Initial algebra theorem for nominal algebraic data types) *Let*

Σ be a nominal algebraic signature with a single name-sort $\mathtt{N}$, with n data-sorts $\mathtt{D}_1, \ldots, \mathtt{D}_n$, and with operations as shown in (5.15)*. The n-tuple of nominal algebraic data types*

$$D = (\Sigma_\alpha(\mathtt{D}_1), \ldots, \Sigma_\alpha(\mathtt{D}_n))$$

equipped with the morphism (5.25) *is an initial algebra for the* **Nom**-*enriched functor* $\mathrm{T} : \mathbf{Nom}^n \to \mathbf{Nom}^n$ *associated with* Σ *as in section 5.3. In other words, for each* $X \in \mathbf{Nom}^n$ *and* $F \in \mathrm{T}X \to^n_{\mathrm{fs}} X$ *there is a unique* $\hat{F} \in D \to^n_{\mathrm{fs}} X$ *satisfying*

$$F \circ (\mathrm{T}\hat{F}) = \hat{F} \circ I. \tag{5.30}$$

Moreover, $\operatorname{supp} \hat{F} \subseteq \operatorname{supp} F$.

Remark The fact that any nominal algebraic functor T has a **Nom**-enriched initial algebra follows from general, category-theoretic considerations. It can be constructed by taking the colimit of the countable chain $\emptyset \to \mathrm{T}\emptyset \to \mathrm{T}(\mathrm{T}\emptyset) \to \cdots$ and using the fact that the **Nom**-enriched functor T preserves such colimits. So the force of the theorem is that this initial algebra can be presented in terms of the sets of nominal algebraic terms associated with the signature.

Proof of existence of $\hat{F}$ Define relations $\overline{F}_{\mathtt{S}} \subseteq \Sigma(\mathtt{S}) \times \llbracket \mathtt{S} \rrbracket X$, as $\mathtt{S}$ ranges over sorts, simultaneously inductively by the following rules.

$$\begin{gathered}
\frac{a \in \mathbb{A}}{(a, a) \in \overline{F}_{\mathtt{N}}} \qquad \frac{(t, d) \in \overline{F}_{\mathtt{S}_{i,j}} \qquad i \in \{1..n\} \quad j \in \{1..m_i\}}{(\mathsf{op}_{i,j}\, t, F_i(\mathrm{inj}_j\, d)) \in \overline{F}_{\mathtt{D}_i}} \\
\frac{}{((), ()) \in \overline{F}_{1}} \qquad \frac{(t_1, d_1) \in \overline{F}_{\mathtt{S}_1} \qquad (t_2, d_2) \in \overline{F}_{\mathtt{S}_2}}{(t_1 \;\mathbf{,}\; t_2, (d_1, d_2)) \in \overline{F}_{\mathtt{S}_1, \mathtt{S}_2}} \\
\frac{((a_1\ a) \cdot t, (a_2\ a) \cdot d) \in \overline{F}_{\mathtt{S}} \qquad a \mathrel{\#} (a_1, t, a_2, d, F)}{(a_1 \;\mathbf{.}\; t, \langle a_2 \rangle d) \in \overline{F}_{\mathtt{N.S}}}
\end{gathered} \tag{5.31}$$

These relations have the following properties.

1. Each $\overline{F}_{\mathtt{S}}$ is supported by $\operatorname{supp} F$ as a subset of the nominal set $\Sigma(\mathtt{S}) \times \llbracket \mathtt{S} \rrbracket X$, because the set of rules inductively defining these subsets is supported by this finite set.
2. The relations respect α-equivalence in their first components and are single-valued:

$$(t, d) \in \overline{F}_{\mathtt{S}} \wedge t =_\alpha t' \Rightarrow (t', d) \in \overline{F}_{\mathtt{S}} \tag{5.32}$$

$$(t, d) \in \overline{F}_{\mathtt{S}} \wedge (t, d') \in \overline{F}_{\mathtt{S}} \Rightarrow d = d'. \tag{5.33}$$

These are proved, simultaneously for all sorts $\mathtt{S}$, by induction on the derivation of $(t, d) \in \overline{F}_{\mathtt{S}}$ from the rules in (5.32). The only interesting induction step is for name abstractions, which uses a typical 'some/any' argument to replace the atomic name a in the hypothesis of the last rule in (5.32) by one that is also fresh for some $a' \mathbin{.} t'$ (for the first property), or for some $\langle a' \rangle d'$ (for the second property).

3. The relations are total:

$$(\forall t \in \Sigma(\mathtt{S}))(\exists d \in [\![\mathtt{S}]\!]X)\ (t, d) \in \overline{F}_{\mathtt{S}}. \tag{5.34}$$

For this we prove a slightly stronger property, namely that the sort-indexed family of subsets

$$H_{\mathtt{S}} \triangleq \{t \in \Sigma(\mathtt{S}) \mid (\forall \pi \in \operatorname{Perm} \mathbb{A})(\exists d \in [\![\mathtt{S}]\!]X)\ (\pi \cdot t, d) \in \overline{F}_{\mathtt{S}}\}$$

is closed under the rules in Definition 5.1 inductively defining the sets $\Sigma(\mathtt{S})$ of raw terms of each sort, and hence that $H_{\mathtt{S}} = \Sigma(\mathtt{S})$. (The quantification over all finitary permutations $\pi \in \operatorname{Perm} \mathbb{A}$ in the definition of $H_{\mathtt{S}}$ is there to ensure that these subsets are equivariant, despite the fact that $\overline{F}_{\mathtt{S}}$ may have non-empty support; indeed, it is easy to see from the definition that we have $t \in H_{\mathtt{S}} \Rightarrow \pi \cdot t \in H_{\mathtt{S}}$.) The only non-trivial induction step is for closure of the subsets under formation of raw terms of the form $a \mathbin{.} t$, which is proved as follows.

Suppose $t \in H_{\mathtt{S}}$ and $a \in \mathbb{A}$; we prove that $a \mathbin{.} t \in H_{\mathtt{N.S}}$. Given any $\pi \in \operatorname{Perm} \mathbb{A}$ we can use the Choose-a-Fresh-Name Principle to pick some $a' \mathrel{\#} (a, \pi, t, F)$. Since $t \in H_{\mathtt{S}}$, there exists $d \in [\![\mathtt{S}]\!]X$ with

$$((\pi\, a\ a') \cdot \pi \cdot t, d) \in \overline{F}_{\mathtt{S}}. \tag{5.35}$$

Now pick some $a'' \mathrel{\#} (a, \pi, t, F, a', d)$; applying $(a\ a')$ to (5.35) and using the fact that $a', a'' \notin \operatorname{supp} F \supseteq \sup \overline{F}$, we get $((\pi\, a\ a'') \cdot \pi \cdot t, (a'\ a'') \cdot d) \in \overline{F}_{\mathtt{S}}$. Hence by definition of $\overline{F}$, $((\pi\, a) \mathbin{.} (\pi \cdot t), \langle a' \rangle d) \in \overline{F}_{\mathtt{N.S}}$. Therefore $a \mathbin{.} t \in H_{\mathtt{N.S}}$, as required.

