# Completeness for algebraic theories of local state

Sam Staton

Computer Laboratory, University of Cambridge

**Abstract.** Every algebraic theory gives rise to a monad, and monads allow a meta-language which is a basic programming language with side-effects. Equations in the algebraic theory give rise to equations between programs in the meta-language. An interesting question is this: to what extent can we put equational reasoning for programs into the algebraic theory for the monad?

In this paper I focus on local state, where programs can allocate, update and read the store. Plotkin and Power (FoSSaCS'02) have proposed an algebraic theory of local state, and they conjectured that the theory is complete, in the sense that every consistent equation is already derivable. The central contribution of this paper is to confirm this conjecture. To establish the completeness theorem, it is necessary to reformulate the informal theory of Plotkin and Power as an enriched algebraic theory in the sense of Kelly and Power (JPAA, 89:163–179). The new presentation can be read as 14 program assertions about three effects.

The completeness theorem for local state is dependent on certain conditions on the type of storable values. When the set of storable values is finite, there is a subtle additional axiom regarding quotient types.

## 1 Introduction

In this paper we are interested in reasoning about local state, about programs such as

$$\texttt{let val a = ref(3) in a:=4; !a end;} \qquad (1)$$

As Moggi suggested [16], one way to give a denotational semantics to a side-effecting program of type $\tau_1 \to \tau_2$ is to give a morphism $[\![\tau_1]\!] \to T([\![\tau_2]\!])$, in a category equipped with a monad $T$.

Many monads arise as free models of algebraic theories. Formally, we may say that an equation for a monad $T$ is a pair of morphisms $B \to T(A)$. Thus an equation can be thought of as a pair of denotations of programs, i.e. an assertion that two programs are the same. A system of equations for a monad typically gives rise to a quotient monad in which the equations are satisfied. In summary: by specifying equations between denotations of programs, we can construct a new denotational semantics which is sound for these equations.

In assessing the power of this technique, it is important to ask whether a monad is complete in the following sense: every equation $(e_1, e_2): B \to T(A)$ is either already true (i.e. $e_1 = e_2$), or it is inconsistent (i.e. only satisfied in trivial models). This property is sometimes called Hilbert-Post completeness. It means

that our monad is 'as good as possible', given the base category. There are no further equations between programs that can be accommodated in our model.

For local state, the category of sets is insufficient, and so we move to the category of nominal sets (first considered by Gabbay and Pitts [9] as a model of variable binding). This category has an object $\mathbb{A}$ of 'atoms', which we will think of as locations. This object captures many of the important properties of locations. It is infinite, although there is no injection $\mathbb{N} \rightarrowtail \mathbb{A}$ from the natural numbers; informally, the locations are local and cannot be globally enumerated.

The theory of local state is enriched in the category of nominal sets. Let $\mathbb{V}$ be some (nominal) set of values. The operations of the theory induce generic effects, which include $\mathsf{upd}\colon \mathbb{A} \times \mathbb{V} \to T(1)$ (update a location with a value), $\mathsf{lk}\colon \mathbb{A} \to T(\mathbb{V})$ (lookup the value in a location and return it), $\mathsf{ref}\colon \mathbb{V} \to T(\mathbb{A})$ (return a new cell containing a given value). With the $\mathsf{let}$ construction of Moggi's meta-language (and some infix notation), we can write programs such as (1).

The central contribution of this paper is Theorem 5, where we show that the enriched algebraic theory of local state is complete. This solves a conjecture of Plotkin and Power [20].

For Theorem 5 to hold, the set of storable values must be infinite. When it is finite, the theory of local state is not complete. If there are only two values, then there is an interesting additional equation involving quotient types.

*Aside: limitations of a naive semantics.* The denotational semantics in nominal sets works well at first order, but it is well known that such a naive semantics is limited at higher-order. At higher order, a more careful treatment of functions is needed. Consider the following programs, of type $1 \to T(\mathbb{A} \to \{\mathsf{tt}, \mathsf{ff}\})$ (see e.g. [19]).

$$\mathsf{let}\ \mathrm{a} \Leftarrow \mathsf{ref}(\mathrm{v})\ \mathsf{in}\ (\lambda \mathrm{b}.\ \mathsf{ff}) \qquad \mathsf{let}\ \mathrm{a} \Leftarrow \mathsf{ref}(\mathrm{v})\ \mathsf{in}\ (\lambda \mathrm{b}.\ [\mathrm{a} \stackrel{?}{=} \mathrm{b}]) \qquad (2)$$

(Here, $[(-) \stackrel{?}{=} (-)]\colon \mathbb{A} \to \{\mathsf{tt}, \mathsf{ff}\}$ is the equality test function.) There is a compelling operational argument for considering these expressions to be equivalent: they both allocate a new location (a), but they never reveal it, so it should never be received as an argument. However, in the category of nominal sets, there is an isomorphism $i\colon [\mathbb{A} \to \{\mathsf{tt}, \mathsf{ff}\}] \cong P(\mathbb{A})$ between the nominal set of functions $[\mathbb{A} \to \{\mathsf{tt}, \mathsf{ff}\}]$ and the nominal set $P(\mathbb{A})$ of finite and cofinite sets of atoms, such that $i(\lambda \mathrm{b}.\ [a = b]) = \{a\}$ and $i(\lambda \mathrm{b}.\ \mathsf{ff}) = \emptyset$. There is also a cardinality function $card\colon P(\mathbb{A}) \to \mathbb{N} \uplus \{\omega\}$. The following argument is typical for showing that equations between programs cannot be made into equations in the algebraic theory. If we equate the programs (2) in our theory of local state, we will be able to conclude that

$$\begin{aligned} 0 &= \mathsf{let}\ \mathrm{f} \Leftarrow \big(\mathsf{let}\ \mathrm{a} \Leftarrow \mathsf{ref}(\mathrm{v})\ \mathsf{in}\ (\lambda \mathrm{b}.\ \mathsf{ff})\big)\ \mathsf{in}\ card(i(\mathrm{f})) \\ &= \mathsf{let}\ \mathrm{f} \Leftarrow \big(\mathsf{let}\ \mathrm{a} \Leftarrow \mathsf{ref}(\mathrm{v})\ \mathsf{in}\ (\lambda \mathrm{b}.\ [\mathrm{a} \stackrel{?}{=} \mathrm{b}])\big)\ \mathsf{in}\ card(i(\mathrm{f})) \\ &= 1 \end{aligned}$$

The problem here is that the composite $(card \cdot i) : [\mathbb{A} \to \{tt, ff\}] \to (\mathbb{N} \uplus \{\omega\})$ will not be definable in any reasonable programming language. Authors have used logical relations [3, 4, 15, 18, 22, 26], game semantics [1, 14, 17], and bisimulation [12, 24] to make denotational semantics that can support some higher-order properties.

*Structure.* After recalling background material, I give a new presentation of a theory of local state in Section 4. In Section 5 I prove that the theory of local state is complete when the set of storable values is infinite, and we discuss the situation when the set of values is finite. I conclude in Section 6 by outlining the innovations that were helpful in moving from the theory proposed by Plotkin and Power [20] to the theory in Sec. 4, so that the completeness result could be stated and proved.

## 2 Presentations of enriched algebraic theories and strong monads

We now recall some aspects of the presentation of enriched algebraic theories from the exposition by Kelly and Power [13], simplified and adapted to the needs of this work. In particular, we focus on the case of cartesian structure, rather than arbitrary monoidal structure.

Let $\mathcal{C}$ be a cartesian closed category. A *signature* in $\mathcal{C}$ is a set $Op$, thought of as a set of operators, and an assignment to each operator $op \in Op$ of two objects of $\mathcal{C}$, called the arity and coarity of $op$. When $op \in Op$ has arity $A$ and coarity $B$, we write $(op \colon B \to A) \in Op$.