In view of properties 1–3, for each sort $\mathtt{S}$ the relation $\overline{F}_{\mathtt{S}}$ induces a function from $\Sigma_\alpha(\mathtt{S})$ to $[\![\mathtt{S}]\!]X$ that is supported by $\operatorname{supp} F$. Taking $\mathtt{S}$ to be $\mathtt{D}_i$ this gives us a function $\hat{F}_i \in \Sigma_\alpha(\mathtt{D}_i) \to_{\mathrm{fs}} X_i$ satisfying

$$(\forall t \in \Sigma(\mathtt{D}_i))\ (t, \hat{F}_i[t]_{=_\alpha}) \in \overline{F}_{\mathtt{D}_i}. \tag{5.36}$$

So we have constructed $\hat{F} = (\hat{F}_1, \ldots, \hat{F}_n) \in D \to^n_{\mathrm{fs}} X$ supported by $\operatorname{supp} F$ and it remains to prove that it satisfies (5.30), that is, $F_i \circ (\mathrm{T}_i \hat{F}) = \hat{F}_i \circ I_i$ holds for $i = 1..n$. From the definitions of T_i and I_i, this amounts to proving that for all $j = 1..m_i$ and all $d \in [\![\mathtt{S}_{i,j}]\!]D$

$$F_i(\mathrm{inj}_j([\![\mathtt{S}_{i,j}]\!]\, \hat{F}\, d)) = \hat{F}_i(\mathtt{op}_{i,j}(I_{\mathtt{S}_{i,j}}\, d)). \tag{5.37}$$

One can see this by proving for all sorts $\mathtt{S}$, all $d \in [\![\mathtt{S}]\!]D$ and all $t \in \Sigma(\mathtt{S})$ that

$$I_{\mathtt{S}}\, d = [t]_{=_\alpha} \Rightarrow (t, [\![\mathtt{S}]\!]\, \hat{F}\, d) \in \overline{F}_{\mathtt{S}}. \tag{5.38}$$

For then if $I_{\mathtt{S}_{i,j}}\, d = [t]_{=_\alpha}$, we have $(t, [\![\mathtt{S}_{i,j}]\!]\, \hat{F}\, d) \in \overline{F}_{\mathtt{S}_{i,j}}$ and therefore by definition of $\overline{F}$, also $(\mathtt{op}_{i,j}\, t, F_i(\mathrm{inj}_j([\![\mathtt{S}_{i,j}]\!]\, \hat{F}\, d))) \in \overline{F}_{\mathtt{D}_i}$; but then (5.36) and (5.33) together imply (5.37).

That leaves the proof of property (5.38). This can be done by induction on the structure of the sort $\mathtt{S}$. The induction step when $\mathtt{S} = \mathtt{D}_i$ uses (5.36); and the induction step for name abstraction sorts uses the fact that every $d \in [\![\mathtt{N}\ .\ \mathtt{S}]\!]D = [\mathbb{A}]([\![\mathtt{S}]\!]D)$ is of the form $\langle a\rangle d'$ for some $a \mathbin{\#} F$, together with the fact that

$$a \mathbin{\#} F \wedge (t, d) \in \overline{F}_{\mathtt{S}} \Rightarrow (a\ .\ t, \langle a\rangle d) \in \overline{F}_{\mathtt{N.S}} \tag{5.39}$$

which is a consequence of the definition of $\overline{F}$ and the fact that it is supported by $\operatorname{supp} F$. □

Proof of uniqueness of $\hat{F}$ Suppose $F' \in D \to^n_{\mathrm{fs}} X$ also satisfies $F \circ (\mathrm{T}F') = F' \circ I$ and hence for all $i = 1..n$, all $j = 1..m_i$ and all $d \in [\![\mathtt{S}_{i,j}]\!]D$

$$F_i(\mathrm{inj}_j([\![\mathtt{S}_{i,j}]\!]\, F'\, d)) = F'_i(\mathtt{op}_{i,j}(I_{\mathtt{S}_{i,j}}\, d)). \tag{5.40}$$

From this it follows by induction of the structure of sorts $\mathtt{S}$ that for all $d \in [\![\mathtt{S}]\!]D$ and all $t \in \Sigma(\mathtt{S})$

$$I_{\mathtt{S}}\, d = [t]_{=_\alpha} \Rightarrow (t, [\![\mathtt{S}]\!]\, F'\, d) \in \overline{F}_{\mathtt{S}}. \tag{5.41}$$

Taking $\mathtt{S} = \mathtt{D}_i$, this gives $(t, F'_i[t]_{=_\alpha}) \in \overline{F}_{\mathtt{D}_i}$ for each $t \in \Sigma(\mathtt{D}_i)$. Combining this with (5.36) and (5.33), we get $F'_i[t]_{=_\alpha} = \hat{F}_i[t]_{=_\alpha}$ for all $t \in \Sigma(\mathtt{D}_i)$ and all $i = 1..n$. Therefore $F' = \hat{F}$. □

5.5 Primitive recursion

The initial algebra property of ordinary algebraic data types is equivalent to a familiar and widely used principle of structural recursion for such data. When we add names and name abstraction to get nominal algebraic data types, it turns out that the initial algebra property (Theorem 5.12) gives rise to a principle of structural recursion 'modulo α-equivalence'. This formalizes many common informal uses of structural recursion in the presence of binding operations, where one identifies α-equivalence classes with representative raw terms, dynamically freshening bound names as necessary. Pitts (2006) investigates this 'α-structural' recursion in some generality. Here we will just treat one simple example, untyped λ-terms.

Let Σ be the signature from Example 5.2. The associated nominal algebraic functor **Nom** → **Nom** is $\mathrm{T} = \mathbb{A} + [\mathbb{A}]\_ + (\_ \times \_)$. Thus to give a T-algebra $F \in \mathrm{T}X \to_{\mathrm{fs}} X$

is equivalent to giving a nominal set X equipped with three finitely supported functions, $F_1 \in \mathbb{A} \to_{\mathrm{fs}} X$, $F_2 \in ([\mathbb{A}]X) \to_{\mathrm{fs}} X$ and $F_3 \in X \times X \to_{\mathrm{fs}} X$. By Theorem 4.14, F_2 is induced by a finitely supported partial function $(\mathbb{A} \times X) \rightharpoonup_{\mathrm{fs}} X$ satisfying (4.26). It seems that the added generality of using a *partial* function is not needed in practice; so we will use a total function $F'_2 \in (\mathbb{A} \times X) \to_{\mathrm{fs}} X$, for which condition (4.26) becomes $(\text{И}a)(\forall x \in X)\ a \mathrel{\#} F'_2(a, x)$. In particular, the intial T-algebra $\Sigma_\alpha(\mathtt{Term})$ is the nominal set of λ-terms modulo α-equivalence equipped with the equivariant functions

$$
\begin{aligned}
&I_1 : \mathbb{A} \longrightarrow \Sigma_\alpha(\mathtt{Term})\\
&I_1 \triangleq \lambda a \in \mathbb{A} \to \mathtt{V}\,a\\[1ex]
&I'_2 : \mathbb{A} \times \Sigma_\alpha(\mathtt{Term}) \longrightarrow \Sigma_\alpha(\mathtt{Term})\\
&I'_2 \triangleq \lambda(a, e) \in \mathbb{A} \times \Sigma_\alpha(\mathtt{Term}) \to \mathtt{L}\,a\ \mathtt{.}\ e\\[1ex]
&I_3 : \Sigma_\alpha(\mathtt{Term}) \times \Sigma_\alpha(\mathtt{Term}) \longrightarrow \Sigma_\alpha(\mathtt{Term})\\
&I_3 \triangleq \lambda(e_1, e_2) \in \Sigma_\alpha(\mathtt{Term}) \times \Sigma_\alpha(\mathtt{Term}) \to \mathtt{A}(e_1\ \mathtt{,}\ e_2).
\end{aligned}
$$

The initial algebra theorem for nominal algebraic data types in this case gives the following recursion principle for λ-terms.

Theorem 5.13 (**α-Structural primitive recursion for λ-terms**) *Let Σ be the signature from Example 5.2. Given a nominal set X and finitely supported functions*

$$
\begin{aligned}
&F_1 \in \mathbb{A} \to_{\mathrm{fs}} X\\
&F_2 \in \mathbb{A} \times \Sigma_\alpha(\mathtt{Term}) \times X \to_{\mathrm{fs}} X\\
&F_3 \in \Sigma_\alpha(\mathtt{Term}) \times \Sigma_\alpha(\mathtt{Term}) \times X \times X \to_{\mathrm{fs}} X
\end{aligned}
$$

with F_2 satisfying the following 'freshness condition for binders'

$$
(\text{И}a \in \mathbb{A})(\forall e \in \Sigma_\alpha(\mathtt{Term}))(\forall x \in X)\ a \mathrel{\#} F_2(a, e, x), \tag{5.42}
$$

then there is a unique finitely supported function $\hat{F} \in \Sigma_\alpha(\mathtt{Term}) \to_{\mathrm{fs}} X$ satisfying for all $a \in \mathbb{A}$ and $e, e_1, e_2 \in \Sigma_\alpha(\mathtt{Term})$

$$
\begin{aligned}
\hat{F}(\mathtt{V}\,a) &= F_1\,a\\
a \mathrel{\#} (F_1, F_2, F_2) \Rightarrow \hat{F}(\mathtt{L}\,a\ \mathtt{.}\ e) &= F_2(a, e, \hat{F}\,e)\\
\hat{F}(\mathtt{A}(e_1\ \mathtt{,}\ e_2)) &= F_3(e_1, e_2, \hat{F}\,e_1, \hat{F}\,e_2).
\end{aligned}
\tag{5.43}
$$

Proof Let $D = \Sigma_\alpha(\mathtt{Term})$. Define $G_1 \in \mathbb{A} \to_{\mathrm{fs}} (X \times D)$, $G_2 \in [\mathbb{A}](X \times D) \to_{\mathrm{fs}} (X \times D)$

and $G_3 \in (X \times D) \times (X \times D) \to_{\mathrm{fs}} (X \times D)$ as follows.

$$\begin{aligned}
&G_1 \triangleq \lambda a \in \mathbb{A} \to (F_1\,a, \mathtt{V}\,a)\\
&G_2 \triangleq \lambda\langle a\rangle(x,e) \in [\mathbb{A}](X \times D) \to (F_2(a,e,x), \mathtt{L}\,a\;\mathtt{.}\;e)\\
&G_3 \triangleq \lambda((x_1,e_1),(x_2,e_2)) \in (X \times D) \times (X \times D) \to (F_3(e_1,e_2,x_1,x_2), \mathtt{A}(e_1\;\mathtt{,}\;e_2)).
\end{aligned}$$

The definition of G_2 uses the name abstraction pattern notation introduced after Theorem 4.14. Thus G_2 is the unique function corresponding as in that theorem to $(a,(x,e)) \mapsto (F_2(a,e,x), \mathtt{L}\,a\;\mathtt{.}\;e)$; it is well-defined, because $(\reflectbox{\textsf{N}}a)(\forall(x,e) \in X \times D)\; a \mathbin{\#} (F_2(a,e,x), \mathtt{L}\,a\;\mathtt{.}\;e)$ holds by (5.42) and Proposition 5.7.

From (G_1, G_2, G_3) we get a T-algebra $G \in \mathrm{T}(X \times D) \to_{\mathrm{fs}} (X \times D)$ for the functor $\mathrm{T} = \mathbb{A} + [\mathbb{A}]\_ + (\_ \times \_)$. Applying Theorem 5.12, there is a unique $\hat{G} \in D \to_{\mathrm{fs}} (X \times D)$ satisfying $G \circ (\mathrm{T}\hat{G}) = \hat{G} \circ I$. Define

$$\begin{aligned}
F &\triangleq \mathrm{proj}_1 \circ \hat{G} \in D \to_{\mathrm{fs}} X\\
J &\triangleq \mathrm{proj}_2 \circ \hat{G} \in D \to_{\mathrm{fs}} D.
\end{aligned}$$