An *algebra* for a signature $Op$ is an object $X$ of $\mathcal{C}$ together with, for every $(op \colon B \to A) \in Op$, a morphism $op_X \colon B \times X^A \to X$. Homomorphism of algebras is defined in the evident way.

We will only consider signatures $Op$ for which the category of $Op$-algebras is monadic over $\mathcal{C}$. A sufficient condition for this is that $\mathcal{C}$ is a Grothendieck topos.

Every monad $T_{Op}$ arising from a signature $Op$ on $\mathcal{C}$ has a *strength*. For every pair of objects $X$, $Y$, there is a morphism $str_{X,Y} \colon T_{Op}(X) \times Y \to T_{Op}(X \times Y)$ making certain diagrams commute (see e.g. [16, def. 32]).

For any strong monad $T$ on $\mathcal{C}$, every morphism $f \colon B \to T(A)$ induces interpretations in $T$-algebras. The interpretation of $f$ in a $T$-algebra $(X, x \colon T(X) \to X)$ is the following composite:

$$f_{(X,x)} = B \times X^A \xrightarrow{f \times \mathsf{id}} T(A) \times X^A \xrightarrow{str} T(A \times X^A) \xrightarrow{T(eval)} T(X) \xrightarrow{x} X.$$

Informally, $f_{(X,x)}$ takes an element of $B$ and a valuation of $A$ in $X$, and returns an element of $X$.

An *equation* for a monad $T$ is a pair of morphisms $\lambda, \rho \colon B \to T(A)$ with common domain and codomain. The object $B$ is to be thought of as the context of the equation, while $A$ is to be thought of as the type of the variables. A $T$-algebra $(X, x)$ is said to *satisfy* an equation $\lambda, \rho \colon B \to T(A)$ if we have two equal morphisms: $\lambda_{(X,x)} = \rho_{(X,x)} \colon B \times X^A \to X$.

A *theory* in $\mathcal{C}$ is a pair $(Op, Eq)$ of a signature $Op$ and a set of equations $Eq$ for $T_{Op}$. An *algebra* for a theory $(Op, Eq)$ is an $Op$-algebra that satisfies all the equations in $Eq$. We will only consider theories for which the category of $(Op, Eq)$-algebras is monadic over $\mathcal{C}$ (e.g. $\mathcal{C}$ is a Grothendieck topos). The resulting monad $T_{(Op, Eq)}$ is again strong.

## 2.1  Simple meta-language for a strong monad

We will use a variant of Moggi's simple meta-language [16] for reasoning about generalized elements of a strong monad. Let $T$ be a strong monad on a category $\mathcal{C}$ with products.

The types of the meta-language are the objects of $\mathcal{C}$. The terms of the meta-language are built from variables (roman type) and the following grammar:

$$\mathsf{t} ::= \mathsf{let}\ \mathsf{y} : Y \Leftarrow \mathsf{t}\ \mathsf{in}\ \mathsf{t} \mid f(\mathsf{x}_1, \ldots, \mathsf{x}_n) \qquad (\text{for } f : X_1 \times \cdots \times X_n \to T(Y)).$$

When $f = \eta \cdot g$, we will elide $\eta$, the unit of $T$.

A typing context is an assignment of variables to types. The typing rules include structural rules such as

$$\frac{}{\mathsf{x} : X \vdash \mathsf{x} : X} \qquad \text{and} \qquad \frac{\Gamma \vdash \mathsf{t} : Z}{\Gamma, \mathsf{y} : Y \vdash \mathsf{t} : Z}$$

For every morphism $f : X_1 \times \cdots \times X_n \to T(Y)$ in $\mathcal{C}$, we have a well-typed term $\mathsf{x}_1 : X_1, \ldots, \mathsf{x}_n : X_n \vdash f(\mathsf{x}_1, \ldots, \mathsf{x}_n) : Y$. The $\mathsf{let}$ construction is typed by the following rule:

$$\frac{\Gamma \vdash \mathsf{t} : Y \quad \Gamma, \mathsf{y} : Y \vdash \mathsf{u} : Z}{\Gamma \vdash \mathsf{let}\ \mathsf{y} : Y \Leftarrow \mathsf{t}\ \mathsf{in}\ \mathsf{u} : Z}$$

We use the common syntactic sugar, pattern matching in $\mathsf{let}$, writing $(\mathsf{t}; \mathsf{u})$ for $\mathsf{let}\ \_ : Y \Leftarrow \mathsf{t}\ \mathsf{in}\ \mathsf{u}$, etc.

For a typing context $\Gamma = (\mathsf{x}_1 : X_1, \ldots, \mathsf{x}_n : X_n)$, we let $[\![\Gamma]\!] = X_1 \times \cdots \times X_n$. Every typed term-in-context $(\Gamma \vdash \mathsf{t} : X)$ has a semantics in the category $\mathcal{C}$, $[\![\Gamma \vdash \mathsf{t} : X]\!] : [\![\Gamma]\!] \to T(X)$, given by induction on the structure of typing derivations. The interesting case is the $\mathsf{let}$ construction: $\Gamma \vdash \mathsf{let}\ \mathsf{y} : Y \Leftarrow \mathsf{t}\ \mathsf{in}\ \mathsf{u} : Z$ is interpreted as the following composite:

$$[\![\Gamma]\!] \xrightarrow{(\mathsf{id}, [\![\mathsf{t}]\!])} [\![\Gamma]\!] \times T(Y) \xrightarrow{str} T([\![\Gamma]\!] \times Y) \xrightarrow{T([\![\mathsf{u}]\!])} T(T(Z)) \xrightarrow{\mu} T(Z) \qquad .$$

For a monad $T_{Op}$ arising from a signature $Op$, every operation $(op : B \to A) \in Op$ induces a morphism $B \to T_{Op}(A)$. These morphisms can be thought of as "generic effects" in our meta-language.

When two terms are typed in the same context, e.g. $\Gamma \vdash \mathsf{t} : X$ and $\Gamma \vdash \mathsf{u} : X$, then we will write $\Gamma \vdash \mathsf{t} = \mathsf{u} : X$ to indicate that the corresponding morphisms $[\![\mathsf{t}]\!], [\![\mathsf{u}]\!] : [\![\Gamma]\!] \to T(X)$ are equal. There are various sound rules for this notion of equality. For instance,

$$\frac{\Gamma \vdash \mathsf{t}_1 : X_1 \qquad \Gamma \vdash \mathsf{t}_2 : X_2 \qquad \Gamma, \mathsf{x}_1 : X_1, \mathsf{x}_2 : X_2 \vdash \mathsf{t}_3 : X_3}{\Gamma \vdash \big(\mathsf{let}\ \mathsf{x}_2 \Leftarrow (\mathsf{let}\ \mathsf{x}_1 \Leftarrow \mathsf{t}_1\ \mathsf{in}\ \mathsf{t}_2)\ \mathsf{in}\ \mathsf{t}_3\big) = \big(\mathsf{let}\ \mathsf{x}_1 \Leftarrow \mathsf{t}_1\ \mathsf{in}\ (\mathsf{let}\ \mathsf{x}_2 \Leftarrow \mathsf{t}_2\ \mathsf{in}\ \mathsf{t}_3)\big) : X_3}$$

Another important rule is the following substitution law:

$$\frac{\Gamma, x : X \vdash t_1 = t_2 : Y \quad \Gamma \vdash u : X}{\Gamma \vdash [^u/_x]t_1 = [^u/_x]t_2 \; : Y}$$

## 3 Rudiments of nominal sets

We now recall the category of nominal sets. As a category of continuous group actions, it was considered as a base category for local state by Stark [22]. The present formulation is due to Gabbay and Pitts [9].