Then $G \circ (\mathrm{T}\hat{G}) = \hat{G} \circ I$ gives

$$\begin{aligned}
&(\forall a \in \mathbb{A})\; (F(\mathtt{V}\,a), J(\mathtt{V}\,a)) = (F_1\,a, \mathtt{V}\,a)\\
&\qquad \wedge (\reflectbox{\textsf{N}}a)(\forall e \in D)\; (F(\mathtt{L}\,a\;\mathtt{.}\;e), J(\mathtt{L}\,a\;\mathtt{.}\;e)) = (F_2(a, J\,e, F\,e), \mathtt{L}\,a\;\mathtt{.}\;(J\,e))\\
&\qquad\qquad \wedge (\forall e_1, e_2 \in D)\; (F(\mathtt{A}(e_1\;\mathtt{,}\;e_2)), J(\mathtt{A}(e_1\;\mathtt{,}\;e_2))) =\\
&\qquad\qquad\qquad (F_3(J\,e_1, J\,e_2, F\,e_1, F\,e_2), \mathtt{A}((J\,e_1)\;\mathtt{,}\;(J\,e_2))). \qquad (5.44)
\end{aligned}$$

The second components of the equalities in (5.44) give

$$\begin{aligned}
&(\forall a \in \mathbb{A})\; J(\mathtt{V}\,a) = \mathtt{V}\,a\\
&\qquad \wedge (\reflectbox{\textsf{N}}a)(\forall e \in D)\; J(\mathtt{L}\,a\;\mathtt{.}\;e) = \mathtt{L}\,a\;\mathtt{.}\;(J\,e)\\
&\qquad\qquad \wedge (\forall e_1, e_2 \in D)\; J(\mathtt{A}(e_1\;\mathtt{,}\;e_2)) = \mathtt{A}((J\,e_1)\;\mathtt{,}\;(J\,e_2)).
\end{aligned}$$

So J satisfies $I \circ (\mathrm{T}\,J) = J \circ I$; but so does id_D and by Theorem 5.12 there is only one such function corresponding to the T-algebra $I : \mathrm{T}\,D \to D$. Therefore $J = \mathrm{id}_D$ and hence the first components of the equalities in (5.44) imply that we can take $\hat{F} = F$ to satisfy (5.43).

For the uniqueness of $\hat{F}$, note that if F' is any other such function, then $G' \triangleq \langle F', \mathrm{id}_D\rangle \in D \to_{\mathrm{fs}} (X \times D)$ satisfies $G \circ (\mathrm{T}\,G') = G' \circ I$. So by the uniquness part of Theorem 5.12, $G' = \hat{G}$ and hence $F' = \mathrm{proj}_1 \circ G' = \mathrm{proj}_1 \circ \hat{G} = F = \hat{F}$. □

Remarks 1. The proof of the theorem uses a standard technique for deriving primitive recursion from the more simple, ‘iterative’ form of recursion inherent in the existence part of the initial algebra property, using the uniqueness part of initiality.

2. The ‘freshness condition on binders’ (5.42) ensures that the functions F_1, F_2 and F_3 induce a T-algebra structure on X. Similar conditions can be given for any nominal algebraic signature: see (Pitts, 2006, Theorem 5.1).

Example 5.14 (Counting λ-abstractions) Here is a simple example to illustrate the use of Theorem 5.13. Taking X to be the discrete nominal set of natural number $\mathbb{N}$ (section 2.3), consider the functions F_1, F_2, F_3 given by

$$\begin{aligned} F_1\,a &\triangleq 0 \\ F_2(a, e, x) &\triangleq x + 1 \\ F_3(e_1, e_2, x_1, x_2) &\triangleq x_1 + x_2. \end{aligned} \tag{5.45}$$

Note that these functions have empty support; and since every element of $\mathbb{N}$ has empty support, the freshness condition for binders (5.42) is trivially satisfied. So the theorem gives us a function $|\_| \triangleq \hat{F} : \Sigma_\alpha(\mathtt{Term}) \to \mathbb{N}$ satisfying

$$\begin{aligned} |\mathtt{V}\,a| &= 0 \\ |\mathtt{L}\,a\ \mathtt{.}\ e| &= |e| + 1 \\ |\mathtt{A}(e_1\ \mathtt{,}\ e_2)| &= |e_1| + |e_2|. \end{aligned} \tag{5.46}$$

Note also that the theorem tells us that $|\_|$ is supported by $\mathrm{supp}(F_1, F_2, F_3) = \emptyset$ and hence is equivariant.

For example $|\mathtt{L}\,a\ \mathtt{.}\ \mathtt{V}\,a| = |\mathtt{V}\,a| + 1 = 0 + 1 = 1$ and in general $|e|$ is the number of occurrences of λ-abstractions in the λ-term e. Although the well-definedness of such a function is a simple application of the theorem, it is interesting to note that previous formal recursion schemes for α-equivalence classes of raw λ-terms find it troublesome: see (Gordon and Melham, 1996, Sect. 3.3) and (Norrish, 2004, Sect. 3). It is also interesting to compare the ease with which $|\_|$ can be defined compared with schemes for primitive recursion based on using higher-order abstract syntax: see (Schürmann et al., 2001, Example 4.4).

Example 5.15 (Capture-avoiding substitution) Given $a \in \mathbb{A}$ and $e \in \Sigma_\alpha(\mathtt{Term})$, if in Theorem 5.13 we take $X = \Sigma_\alpha(\mathtt{Term})$ and

$$\begin{aligned} F_1\,a_1 &\triangleq \text{if } a_1 = a \text{ then } e \text{ else } \mathtt{V}\,a_1 \\ F_2(a_1, e_1, x_1) &\triangleq \mathtt{L}\,a_1\ \mathtt{.}\ x_1 \\ F_3(e_1, e_2, x_1, x_2) &\triangleq \mathtt{A}(x_1\ \mathtt{,}\ x_2), \end{aligned}$$

then F_2 clearly satisfies the freshness condition for binders (5.42) and the function $\hat{F}$ given by the theorem is the operation $(a := e) \in \Sigma_\alpha(\mathtt{Term}) \to_{\mathrm{fs}} \Sigma_\alpha(\mathtt{Term})$ of capture-avoiding substitution from Example 5.11.

In the previous two examples it was easy to verify that the freshness condition for binders (5.42) held. The following example illustrates that this is not always the case.

Example 5.16 (**Counting bound variable occurrences**) We wish to define a function $cbv : \Sigma_\alpha(\mathtt{Term}) \to \mathbb{N}$ that counts the number of occurrences of bound variables in a λ-term. For example, assuming a and b are distinct, the λ-term $(\lambda a.\lambda b.a)\,b$ contains a single bound variable occurrence (named a in the raw term we have used to represent the λ-term) and correspondingly we want the value of cbv at $\mathtt{A}(\mathtt{L}\,a\;\mathtt{.}\;\mathtt{L}\,b\;\mathtt{.}\;\mathtt{V}\,a\;\mathtt{,}\;\mathtt{V}\,b) \in \Sigma_\alpha(\mathtt{Term})$ to be 1.