We begin by fixing an infinite set $\mathbb{A}$ of atoms. In this paper, we think of these atoms as locations in the store. Let $\mathsf{Sym}(\mathbb{A})$ be the group of permutations on $\mathbb{A}$, ranged over by $\pi$. Recall that a $\mathsf{Sym}(\mathbb{A})$-set is a set $X$ together with a function $\mathsf{Sym}(\mathbb{A}) \times X \to X$, such that $(\pi' \cdot \pi) \bullet x = \pi' \bullet (\pi \bullet x)$ and $\mathsf{id} \bullet x = x$.

A finite set of atoms, $A \subseteq_f \mathbb{A}$, is said to support $x \in X$ if whenever $\pi|_A = \mathsf{id}$, $\pi \bullet x = x$. A $\mathsf{Sym}(\mathbb{A})$-set is a *nominal set* if every element has a finite support. In this case, every element $x \in X$ has a smallest supporting set, $supp(x)$.

Nominal sets form a category, **Nom**. A morphism $f : X \to Y$ is an equivariant function, i.e. for all $\pi \in \mathsf{Sym}(\mathbb{A})$ and $x \in X$, we have $f(\pi \bullet x) = \pi \bullet (f(x))$. The category of nominal sets has lots of structure.

- The set $\mathbb{A}$ of atoms is a nominal set, with action $\pi \bullet a = \pi(a)$.
- Any set $X$ can be made into a nominal set with discrete action: for all $x \in X$, let $\pi \bullet x = x$. For example, the terminal nominal set has one element.
- The product of two nominal sets can be made into a nominal set.
- The set of all (not necessarily equivariant) functions $X \to Y$ between two nominal sets has a $\mathsf{Sym}(\mathbb{A})$-set structure given by $(\pi \bullet f)(x) = \pi \bullet (f(\pi^{-1} \bullet x))$. With this structure, not all functions have finite support. We write $[X \to_{\mathsf{fs}} Y]$ or $Y^X$ for the set of finitely supported functions $X \to Y$. This is the cartesian closed structure of the category of nominal sets.
- If $X$ is a nominal set and $R$ is an equivalence relation on $X$ that is equivariant (i.e. $x \; R \; x' \implies (\pi \bullet x) \; R \; (\pi \bullet x')$) then the quotient $X/_R$ has a natural nominal set structure.
- Given two nominal sets $X$, $Y$, we can form the disjoint product:

$$X \otimes Y = \{(x, y) \, | \, x \in X, \; y \in Y, \; supp(x) \cap supp(y) = \emptyset\} \quad .$$

  In particular $\mathbb{A} \otimes \mathbb{A}$ is the set of pairs of distinct atoms. We will write $\mathbb{A}^{\otimes n}$ for the $n$-fold disjoint product of $\mathbb{A}$. The nominal sets $\{\mathbb{A}^{\otimes n} \mid n \in \mathbb{N}\}$ form a generator of **Nom**: if two equivariant functions $f, g : X \to Y$ are different then there is $n \in \mathbb{N}$ and $h : \mathbb{A}^{\otimes n} \to X$ such that $f \cdot h \neq g \cdot h$.

## 4 A theory of local state

A new presentation of the theory of local state is given in Figure 1. It is an algebraic theory enriched in the category of nominal sets. We build it from a

theory of global state, a theory of block, and four additional equations specifying how these theories interact. I have specified the equations for the theory using the syntax for the meta-language for $T_{Op}$. For example, the notation

GS2.  $a : \mathbb{A} \vdash \text{let } v \Leftarrow \ !a \text{ in let } w \Leftarrow \ !a \text{ in } (v, w) \ \approx \ \text{let } v \Leftarrow \ !a \text{ in } (v, v) \ : \mathbb{V} \times \mathbb{V}$

describes two morphisms $\mathbb{A} \to T_{Op}(\mathbb{V} \times \mathbb{V})$.

The theory is parametrized in a set $\mathbb{V}$ of values. We consider $\mathbb{V}$ as a discrete nominal set.

Note that, by equation $B3_1$, and basic properties of nominal sets, we have

$v, w : \mathbb{V} \vdash \text{let } a \Leftarrow \text{ref}(v) \text{ in let } b \Leftarrow \text{ref}(w) \text{ in } [a \overset{?}{=} b] \ = \ \text{ff} \ : \{\text{tt}, \text{ff}\} \quad .$

---

The theory of **global state** has two operations, $\text{lk} \colon \mathbb{A} \to \mathbb{V}$ and $\text{upd} \colon \mathbb{A} \times \mathbb{V} \to 1$. We use infix notation, respectively $!a$ ("*look-up location $a$*") and $a := v$ ("*update location $a$ to $v$*"). There are 7 equations:

GS1. $a : \mathbb{A} \vdash \text{let } v \Leftarrow \ !a \text{ in } a := v \ \approx \ () \ : 1$

GS2. $a : \mathbb{A} \vdash \text{let } v \Leftarrow \ !a \text{ in let } w \Leftarrow \ !a \text{ in } (v, w) \ \approx \ \text{let } v \Leftarrow \ !a \text{ in } (v, v) \ : \mathbb{V} \times \mathbb{V}$

GS3. $a : \mathbb{A}, v, w : \mathbb{V} \vdash a := v; a := w \ \approx \ a := w \ : 1$

GS4. $a : \mathbb{A}, v : \mathbb{V} \vdash a := v; \text{let } w \Leftarrow \ !a \text{ in } w \ \approx \ a := v; v \ : \mathbb{V}$

GS5. $a, b : \mathbb{A} \vdash \text{let } v \Leftarrow \ !a \text{ in let } w \Leftarrow \ !b \text{ in } (v, w)$
$$\approx \quad \text{let } w \Leftarrow \ !b \text{ in let } v \Leftarrow \ !a \text{ in } (v, w) \ : \mathbb{V} \times \mathbb{V}$$

GS6. $(a, b) : \mathbb{A} \otimes \mathbb{A}, v, w : \mathbb{V} \vdash a := v; b := w \ \approx \ b := w; a := v \ : 1$

GS7. $(a, b) : \mathbb{A} \otimes \mathbb{A}, v : \mathbb{V} \vdash a := v; !b \ \approx \ \text{let } w \Leftarrow \ !b \text{ in } a := v; w \ : \mathbb{V}$

The theory of **block** has an operation, $\text{ref}^n \colon \mathbb{A}^{\otimes n} \times \mathbb{V} \to \mathbb{A}^{\otimes (n+1)}$, for every natural number $n \in \mathbb{N}$. Infix, we write, $\text{ref}^n(\vec{a}; v)$; the intuition is "*allocate a new location, different from $\vec{a}$, initialized with $v$*". We use a shorthand: $\text{ref}(v) = \text{ref}^0(\vec{a}; v)$.

There are two equations and one equation schema. For each $n \in \mathbb{N}$, we write $p_n$ for the injection $\mathbb{A}^{\otimes (n+1)} \rightarrowtail \mathbb{A}^{\otimes n} \times \mathbb{A}$.

B1.  $v : \mathbb{V} \vdash \text{let } a \Leftarrow \text{ref}(v) \text{ in } () \ \approx \ () \ : 1$

B2.  $v, w : \mathbb{V} \vdash \text{let } a \Leftarrow \text{ref}(v) \text{ in let } b \Leftarrow \text{ref}(w) \text{ in } (a, b)$
$$\approx \quad \text{let } b \Leftarrow \text{ref}(w) \text{ in let } a \Leftarrow \text{ref}(v) \text{ in } (a, b) \ : \mathbb{A} \times \mathbb{A}$$

$B3_n$. $v : \mathbb{V}, \vec{a} \colon \mathbb{A}^{\otimes n} \vdash \text{let } \vec{b} \Leftarrow \text{ref}^n(\vec{a}; v) \text{ in } p_n(\vec{b}) \ \approx \ \text{let } b \Leftarrow \text{ref}(v) \text{ in } (\vec{a}, b) \ : \mathbb{A}^{\otimes n} \times \mathbb{A}$

The theory of **local state** combines the theory of global state with the theory of block, with 4 additional equations:

LS1. $v, w : \mathbb{V} \vdash \text{let } a \Leftarrow \text{ref}(v) \text{ in } a := w; a \ \approx \ \text{let } a \Leftarrow \text{ref}(w) \text{ in } a \ : \mathbb{A}$

LS2. $v : \mathbb{V} \vdash \text{let } a \Leftarrow \text{ref}(v) \text{ in let } w \Leftarrow \ !a \text{ in } (w, a)$
$$\approx \quad \text{let } a \Leftarrow \text{ref}(v) \text{ in } (v, a) \ : \mathbb{V} \times \mathbb{A}$$

LS3. $a : \mathbb{A}, v, w : \mathbb{V} \vdash \text{let } b \Leftarrow \text{ref}(v) \text{ in } a := w; b \ \approx \ a := w; \text{let } b \Leftarrow \text{ref}(v) \text{ in } b \ : \mathbb{A}$

LS4. $a : \mathbb{A}, v : \mathbb{V} \vdash \text{let } b \Leftarrow \text{ref}(v) \text{ in let } w \Leftarrow \ !a \text{ in } (w, b)$
$$\approx \quad \text{let } w \Leftarrow \ !a \text{ in let } b \Leftarrow \text{ref}(v) \text{ in } (w, b) \ : \mathbb{A}$$

**Fig. 1.** The theory of local state, enriched in the category of nominal sets. The theory is parametrized on a set $\mathbb{V}$ of values.

(We could alternatively reason about the theory of local state in a 'nominal equational logic' (e.g. [5, 6, 8]) but we would then have to restrict to a finite set of values and we would have no guarantee of the strength of the resulting monad.)

## 4.1 Algebras for local state

We now construct algebras for the theory of local state. We begin with algebras for global state; we then consider algebras for block; and finally we combine these ideas to arrive at algebras for local state.

To begin, we consider the nominal set $\mathbb{S} = [\mathbb{A} \to_{\text{fs}} \mathbb{V}]$. These functions are to be thought of as stores. Notice that a function $S \colon \mathbb{A} \to \mathbb{V}$ has support $A \subseteq_{\text{f}} \mathbb{A}$ if and only if there is $v \in \mathbb{V}$ such that for all $a \in (\mathbb{A} \setminus A)$ we have $s(a) = v$. We can think of a store as being initialized to some value, and then subjected to a finite modification.

For any nominal set $X$, a *computation in* $X$ is a finitely supported function $\chi \colon \mathbb{S} \to_{\text{fs}} (\mathbb{S} \times X)$. We write $\chi_1$ and $\chi_2$ for the left and right projections, respectively. The computations form a model of the theory of global state:

- We define an equivariant function $\text{upd}_X \colon \mathbb{A} \times \mathbb{V} \times (\mathbb{S} \times X)^{\mathbb{S}} \to (\mathbb{S} \times X)^{\mathbb{S}}$ as follows: let $(\text{upd}_X(a, v, \chi))(S) = ((\chi_1(S))[^v/a], \chi_2(S))$, where $(\chi_1(S))[^v/a]$ is the store which behaves like $\chi_1(S)$, except that location $a$ maps to $v$.
- We define an equivariant function $\text{lk}_X \colon \mathbb{A} \times \left((\mathbb{S} \times X)^{\mathbb{S}}\right)^{\mathbb{V}} \to (\mathbb{S} \times X)^{\mathbb{S}}$ as follows: let $(\text{lk}_X(a, \bar{\chi}))(S) = (\bar{\chi}(S(a)))(S)$. Here, $\bar{\chi}$ is a finitely supported function $\mathbb{V} \to_{\text{fs}} (\mathbb{S} \times X)^{\mathbb{S}}$.

We are primarily interested in the free model of global state. The structure $(\mathbb{S} \times X)^{\mathbb{S}}$ is not the free model of global state on $X$ because it typically contains too much. For instance, there is a computation in $(\mathbb{S} \times \mathbb{N})^{\mathbb{S}}$ that counts the number of different values in memory. Assuming that the values are numbers, there is a computation in $(\mathbb{S} \times 1)^{\mathbb{S}}$ that adds 7 to every memory cell.

To cut down our model, we say a finite set $A \subseteq_{\text{f}} \mathbb{A}$ *storage-supports* $\chi$ if whenever two stores $S, S' \in \mathbb{S}$ agree on $A$ (i.e. $S|_A = S'|_A$) then we have

1. $(\chi_1(S))|_A = (\chi_1(S'))|_A$; and
2. $\chi_2(S) = \chi_2(S')$; and
3. $(\chi_1(S))|_{\mathbb{A} \setminus A} = S|_{\mathbb{A} \setminus A}$ and $(\chi_1(S'))|_{\mathbb{A} \setminus A} = S'|_{\mathbb{A} \setminus A}$.

Storage-supporting is an equivariant property, and so we can define a nominal set $T_{\text{GS}}(X)$ as follows:

$$T_{\text{GS}}(X) = \{\chi : \mathbb{S} \to_{\text{fs}} (\mathbb{S} \times X) \mid supp(\chi) \text{ storage-supports } \chi\} . \qquad (3)$$

It is straightforward to check that the above model of global state in $(\mathbb{S} \times X)^{\mathbb{S}}$ restricts to a model in $T_{\text{GS}}(X)$.

**Proposition 1.** *For every nominal set $X$, the nominal set $T_{\text{GS}}(X)$ is the free algebra of the theory of global state over $X$.*

Next, we consider the free block algebra $T_{\mathrm{BK}}(X)$ on a nominal set:

$$T_{\mathrm{BK}}(X) \;=\; \{(s, x) \mid x \in X, s \colon supp(x) \rightharpoonup \mathbb{V}\}/_{\simeq} \quad . \tag{4}$$

Here, we are writing $s \colon supp(x) \rightharpoonup \mathbb{V}$ to indicate that $s$ might be partially defined, and $\simeq$ is the equivalence relation generated as follows: if $\pi \in \mathsf{Sym}(\mathbb{A})$ is permutation such that $\pi|_{supp(x)\backslash\mathrm{dom}(s)} = \mathsf{id}$, then $(s, x) \simeq (s \cdot \pi^{-1}, \pi \bullet x)$. The $\mathsf{Sym}(\mathbb{A})$-set structure is given by $\pi \bullet [s, x]_{\simeq} = [s \cdot \pi^{-1}, \pi \bullet x]_{\simeq}$.

- Define an equivariant function $\mathsf{ref}_X^n \colon \mathbb{A}^{\otimes n} \times \mathbb{V} \times (T_{\mathrm{BK}}(X))^{\mathbb{A}^{\otimes (n+1)}} \to T_{\mathrm{BK}}(X)$
  as follows: let

$$\mathsf{ref}_X^n(\vec{a}, v, f) = \begin{cases} [s + (b \mapsto v), x]_{\simeq} & \text{where there is } b \in \mathbb{A} \text{ such that } [s, x]_{\simeq} = f(\vec{a}, b) \\ & \qquad \text{and } b \notin supp(f, \vec{a}), \; b \in supp(x), \; b \notin \mathrm{dom}(s). \\[2mm] [s, x]_{\simeq} & \text{where there is } b \in \mathbb{A} \text{ such that } [s, x]_{\simeq} = f(\vec{a}, b) \\ & \qquad \text{and } b \notin supp(x), \; b \notin supp(f(\vec{a}, b)). \end{cases}$$

It is important to note that the definition of $\mathsf{ref}_X^n(\vec{a}, v, f)$ is independent of the particular choice of $b$.