We define cbv using the approach of (Schürmann et al., 2001, Example 4.3). We first define an auxiliary function $cbvs$ satisfying

$$\begin{aligned} cbvs(\mathtt{V}\,a)\,\rho &= \rho\,a \\ cbvs(\mathtt{L}\,a\;\mathtt{.}\;e)\,\rho &= (cbvs\,e)(\rho[a \mapsto 1]) \\ cbvs(\mathtt{A}(e_1\;\mathtt{,}\;e_2)) &= (cbvs\,e_1\,\rho) + (cbvs\,e_2\,\rho) \end{aligned} \tag{5.47}$$

where ρ ranges over environments mapping atomic names to numbers; in the second clause above, $\rho[a \mapsto 1]$ indicates the updated environment mapping a to 1 and otherwise acting like ρ. Then we define $cbv\,e \triangleq cbvs\,e\,\rho_0$ where ρ_0 is the environment mapping all atomic names to 0.

For environments we do not use aritrary functions from names to numbers, but rather finitely supported ones. Then as a first attempt to use α-structural primitive recursion to prove the existence of a function $cbvs$ satisfying (5.47), in Theorem 5.13 one could try taking X to be $(\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$ and using the functions F_1, F_2, F_3 given by

$$\begin{aligned} F_1\,a &\triangleq \lambda\rho \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to \rho\,a \\ F_2(a,e,x) &\triangleq \lambda\rho \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to x(\rho[a \mapsto 1]) \\ F_3(e_1,e_2,x_1,x_2) &\triangleq \lambda\rho \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to x_1\,\rho + x_2\,\rho. \end{aligned} \tag{5.48}$$

The problem is that F_2 does not satisfying the freshness condition for binders; in other words there is an $a \in \mathbb{A}$ and $x \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$ for which $a \mathrel{\#} \lambda\rho \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to x(\rho[a \mapsto 1])$ does not hold (see Exercise 5.3(ii)).

To solve this problem we identify a property of environment functionals that is preserved by the operations needed in (5.47). (This is analogous to 'strengthening the induction hypothesis' in a proof by induction, given the close relationship that exists between recursion and induction.) Specifically, we cut down to those $x \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$ whose value at an environment ρ only depends on the values of ρ at names in the support of x. More precisely, consider the nominal subset of $(\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$ given by

$$X \triangleq \{x \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N} \mid (\text{И}a)(\forall\rho \in \mathbb{A} \to_{\mathrm{fs}} \mathbb{N})(\forall n \in \mathbb{N})\; x(\rho[a \mapsto n]) = x\,\rho\}.$$

It is not hard to see that the functions F_1, F_2, F_3 defined in (5.48) are equivariant and satisfy

$$\begin{aligned} F_1\, a &\in X \\ x \in X &\Rightarrow F_2(a, e, x) \in X \\ x_1, x_2 \in X &\Rightarrow F_3(e_1, e_2, x_1, x_2) \in X. \end{aligned}$$

So they give morphisms in **Nom**

$$\begin{aligned} F_1 &: \mathbb{A} \to X \\ F_2 &: \mathbb{A} \times \Sigma_\alpha(\texttt{Term}) \times X \to X \\ F_3 &: \Sigma_\alpha(\texttt{Term}) \times \Sigma_\alpha(\texttt{Term}) \times X \times X \to X. \end{aligned}$$

Furthermore, F_2 satisfies the freshness condition for binders (5.42), because for any $a \in \mathbb{A}$ and $x \in X$, using the Choose-a-Fresh-Name Principle to pick $a_1 \mathrel{\#} (a, x, \lambda\rho \to x(\rho[a \mapsto 1]))$, we have

$$\begin{array}{ll} & a \\ = & \\ & (a\ a_1) \cdot a_1 \\ \# & \{\text{since } a_1 \mathrel{\#} \lambda\rho \to x(\rho[a \mapsto 1])\} \\ & (a\ a_1) \cdot (\lambda\rho \to x(\rho[a \mapsto 1])) \\ = & \{\text{by definition of the action of permutations on functions}\} \\ & \lambda\rho \to (a\ a_1) \cdot (x(((a\ a_1) \cdot \rho)[a \mapsto 1])) \\ = & \{\text{by Exercise 5.3(i) and since } a_1 \neq a\} \\ & \lambda\rho \to (a\ a_1) \cdot (x(\rho[a \mapsto 1][a_1 \mapsto \rho\, a])) \\ = & \{\text{since } x \in X \text{ and } a_1 \mathrel{\#} x\} \\ & \lambda\rho \to (a\ a_1) \cdot (x(\rho[a \mapsto 1])) \\ = & \{\text{since } x(\rho[a \mapsto 1]) \in \mathbb{N} \text{ and hence has empty support}\} \\ & \lambda\rho \to x(\rho[a \mapsto 1]). \end{array}$$

Therefore we can apply Theorem 5.13 to get $\hat{F} \in \Sigma_\alpha(\texttt{Term}) \to_{\mathrm{fs}} X$ satisfying (5.43); and since $\mathrm{supp}(F_1, F_2, F_3) = \emptyset$, this implies that we do have the required recursion properties (5.47) once we take $cbvs = \hat{F}$.

5.6 Induction

Initial algebras $I : \mathrm{T}\, D \to D$ for functors $\mathrm{T} : \mathbf{C} \to \mathbf{C}$ automatically satisfy a category-theoretic induction principle: if a subobject of D, given by a monomorphism $M : P \rightarrowtail D$ say, is such that $I \circ \mathrm{T}\, M$ factors through M

$$\begin{tikzcd}
\mathrm{T}\, P \arrow[r, "\mathrm{T}\, M"] \arrow[d, dashed] & \mathrm{T}\, D \arrow[d, "I"] \\
P \arrow[r, tail, "M"'] & D
\end{tikzcd} \tag{5.49}$$

then the subobject is necessarily the whole of D, that is, M is an isomorphism. We leave the proof of this as an exercise (Exercise 5.4).

When **C** is **Nom**, we know from section 2.5 that subobjects correspond to equivariant subsets. Just as for recursion, in order to capture some common informal uses of induction in the presence of α-equivalence, we need to establish a stronger version of this induction principle, one that applies to finitely supported subsets rather than just equivariant ones. To do so we exploit the fact that not only do nominal algebraic functors preserve monomorphisms and hence act on subobjects, but also this action internalizes to **Nom**'s power objects, the nominal sets of finitely supported subsets. (See Exercise 5.5.)