An intuition for an equivalence class $[s, x]_{\simeq}$ in $T_{\mathrm{BK}}(X)$ is that $s$ is a local store that assigns values to some of the locations involved in $x$. Notice that $supp([s, x]_{\simeq}) = supp(x) \backslash \mathrm{dom}(s)$: those locations that are assigned values are local, so that it doesn't matter if they are renamed.

Now, we can consider free algebras for the full theory of local state:

**Proposition 2.** *The free algebra over $X$ for the theory of local state has carrier $T_{\mathrm{GS}}(T_{\mathrm{BK}}(X))$.*

Recall that the composition of two monads is not a monad unless one can give a distributive law (e.g. [2]). Equations LS1–4 can be understood as defining a distributive law $\delta \colon T_{\mathrm{BK}} \cdot T_{\mathrm{GS}} \to T_{\mathrm{GS}} \cdot T_{\mathrm{BK}}$ (see also [11, §4], [21]): let

$$\bigl(\delta_X[s, \chi]_{\simeq}\bigr)(S) \;=\; \bigl(S'|_{(\mathbb{A}\backslash\mathrm{dom}(s))} \cup S|_{\mathrm{dom}(s)} \;,\; \bigl[S'|_{(\mathrm{dom}(s) \cap supp(x))} \;,\; x\bigr]_{\simeq}\bigr)$$
$$\text{where } (S', x) = \chi(S|_{(\mathbb{A}\backslash\mathrm{dom}(s))} \cup s).$$

## 5  Completeness

We now show that the theory of local state is complete, in the following sense. To make some preliminary definitions, we return to the situation of Section 2.

**Definition 3.** *A theory $(Op, Eq)$ in a cartesian closed category is* complete *if every additional equation $B \rightrightarrows T_{(Op, Eq)}(A)$ is either satisfied in all algebras, or satisfied only in subterminal algebras.*

(Recall that a subterminal object is a subobject of the terminal object. In the category of nominal sets, the only proper subterminal object is the empty set.) Some authors call this property "Hilbert-Post completeness", after Hilbert and Post proved this property of the propositional calculus.

A useful technique for showing that an equation is only satisfied in subterminal algebras is to derive the equation $\vdash \mathsf{tt} = \mathsf{ff} : \{\mathsf{tt}, \mathsf{ff}\}$ from it.

**Lemma 4.** *Let $T$ be a strong monad on a distributive category. The two injections $1 \to T(1+1)$ are equal if and only if all $T$-algebras are subterminal.*

### 5.1 Completeness when the storable values are infinite

**Theorem 5.** *If the set $\mathbb{V}$ of values is infinite then the theory of local state is complete.*

This subsection is devoted to the proof of this theorem.

Consider an equation $\Gamma \vdash \lambda \approx \rho : X$. Suppose that it is not satisfied in the free algebra, $T_{\mathrm{LS}}(X)$, so that $\lambda \neq \rho \colon [\![\Gamma]\!] \to T_{\mathrm{LS}}(X)$. We proceed roughly as follows: by considering the ways that $\lambda$ and $\rho$ could be different, we construct a context $\mathcal{E}$ so that $\mathsf{tt} = \mathcal{E}[\lambda]$ and $\mathsf{ff} = \mathcal{E}[\rho]$. We then use Lemma 4 to conclude that the extra equation $(\lambda \approx \rho)$ is only satisfied in the subterminal models.

It is helpful to make use of Proposition 2 and to prove the theorem in the following two steps.

**Step 1.** We will first prove the following result. Let $(Op, Eq)$ and $(Op', Eq')$ be theories in **Nom**, such that $T_{(Op, Eq)} = T_{\mathrm{GS}} \cdot T_{(Op', Eq')}$. We will assume that the theory $(Op', Eq')$ is complete with respect to $(Op, Eq)$-algebras, that is, that every equation of the form $\Gamma \rightrightarrows T_{(Op', Eq')}(X)$ is either satisfied in all $(Op', Eq')$-algebras, or satisfied only in subterminal $(Op, Eq)$-algebras. From this assumption we will conclude that the theory $(Op, Eq)$ is complete.

**Step 2.** To conclude Theorem 5, we will assume that the nominal set $\mathbb{V}$ of values is infinite, and prove that the theory of block is complete with respect to local state algebras.

Under the hypothesis of Step 1, we consider an equation $\Gamma \vdash \lambda \approx \rho : X$ with $\lambda \neq \rho \colon [\![\Gamma]\!] \to T_{\mathrm{GS}}(T_{(Op', Eq')}(X))$. Since the nominal sets $\{\mathbb{A}^{\otimes n} \mid n \in \mathbb{N}\}$ form a generator, we have $n \in \mathbb{N}$ and an equivariant function $\gamma : \mathbb{A}^{\otimes n} \to [\![\Gamma]\!]$ such that $\lambda \cdot \gamma \neq \rho \cdot \gamma$. Pick an enumeration of distinct atoms $\{b_1, \ldots, b_n\}$. From the characterization of $T_{\mathrm{GS}}$ (see (3)), we know that there must be a store $S_0 \in \mathbb{S}$ with support $\{b_1, \ldots, b_n\}$ such that $\lambda(\gamma(b_1, \ldots, b_n))(S_0) \neq \rho(\gamma(b_1, \ldots, b_n))(S_0)$. Either

$$
\begin{aligned}
& (\pi_1(\lambda \cdot \gamma(b_1, \ldots, b_n)))(S_0) \ \neq\ (\pi_1(\rho \cdot \gamma(b_1, \ldots, b_n)))(S_0) \\
\text{or} \quad & (\pi_2(\lambda \cdot \gamma(b_1, \ldots, b_n)))(S_0) \ \neq\ (\pi_2(\rho \cdot \gamma(b_1, \ldots, b_n)))(S_0) \quad .
\end{aligned}
\tag{5}
$$

In the first case, we have two different stores, both supported by $\{b_1, \ldots, b_n\}$. There must therefore be $i \leq n$ such that we have two different values:

$$((\pi_1(\lambda \cdot \gamma(b_1, \ldots, b_n)))(S_0))(b_i) \ \neq\ ((\pi_1(\rho \cdot \gamma(b_1, \ldots, b_n)))(S_0))(b_i) \text{ in } \mathbb{V}.$$

We now rewrite this observation in the monadic metalanguage. We define a term $\vec{a} : \mathbb{A}^{\otimes n} \vdash \mathbf{S_0}(\vec{a}) : 1$ by $\vec{a} : \mathbb{A}^{\otimes n} \vdash a_1 := S_0(b_1); \ldots a_n := S_0(b_n)$. We have the following equations.

$$\vec{a} : \mathbb{A}^{\otimes n} \vdash \mathbf{S_0}(\vec{a}); (\lambda \cdot \gamma)(\vec{a}); !a_i \approx \mathbf{S_0}(\vec{a}); (\lambda \cdot \gamma)(\vec{a}); (\pi_1(\lambda(\gamma(\vec{b}))(S_0))(b_i))$$

$$\vec{a} : \mathbb{A}^{\otimes n} \vdash \mathbf{S_0}(\vec{a}); (\rho \cdot \gamma)(\vec{a}); !a_i \approx \mathbf{S_0}(\vec{a}); (\rho \cdot \gamma)(\vec{a}); (\pi_1(\rho(\gamma(\vec{b}))(S_0))(b_i))$$

We will use the context $\mathcal{S}nap_{\vec{a}}[-]$:

$\mathcal{S}nap_{\vec{a}}[-] =$
     let $v_1 \Leftarrow !a_1$ in $\ldots$ let $v_n \Leftarrow !a_n$ in let $r \Leftarrow [-]$ in $a_1 := v_1; \ldots a_n := v_n; r$.