Given $X = (X_1, \ldots, X_n) \in \mathbf{Nom}^n$, define

$$\mathrm{P}^n_{\mathrm{fs}}\, X \triangleq \mathrm{P}_{\mathrm{fs}}\, X_1 \times \cdots \times \mathrm{P}_{\mathrm{fs}}\, X_n. \tag{5.50}$$

If Σ is a nominal algebraic signature with a single name-sort $\mathtt{N}$ and n data-sorts $\mathtt{D}_1, \ldots, \mathtt{D}_n$, then we get a sort-indexed family of equivariant functions

$$\Diamond_{\mathtt{S}} : \mathrm{P}^n_{\mathrm{fs}}\, X \to \mathrm{P}_{\mathrm{fs}}\, ([\![\mathtt{S}]\!]X) \tag{5.51}$$

defined by recursion on the structure of the sort $\mathtt{S}$ as follows.

$$\begin{aligned}
\Diamond_{\mathtt{N}} P &= \mathbb{A} \\
\Diamond_{\mathtt{D}_i} P &= P_i \\
\Diamond_{\mathtt{1}} P &= 1 \\
\Diamond_{\mathtt{S}_1 , \mathtt{S}_2} P &= \{(d_1, d_2) \mid d_1 \in \Diamond_{\mathtt{S}_1} P \wedge d_2 \in \Diamond_{\mathtt{S}_2} P\} \\
\Diamond_{\mathtt{N}.\mathtt{S}} P &= \{\langle a \rangle d \mid a \mathrel{\#} P \wedge d \in \Diamond_{\mathtt{S}} P\}.
\end{aligned} \tag{5.52}$$

Theorem 5.17 (**α-Structural induction for nominal algebraic data types**) *Let Σ be a nominal algebraic signature with a single name-sort $\mathtt{N}$, with n data-sorts $\mathtt{D}_1, \ldots, \mathtt{D}_n$, and with operations as shown in* (5.15). *The initial algebra*

$$D = (\Sigma_\alpha(\mathtt{D}_1), \ldots, \Sigma_\alpha(\mathtt{D}_n))$$

for the associated nominal algebraic functor $\mathrm{T} : \mathbf{Nom}^n \to \mathbf{Nom}^n$ *has the following induction property: for any $P = (P_1, \ldots, P_n) \in \mathrm{P}^n_{\mathrm{fs}}\, D$, to show that $P_i = \Sigma_\alpha(\mathtt{D}_i)$ for* *each* $i = 1..n$ *it suffices to show for each of the signature's operations* $\mathtt{op}_{i,j} : \mathtt{S}_{i,j} \to \mathtt{D}_i$ *that*

$$(\forall d \in [\![\mathtt{S}_{i,j}]\!]D)\ d \in \Diamond_{\mathtt{S}_{i,j}} P \Rightarrow \mathtt{op}_{i,j}(I_{\mathtt{S}i,j}\, d) \in P_i \tag{5.53}$$

where $I_{\mathtt{S}_{i,j}} : [\![\mathtt{S}_{i,j}]\!]D \to \Sigma_\alpha(\mathtt{S}_{i,j})$ *is as in* (5.23).

Proof Given $P \in \mathrm{P}^n_{\mathrm{fs}}\, D$ satisfying (5.53) for all $i = 1..n$ and $j = 1..m_i$, we will show that the sort-indexed family of subsets

$$H_{\mathtt{S}} \triangleq \{t \in \Sigma(\mathtt{S}) \mid (\forall \pi \in \mathrm{Perm}\,\mathbb{A})(\exists d \in \Diamond_{\mathtt{S}} P)\ I_{\mathtt{S}}\, d = [\pi \cdot t]_{=_\alpha}\}$$

is closed under the rules in Definition 5.1 inductively defining the sets $\Sigma(\mathtt{S})$ of raw terms of each sort, and hence that $H_{\mathtt{S}} = \Sigma(\mathtt{S})$. (The quantification over all finitary permutations $\pi \in \mathrm{Perm}\,\mathbb{A}$ in the definition of $H_{\mathtt{S}}$ is there to ensure that these subsets are equivariant, despite the fact that P may have non-empty support; indeed, it is easy to see from the definition that we have $t \in H_{\mathtt{S}} \Rightarrow \pi \cdot t \in H_{\mathtt{S}}$.). We get $P_i = \Sigma_\alpha(\mathtt{D}_i)$ from $H_{\mathtt{S}} = \Sigma(\mathtt{S})$ in case $\mathtt{S} = \mathtt{D}_i$, since $\Sigma_\alpha(\mathtt{D}_i) = \Sigma(\mathtt{D}_i)/{=_\alpha}$, $\Diamond_{\mathtt{D}_i} P = P_i$ and $I_{\mathtt{D}_i} = \mathrm{id}_{\Sigma_\alpha(\mathtt{D}_i)}$.

Closure of $H_{\mathtt{S}}$ under the rules in Definition 5.1 for atomic name, unit and pair raw terms is straightforward. Closure under the rule for raw terms of the form $\mathtt{op}\, t$ follows directly from the assumption (5.53). So it just remains to show closure under the rule for raw terms of the form $a\ .\ t$. So suppose $t \in H_{\mathtt{S}}$, $a \in \mathbb{A}$ and $\pi \in \mathrm{Perm}\,\mathbb{A}$. We have to find $d \in \Diamond_{\mathtt{N.S}} P$ with $I_{\mathtt{N.S}}\, d = [\pi \cdot (a\ .\ t)]_{=_\alpha}$. Use the Choose-a-Fresh-Name Principle to pick $a' \mathrel{\#} (a, \pi, t, P)$. Since $t \in H_{\mathtt{S}}$, there exists $d' \in \Diamond_{\mathtt{S}} P$ with $I_{\mathtt{S}}\, d' = [(\pi\, a\ a') \cdot \pi \cdot t]_{=_\alpha}$. Since $a' \mathrel{\#} P$, by definition of $\Diamond_{\mathtt{N.S}} P$ it contains $\langle a' \rangle d'$; and by definition of $I_{\mathtt{N.S}}$, we have $I_{\mathtt{N.S}}(\langle a' \rangle d') = a'\ .\ (I_{\mathtt{S}}\, d') = [a'\ .\ ((\pi\, a\ a') \cdot \pi \cdot t)]_{=_\alpha} = [(\pi\, a)\ .\ (\pi \cdot t)]_{=_\alpha}$, since $a' \mathrel{\#} (\pi\, a, \pi \cdot t)$. So we can take $d = \langle a' \rangle d'$. □

We saw by example in section 5.5 that the structure needed for a T-algebra is equivalent to giving some functions not directly involving the name abstraction construct, together with a 'freshness condition for binders'. Similarly, the induction hypothesis (5.53) in the α-structural induction theorem is equivalent to a more elementary, albeit more involved condition involving the freshness relation. See (Pitts, 2006, Theorem 5.2) for the general case. Here we just illustrate this for the signature from Example 5.2. In this case $D = \Sigma_\alpha(\mathtt{Term})$ is the nominal set of λ-terms modulo α-equivalence. The induction hypothesis (5.53) for the signature's three operations, $\mathtt{V}$, $\mathtt{L}$ and $\mathtt{A}$ is equivalent to asserting of a finitely supported subset

$P \in \mathrm{P}_{\mathrm{fs}}(\Sigma_\alpha(\mathtt{Term}))$ that it satisfies

$$\begin{aligned} &(\forall a \in \mathbb{A})\ \mathtt{V}\,a \in P \\ &\wedge (\text{И}a)(\forall e \in \Sigma_\alpha(\mathtt{Term}))\ e \in P \Rightarrow \mathtt{L}\,a\ \mathtt{.}\ e \in P \\ &\wedge (\forall e_1, e_2 \in \Sigma_\alpha(\mathtt{Term}))\ e_1 \in P \wedge e_2 \in P \Rightarrow \mathtt{A}(e_1\ \mathtt{,}\ e_2) \in P. \end{aligned} \tag{5.54}$$

(In the case of L we use the equivalence of

$$(\forall a \in \mathbb{A})(\forall e \in \Sigma_\alpha(\mathtt{Term}))\ a \# P \wedge e \in P \Rightarrow \mathtt{L}\,a\ \mathtt{.}\ e \in P$$

with

$$(\text{И}a)(\forall e \in \Sigma_\alpha(\mathtt{Term}))\ e \in P \Rightarrow \mathtt{L}\,a\ \mathtt{.}\ e \in P$$

which follows from the 'some/any' theorem.) So we get the following corollary of Theorem 5.17.

Corollary 5.18 (**α-Structural induction principle for λ-terms**) *With Σ as in Example 5.2, for any finitely supported subset $P \in \mathrm{P}_{\mathrm{fs}}(\Sigma_\alpha(\mathtt{Term}))$, if* (5.54) *holds, then $(\forall e \in \Sigma_\alpha(\mathtt{Term}))\ e \in P$.*

Example 5.19 Consider the definition (5.29) in Example 5.11 of capture-avoiding substitution $(a := e)e'$ of e for free occurrences of the variable $\mathtt{V}\,a$ in e'. We saw above that $(a := e) : \Sigma_\alpha(\mathtt{Term}) \to \Sigma_\alpha(\mathtt{Term})$ is the function $\hat{F}$ in Theorem 5.13 when F_1, F_2, F_3 are defined as in Example 5.15. We illustrate the α-structural induction principle for λ-terms by using Corollary 5.18 to prove

$$a \# e_1 \Rightarrow (a := e)e_1 = e_1.$$

Given $a \in \mathbb{A}$ and $e \in \Sigma_\alpha(\mathtt{Term})$, define

$$P \triangleq \{e_1 \in \Sigma_\alpha(\mathtt{Term}) \mid a \# e_1 \Rightarrow (a := e)e_1 = e_1\}.$$

Thus $P \in \mathrm{P}_{\mathrm{fs}}(\Sigma_\alpha(\mathtt{Term}))$ is supported by $\{a\} \cup \operatorname{supp} e$. To see that P is the whole of $\Sigma_\alpha(\mathtt{Term})$ we need to check that it satisifes (5.54).

Proof of $(\forall a_1 \in \mathbb{A})\ \mathtt{V}\,a_1 \in P$. For any $a_1 \in \mathbb{A}$, if $a \# \mathtt{V}\,a_1$ then $a \neq a_1$ and hence $(a := e)(\mathtt{V}\,a_1) = F_1\,a_1 = \mathtt{V}\,a_1$; so $\mathtt{V}\,a_1 \in P$. □

Proof of $(\text{И}a_1)(\forall e_1 \in \Sigma_\alpha(\mathtt{Term}))\ e_1 \in P \Rightarrow \mathtt{L}\,a_1\ \mathtt{.}\ e_1 \in P$. Since P is supported by $\{a\} \cup \operatorname{supp} e$, by the 'some/any' theorem it suffices to prove

$$(\exists a_1 \in \mathbb{A})\ a_1 \# (a, e) \wedge (\forall e_1 \in \Sigma_\alpha(\mathtt{Term}))\ e_1 \in P \Rightarrow \mathtt{L}\,a_1\ \mathtt{.}\ e_1 \in P.$$

Use the Choose-a-Fresh-Name Principle to pick $a_1 \in \mathbb{A}$ with $a_1 \# (a, e)$. For any

$e_1 \in P$, if $a \mathrel{\#} \mathtt{L}\, a_1 \mathrel{\mathtt{.}} e_1$, then since $a \neq a_1$ we must have $a \mathrel{\#} e_1$ and therefore

$$\begin{aligned}
& (a := e)(\mathtt{L}\, a_1 \mathrel{\mathtt{.}} e_1) \\
= \ & \{\text{since } (a := e) \text{ is } \hat{F} \text{ and } a_1 \mathrel{\#} (a, e)\} \\
& F_2(a_1, e_1, (a := e)e_1) \\
= \ & \{\text{by definition of } F_2\} \\
& \mathtt{L}\, a_1 \mathrel{\mathtt{.}} (a := e)e_1 \\
= \ & \{\text{since } e_1 \in P \text{ and } a \mathrel{\#} e_1\} \\
& \mathtt{L}\, a_1 \mathrel{\mathtt{.}} e_1.
\end{aligned}$$

So $\mathtt{L}\, a_1 \mathrel{\mathtt{.}} e_1 \in P$. □

Proof of $(\forall e_1, e_2 \in \Sigma_\alpha(\mathtt{Term}))\ e_1 \in P \wedge e_2 \in P \Rightarrow \mathtt{A}(e_1 \mathrel{\mathtt{,}} e_2) \in P$. Suppose $e_1 \in P$ and $e_2 \in P$. If $a \mathrel{\#} \mathtt{A}(e_1 \mathrel{\mathtt{,}} e_2)$, then $a \mathrel{\#} e_1 \wedge a \mathrel{\#} e_2$ and hence $(a := e)e_i = e_i$ for $i = 1, 2$. Therefore $(a := e)(\mathtt{A}(e_1 \mathrel{\mathtt{,}} e_2)) = F_3(e_1, e_2, (a := e)e_1, (a := e)e_2) = \mathtt{A}((a := e)e_1 \mathrel{\mathtt{,}} (a := e)e_2) = \mathtt{A}(e_1 \mathrel{\mathtt{,}} e_2)$. Thus $\mathtt{A}(e_1 \mathrel{\mathtt{,}} e_2) \in P$. □

Exercises

5.1 Show that nominal algebraic functors preserve countable colimits of chains in **Nom**.

5.2 Prove *Lambek's Lemma*: if $I : \mathrm{T}D \to D$ is an initial algebra for a functor $\mathrm{T} : \mathbf{C} \to \mathbf{C}$, then I is an isomorphism.