This simple context has the property that for any nominal set $X$ and for any term $\vec{a} : \mathbb{A}^{\otimes n} \vdash t(\vec{a}) : 1$,

$$\vec{a} : \mathbb{A}^{\otimes n} \vdash \mathcal{S}nap_{\vec{a}}[t(\vec{a}); \mathsf{ff}] \approx \mathsf{ff} \quad \text{and} \quad \vec{a} : \mathbb{A}^{\otimes n} \vdash \mathcal{S}nap_{\vec{a}}[t(\vec{a}); \mathsf{tt}] \approx \mathsf{tt} \ : \{\mathsf{tt}, \mathsf{ff}\} \quad .$$

So, in this situation we can conclude the following sequence of equations:

$$
\begin{aligned}
\vec{a} : \mathbb{A}^{\otimes n} \vdash \mathsf{tt} &\approx \mathcal{S}nap_{\vec{a}}[\mathbf{S_0}(\vec{a}); (\lambda \cdot \gamma)(\vec{a}); \mathsf{tt}] \\
&\approx \mathcal{S}nap_{\vec{a}}[\mathbf{S_0}(\vec{a}); (\lambda \cdot \gamma)(\vec{a}); [!a_i \stackrel{?}{=} \pi_1(\lambda(\gamma(\vec{b}))(S_0))(b_i)]] \\
&\approx \mathcal{S}nap_{\vec{a}}[\mathbf{S_0}(\vec{a}); (\rho \cdot \gamma)(\vec{a}); [!a_i \stackrel{?}{=} \pi_1(\lambda(\gamma(\vec{b}))(S_0))(b_i)]] \\
&\approx \mathcal{S}nap_{\vec{a}}[\mathbf{S_0}(\vec{a}); (\rho \cdot \gamma)(\vec{a}); \mathsf{ff}] \\
&\approx \mathsf{ff} \quad .
\end{aligned}
$$

At this point, we note that the following rule is valid for the metalanguage in nominal sets, because the projection function $[\![\Gamma]\!] \times \mathbb{A}^{\otimes n} \to [\![\Gamma]\!]$ is always epi:

$$\frac{\Gamma \vdash t : X \quad \Gamma \vdash u : X \quad \Gamma, \vec{a} : \mathbb{A}^{\otimes n} \vdash t = u : X}{\Gamma \vdash t = u : X} \tag{6}$$

We have derived $\vdash \mathsf{tt} = \mathsf{ff}$, and so the subterminal algebras are the only algebras satisfying $\Gamma \vdash \lambda \approx \rho$. This concludes the case where $\pi_1(\lambda(\gamma(\vec{b})))(S_0) \neq \pi_1(\rho(\gamma(\vec{b})))(S_0)$.

For the other case in (5), where $\pi_2(\lambda(\gamma(\vec{b})))(S_0) \neq \pi_2(\rho(\gamma(\vec{b})))(S_0)$, we proceed as follows. We consider the equation

$$\vec{a} : \mathbb{A}^{\otimes n} \vdash \mathcal{S}nap_{\vec{a}}[\mathbf{S_0}(\vec{a}); (\lambda \cdot \gamma)(\vec{a})] \approx \mathcal{S}nap_{\vec{a}}[\mathbf{S_0}(\vec{a}); (\rho \cdot \gamma)(\vec{a})] \ : X \quad .$$

The function described by the left hand side of this equation always returns $\pi_2(\lambda(\gamma(\vec{b})))(S_0)$, leaving the global store unchanged; and the function described by the right hand side of this equation returns $\pi_2(\rho(\gamma(\vec{b})))(S_0)$, leaving the global store unchanged. We thus have two unequal functions $\mathbb{A}^{\otimes n} \to T_{(Op', Eq')}(X)$: an equation that is not satisfied in the free $(Op', Eq')$-algebra $T_{(Op', Eq')}(X)$. By the assumption for Step 1, the only $(Op, Eq)$-algebras that satisfy this equation are subterminal. This concludes Step 1.

We now tackle Step 2: we will show that the theory of block is complete with respect to $T_{\text{LS}}$-algebras. We consider a pair of distinct equivariant functions $\lambda, \rho \colon \varGamma \to T_{\text{BK}}(X)$ and show that this equation is only satisfied in subterminal $T_{\text{LS}}$-algebras. As above, we have $n \in \mathbb{N}$ and $\gamma \colon \mathbb{A}^{\otimes n} \to \varGamma$ such that $\lambda \cdot \gamma \neq \rho \cdot \gamma$.

We begin by setting up some notation. For any nominal set $Y$, a natural number $m$, and a permutation group $G < \mathsf{Sym}(m)$, we write $Y^m/_G$ for the nominal set of $n$-tuples $(y_1, \ldots, y_m)$ up to the equivalence relation generated by $(y_1, \ldots, y_m) \sim_G (y_{\pi(1)}, \ldots, y_{\pi(m)})$ for $\pi \in G$.

The nominal set $X$ admits the following analysis (as does every nominal set). There is an ordinary set $Orb$ and for each $o \in Orb$, a natural number $m_o$ and a permutation group $G_o < \mathsf{Sym}(m_o)$, together with an isomorphism

$$i \colon X \cong \coprod_{o \in Orb} \mathbb{A}^{\otimes m_o}/_{G_o} \quad .$$

We write $orb \colon X \to Orb$ for the evident projection function. This is a version of the orbit-stabilizer theorem: $Orb$ is the set of orbits of $X$; and the finite groups $G_o$ generate the stabilizers. This characterization forms the correspondence between nominal sets and named sets with symmetry [7, 10].

We now pick an enumeration of distinct atoms $\{b_1, \ldots, b_n\}$, and we pick representatives $(s, x)$ and $(s', x')$ of the $\simeq$-equivalence classes $\lambda(\gamma(\vec{b}))$ and $\rho(\gamma(\vec{b}))$ respectively. Without loss of generality, by the definition of $\simeq$, we assume that $\mathrm{dom}(s) \cap supp(x') = \emptyset = \mathrm{dom}(s') \cap supp(x)$.

We proceed differently depending on whether $x$ and $x'$ are in the same orbit, whether $orb(x) = orb(x')$. If $orb(x) \neq orb(x')$, then we have the following sequence of equations in any algebra satisfying $(\lambda \approx \rho)$.

$$\vec{a} \colon \mathbb{A}^{\otimes n} \vdash \mathsf{tt} \approx \mathsf{let}\ \mathrm{r} \Leftarrow (\lambda \cdot \gamma)(\vec{a})\ \mathsf{in}\ [orb(\mathrm{r}) \overset{?}{=} orb(x)]$$

$$\approx \mathsf{let}\ \mathrm{r} \Leftarrow (\rho \cdot \gamma)(\vec{a})\ \mathsf{in}\ [orb(\mathrm{r}) \overset{?}{=} orb(x)]\ \approx \mathsf{ff}\ : \{\mathsf{tt}, \mathsf{ff}\} \quad .$$

If $x$ and $x'$ are in the same orbit then we proceed as follows. It is at this point that we make use of the fact that the set $\mathbb{V}$ of values is infinite: pick $n$ distinct values $v_1, \ldots, v_n$ that lie outside the ranges of $s$ and $s'$. We define a partial function $s'' \colon \mathbb{A} \rightharpoonup \mathbb{V}$ as follows:

| | |
|---|---|
| *For $i \leq n$:* | $s''(b_i) = v_i$ |
| *For $b \in \mathrm{dom}(s)$:* | $s''(b) = s(b)$ |
| *For $b' \in \mathrm{dom}(s')$:* | $s''(b') = s'(b')$ |

We write $\bar{o}$ for $orb(x)$, and let $\mathsf{in}_{\bar{o}}[b_1 \ldots b_{m_{\bar{o}}}]_{G_{\bar{o}}} = i(x)$ and $\mathsf{in}_{\bar{o}}[b'_1 \ldots b'_{m_{\bar{o}}}]_{G_{\bar{o}}} = i(x')$. Notice that $\mathrm{dom}(s'') = supp(x) \cup supp(x')$, and so the function $s''$ is defined at $b_i$ and $b'_i$ for every $i \leq m_{\bar{o}}$. Crucially, the tuple-quotients

$$[s''(b_1), \ldots, s''(b_{m_{\bar{o}}})]_{G_{\bar{o}}} \text{ and } [s''(b'_1), \ldots, s''(b'_{m_{\bar{o}}})]_{G_{\bar{o}}} \text{ in } \mathbb{V}^{m_{\bar{o}}}/_{G_{\bar{o}}} \qquad (7)$$

are different, because $[s, x]_{\simeq} \neq [s', x']_{\simeq}$. In what follows, we abbreviate (7) by writing $[\vec{w}]_{G_{\bar{o}}}$ for the left hand tuple and $[\vec{w}']_{G_{\bar{o}}}$ for the right hand tuple.