5.3 (i) If $x \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$, $\rho \in \mathbb{A} \to_{\mathrm{fs}} \mathbb{N}$ and $\pi \in \operatorname{Perm} \mathbb{A}$, show that $\pi \cdot \rho = \rho \circ \pi$ and $(\pi \cdot x)\rho = x(\rho \circ \pi)$.

(ii) If $\rho \in \mathbb{A} \to_{\mathrm{fs}} \mathbb{N}$ and $a \in \mathbb{A}$, define

$$\rho[a \mapsto 1] \triangleq \lambda b \in \mathbb{A} \to \text{if } b = a \text{ then } 1 \text{ else } \rho\, b.$$

Note that by the Finite Support Principle, $\rho[a \mapsto 1] \in \mathbb{A} \to_{\mathrm{fs}} \mathbb{N}$; and similarly $\lambda\rho \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to x(\rho[a \mapsto 1])$ is in $(\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$. Given $a \in \mathbb{A}$, show that there exists $x \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$ such that a is in the support of $\lambda\rho \in (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to x(\rho[a \mapsto 1])$. [Hint: consider the equivariant function mapping each $\rho \in \mathbb{A} \to_{\mathrm{fs}} \mathbb{N}$ to the cardinality of $\{b \in \mathbb{A} \mid \rho\, b = 1\}$ is that set is finite and to 0 otherwise.]

5.4 Prove the category-theoretic induction principle mentioned at the beginning of section 5.6.

5.5 If $\mathrm{T} : \mathbf{Nom}^n \to \mathbf{Nom}^n$ is the nominal algebraic functor associated with a signature Σ as in Definition 5.9, then from (5.51) we get equivariant functions

$$\Diamond_{\mathrm{T}} : \mathrm{P}^n_{\mathrm{fs}}\, X \to \mathrm{P}^n_{\mathrm{fs}}\, (\mathrm{T}\, X)$$

mapping each $P \in \mathrm{P}^n_{\mathrm{fs}}\, X$ to $\Diamond_{\mathrm{T}} P = (\Diamond_{\mathrm{T}_1} P, \ldots, \Diamond_{\mathrm{T}_n} P)$, where

$$\Diamond_{\mathrm{T}_i} P \triangleq \textstyle\bigcup_{j=1..m_i} \{\mathrm{inj}_j\, d \mid d \in \Diamond_{\mathrm{S}i,j} P\} \in \mathrm{P}_{\mathrm{fs}}\,(\mathrm{T}_i\, X).$$

Show that these functions agree with the functorial action of T in the following sense: if $P_i \subseteq X_i$ $(i = 1..n)$ are equivariant subsets and we write $M_i : P_i \to X_i$ for the inclusion functions, prove that $\mathrm{T}\, M : \mathrm{T}\, P \to \mathrm{T}\, X$ is again a monomorphism in $\mathbf{Nom}^n$ and that the corresponding element of $\mathrm{P}^n_{\mathrm{fs}}\, X$ is $\Diamond_{\mathrm{T}} P$.

5.6 Use the α-structural induction principle for λ-terms to show that capture-avoiding substitution satisfies

$$a_2 \mathbin{\#} (a_1, e_1) \Rightarrow (a_1 := e_1)((a_2 := e_2)e) = (a_2 := (a_1 := e_1)e_2)((a_1 := e_1)e).$$

Bibliography

Barendregt, H. P. 1984. *The Lambda Calculus: Its Syntax and Semantics*. Revised edn. North-Holland.

Gabbay, M. J. 2000. *A Theory of Inductive Definitions with α-Equivalence: Semantics, Implementation, Programming Language*. Ph.D. thesis, University of Cambridge.

Gabbay, M. J., and Pitts, A. M. 2002. A New Approach to Abstract Syntax with Variable Binding. *Formal Aspects of Computing*, **13**, 341–363.

Goguen, J. A., Thatcher, J. W., Wagner, E. G., and Wright, J. B. 1977. Initial Algebra Semantics and Continuous Algebras. *JACM*, **24**, 68–95.

Gordon, A. D., and Melham, T. 1996. Five Axioms of Alpha-Conversion. Pages 173–191 of: *Theorem Proving in Higher Order Logics, 9th International Conference*. Lecture Notes in Computer Science, vol. 1125. Springer-Verlag.

Gordon, M. J. C., and Melham, T. F. 1993. *Introduction to HOL. A theorem proving environment for higher order logic*. Cambridge University Press.

Johnstone, P. T. 2002. *Sketches of an Elephant, A Topos Theory Compendium, Volumes 1 and 2*. Oxford Logic Guides, nos. 43–44. Oxford University Press.

MacLane, S. 1971. *Categories for the Working Mathematician*. Graduate Texts in Mathematics 5. Springer-Verlag.

Menni, M. 2003. About N-Quantifiers. *Applied Categorical Structures*, **11**, 421–445.

Norrish, M. 2004. Recursive Function Definition for Types with Binders. Pages 241–256 of: *Theorem Proving in Higher Order Logics, 17th International Conference*. Lecture Notes in Computer Science, vol. 3223. Springer-Verlag.

Pitts, A. M. 2003. Nominal Logic, A First Order Theory of Names and Binding. *Information and Computation*, **186**, 165–193.

Pitts, A. M. 2006. Alpha-Structural Recursion and Induction. *Journal of the ACM*, **53**, 459–506.

Sangiorgi, D., and Walker, D. 2001. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press.

Schöpp, U. 2006. *Names and Binding in Type Theory*. Ph.D. thesis, University of Edinburgh.

Schürmann, C., Despeyroux, J., and Pfenning, F. 2001. Primitive Recursion for Higher-Order Abstract Syntax. *Theoretical Computer Science*, **266**, 1–57.

Tzevelekos, N. 2008. *Nominal Game Semantics*. Ph.D. thesis, University of Oxford. Available as Oxford University Computing Laboratory Technical Report RR-09-18.

Urban, C., Pitts, A. M., and Gabbay, M. J. 2004. Nominal Unification. *Theoretical Computer Science*, **323**, 473–497.

Notation index

General index