We now translate these observations into the monadic metalanguage. We consider a derived effect

$$\mathsf{lk}\colon \coprod_{o\in Orb} (\mathbb{A}^{m_o}/_{G_o}) \longrightarrow T_{\mathrm{LS}}\Big( \coprod_{o\in Orb} (\mathbb{V}^{m_o}/_{G_o})\Big) \quad .$$

Informally, $\mathsf{lk}(\mathsf{in}_o[a_1,\ldots,a_{m_o}]_{G_o})$ returns the result of simultaneously looking up the values in locations $a_1,\ldots,a_{m_o}$. Formally, it is the unique equivariant function making the following diagram commute:

$$
\begin{array}{ccc}
\coprod_{o\in Orb}(\sim_{G_o}) & \longrightarrow & \coprod_{o\in Orb}(\sim_{G_o}) \\
\downarrow\downarrow & & \downarrow\downarrow \\
\coprod_{o\in Orb}(\mathbb{A}^{m_o}) & \xrightarrow{\;(\star)\;} & T_{\mathrm{LS}}\big(\coprod_{o\in Orb}\mathbb{V}^{m_o}\big) \\
\downarrow & & \downarrow \\
\coprod_{o\in Orb}(\mathbb{A}^{m_o}/_{G_o}) & \cdots\cdots\!\!\xrightarrow{\;\mathsf{lk}\;}\!\!\!\!> & T_{\mathrm{LS}}\big(\coprod_{o\in Orb}(\mathbb{V}^{m_o}/_{G_o})\big)
\end{array}
\tag{8}
$$

The arrow labelled $(\star)$ corresponds to the $Orb$-fold coproduct of terms

$$\vec{a}\colon \mathbb{A}^{m_o} \vdash \mathsf{let}\ v_1 \Leftarrow !a_1\ \mathsf{in}\ \ldots \mathsf{let}\ v_{m_o} \Leftarrow !a_{m_o}\ \mathsf{in}\ (v_1,\ldots,v_{m_o})\colon \mathbb{V}^{m_o}$$

and the upper diagrams commute by axiom GS5. Informally, it does not matter which order the locations are read, and so the $G_o$-equivalence classes are respected.

In any $T_{\mathrm{LS}}$-algebra satisfying $(\lambda \approx \rho)$, we have

$$\vec{a}\colon \mathbb{A}^{\otimes n} \vdash\ \mathsf{tt} \approx \mathsf{let}\ r \Leftarrow (\lambda\cdot\gamma)(\vec{a})\ \mathsf{in}\ a_1 := v_1;\ \ldots a_n := v_n;\ [\mathsf{lk}(i(r)) \overset{?}{=} \mathsf{in}_{\bar{o}}[\vec{w}]_{G_{\bar{o}}}]$$

$$\approx \mathsf{let}\ r \Leftarrow (\rho\cdot\gamma)(\vec{a})\ \mathsf{in}\ a_1 := v_1;\ \ldots a_n := v_n;\ [\mathsf{lk}(i(r)) \overset{?}{=} \mathsf{in}_{\bar{o}}[\vec{w}]_{G_{\bar{o}}}]$$

$$\approx \mathsf{ff}\ \ :\{\mathsf{tt},\mathsf{ff}\}\quad .$$

Using (6), we complete Step 2 and finish our proof of Theorem 5.

## 5.2 An additional axiom when the set of values is finite

The proof of Theorem 5 relies on the hypothesis that the set of values is infinite. This is the situation considered by Plotkin and Power in [20].

There are many applications where the set of storable values is finite. If there is only one value, so we have a model of the $\nu$-calculus [19], the theory is complete, and the proof can be adapted: the theory of block remains complete with respect to the theory of local state. If $\mathbb{V}$ is empty, then all algebras are terminal. If $\mathbb{V}$ is finite and has more than one element, then the theorem fails. We will give a counter-example in the case where $\mathbb{V} = \{\mathsf{tt},\mathsf{ff}\}$. Consider the quotient nominal set $\mathbb{A}^3/_{\mathrm{C}_3}$, with $(a,b,c) \sim_{\mathrm{C}_3} (b,c,a)$. Then the following computations are distinguished in $T_{\mathrm{LS}}(\mathbb{A}^3/_{\mathrm{C}_3})$, but they are equated in some non-trivial $T_{\mathrm{LS}}$-algebras.

$$
\begin{aligned}
c : \mathbb{A} \vdash \mathsf{let}\ a \Leftarrow \mathsf{ref}(\mathsf{tt})\ \mathsf{in}\ \mathsf{let}\ b \Leftarrow \mathsf{ref}(\mathsf{ff})\ \mathsf{in}\ [a,b,c]_{\mathrm{C}_3}\ &:\ \mathbb{A}^3/_{\mathrm{C}_3} \\
c : \mathbb{A} \vdash \mathsf{let}\ a \Leftarrow \mathsf{ref}(\mathsf{ff})\ \mathsf{in}\ \mathsf{let}\ b \Leftarrow \mathsf{ref}(\mathsf{tt})\ \mathsf{in}\ [a,b,c]_{\mathrm{C}_3}\ &:\ \mathbb{A}^3/_{\mathrm{C}_3}
\end{aligned}
\tag{L5}
$$

Notice that if we add an additional constant $\star$ to $\mathbb{V}$, then these terms *are* distinguished in all algebras, using the context $\mathcal{E}[-] = \big(\mathsf{c} := \star;\ \mathsf{let}\ \mathsf{r} \Leftarrow [-]\ \mathsf{in}\ \mathsf{lk}_{C_3}(\mathsf{r})\big)$. Here, the effect $\mathsf{lk}_{C_3} \colon \mathbb{A}^3/_{C_3} \to T_{\mathrm{LS}}(\mathbb{V}^3/_{C_3})$ is to be defined as in (8), in the previous subsection.

The theory of local state is thus not complete, in the sense of Defn. 3, when the set of values is finite. This notion of completeness is arguably too strong, because the nominal set $(\mathbb{A}^3/_{C_3})$ would never arise as the denotation of an intensional type.

Recall from the work of Tzevelekos [25] that a nominal set $X$ is said to be *strong* if, for all $x \in X$ and $a \in \mathit{supp}(x)$, if $\pi \bullet x = x$ then $\pi(a) = a$. Equivalently, a nominal set is strong if it is isomorphic to a coproduct of nominal sets of the form $\mathbb{A}^{\otimes n}$. Arguably, the denotations of first-order intensional types are always strong nominal sets.

**Theorem 6.** *Let $X$ be a strong nominal set. Every equation that is of the form $\Gamma \rightrightarrows T_{\mathrm{LS}}(X)$ is either satisfied in all $T_{\mathrm{LS}}$-algebras, or satisfied only in subterminal $T_{\mathrm{LS}}$-algebras.*

This result is proved in much the same way as Theorem 5. The crucial lemma is a refined form of Step 2 in that proof.

> *Every equation of the form $\Gamma \rightrightarrows T_{\mathrm{BK}}(X)$, with $X$ a strong nominal set, is either satisfied in all $T_{\mathrm{BK}}$-algebras, or satisfied only in subterminal $T_{\mathrm{LS}}$-algebras.*

## 6 Comparison with the theory of Plotkin and Power

Plotkin and Power [20] propose a theory for local state, which is the starting point for the present paper. We conclude this paper by making precise the relationship between their theory and ours. Throughout this section, we suppose that $\mathbb{V}$ is a countably infinite set.

Let $\mathbf{I}$ be the category of natural numbers and injections between them, and consider the category $[\mathbf{I}, \mathbf{Set}]$ of covariant presheaves $\mathbf{I} \to \mathbf{Set}$ and natural transformations between them. We can define a presheaf $\mathbb{A}$ of locations by $\mathbb{A}(n) = n$. There is an endofunctor $\delta$ on $[\mathbf{I}, \mathbf{Set}]$, given by $(\delta X)(n) = X(n+1)$.

**Definition 7.** *A Plotkin-Power algebra for local state is a presheaf $X \colon \mathbf{I} \to \mathbf{Set}$ together with three natural transformations*

$$l \colon X^{\mathbb{A}} \to X^{\mathbb{V}} \qquad u \colon X \to X^{\mathbb{A} \times \mathbb{V}} \qquad b \colon \delta X \to X^{\mathbb{V}}$$

*subject to 13 commuting diagrams [20, Sec. 4].*

**Theorem 8.** *The category of Plotkin-Power algebras for local state is equivalent to the category of algebras for the monad $T_{\mathrm{LS}}$ on $\mathbf{Nom}$ (as in Section 4).*

The theory of Plotkin-Power algebras appears very similar to the theory in Figure 1, but there are two subtle points, outlined in the following proof sketch.

*Every block-algebra preserves pullbacks.* The first discrepancy between Fig. 1 and Defn. 7 is that the carrier of an algebra in Defn. 7 is a presheaf in $[\mathbf{I}, \mathbf{Set}]$, rather than a nominal set. It is well known that the category **Nom** of nominal sets is equivalent to the category of pullback-preserving-functors $\mathbf{I} \to \mathbf{Set}$, but not every presheaf $\mathbf{I} \to \mathbf{Set}$ preserves pullbacks.

The proof of completeness (Sec. 5) is significantly simplified by working in the category of nominal sets. Equality is decidable, and we have the principle (6).

Define a *block algebra* to be a presheaf $X$ together with a natural transformation $\delta X \to X^{\mathbb{V}}$, such that the relevant diagrams from [20, 21, def. 5.1] commute.

**Proposition 9.** *The carrier of every block algebra preserves pullbacks.*

One way to prove Prop. 9 is to consider a small category $\mathbf{B}_{\mathbb{V}}$ whose objects are natural numbers, and where a morphism $f\colon n \to m$ is a function $f\colon n \to (m \uplus \mathbb{V})$ that is injective on $f^{-1}(m)$. Composition is in the style of a Kleisli category. The important observation is that the category of block algebras is equivalent to the category of functors $[\mathbf{B}_{\mathbb{V}}, \mathbf{Set}]$. Indeed, the inclusion $\mathbf{I} \to \mathbf{B}_{\mathbb{V}}$ induces a monadic forgetful functor $[\mathbf{B}_{\mathbb{V}}, \mathbf{Set}] \to [\mathbf{I}, \mathbf{Set}]$. We will say that a morphism $f\colon n \to m$ in $\mathbf{B}_{\mathbb{V}}$ is *total* if $\mathrm{im}(f) \subseteq m$. Pullbacks of total morphisms in $\mathbf{B}_{\mathbb{V}}$ are absolute, in the sense that they are preserved by every functor.

*An alternative presentation of block.* The presentation of Defn. 7 is not an enriched algebraic theory as in Section 2, because it involves the operation $b\colon \delta X \to X^{\mathbb{V}}$. Plotkin and Power are able to give a strength for the resulting monad by hand, but they leave open the problem of finding an algebraic presentation and hence they do not have all the effects in their meta-language.

The following result plays a crucial role in the proof of Theorem 8. The first datum corresponds to the Plotkin-Power block operator, while the second corresponds to our family $\{\mathsf{ref}^n\}$ of operators, with equations B3$_n$ (Fig. 1). We write $\mathbb{A}^{\otimes n}$ for the representable functor $\mathbf{I}(n, -)\colon \mathbf{I} \to \mathbf{Set}$.

**Proposition 10.** *Let $X$, $Z$ be presheaves in $[\mathbf{I}, \mathbf{Set}]$. The following data are equivalent.*

1. *A natural transformation $\delta X \to X^Z$*
2. *A family of natural transformations $\{\beta^n : X^{\mathbb{A}^{\otimes(n+1)}} \to X^{(\mathbb{A}^{\otimes n} \times Z)}\}_{n \in \mathbb{N}}$ making the following diagrams commute:*

$$X^{(\mathbb{A}^{\otimes n} \times \mathbb{A})} \xrightarrow{\ X^{p_n}\ } X^{\mathbb{A}^{\otimes(n+1)}} \qquad \text{for every } n \in \mathbb{N}.$$

$$\Big\downarrow^{(\beta^n)}$$

$$(\beta^0)^{\mathbb{A}^{\otimes n}} \searrow \quad X^{(\mathbb{A}^{\otimes n} \times Z)}$$

I present an alternative solution in [23], but using a more generous enrichment for which the completeness theorem (Thm. 5) does *not* hold.

# References

1. S. Abramsky, D. R. Ghica, A. S. Murawski, C.-H. L. Ong, and I. D. B. Stark. Nominal games and full abstraction for the nu-calculus. In *LICS'04*, 2004.
2. M. Barr and C. Wells. *Toposes, Triples and Theories*. Springer-Verlag, 1984.
3. N. Benton and B. Leperchey. Relational reasoning in a nominal semantics for storage. In *Proc. of TLCA'05*, pages 86–101, 2005.
4. L. Birkedal, K. Støvring, and J. Thamsborg. Realizability semantics of parametric polymorphism, references, and recursive types. In *Proc. of FoSSaCS'09*, 2009.
5. R. A. Clouston and A. M. Pitts. Nominal equational logic. In *Articles dedicated to Gordon Plotkin*, volume 172 of *ENTCS*, pages 223–257. 2007.
6. M. P. Fiore and C.-K. Hur. Term equational systems and logics. In *Proc. of MFPS XXIV*, volume 218 of *Electron. Notes Theor. Comput. Sci.*, pages 171–192, 2008.
7. M. P. Fiore and S. Staton. Comparing operational models of name-passing process calculi. *Inform. and Comput.*, 204(4):435–678, 2006.
8. M. Gabbay and A. Mathijssen. A formal calculus for informal equality with binding. In *Proc. of WoLLIC'07*, pages 162–176, 2007.
9. M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2001.
10. F. Gadducci, M. Miculan, and U. Montanari. About permutation algebras, (pre)sheaves and named sets. *Higher-Order Symb. Comput.*, 19(2–3):283–304, 2006.
11. M. Hyland, G. D. Plotkin, and J. Power. Combining effects: Sum and tensor. *Theoret. Comput. Sci.*, 357:70–99, 2006.
12. A. Jeffrey and J. Rathke. Towards a theory of bisimulation for local names. In *Proc. of LICS'99*, pages 56–66, 1999.
13. G. M. Kelly and A. J. Power. Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads. *J. Pure Appl. Algebra*, 89(1–2):163–179, 1993.
14. J. Laird. A game semantics of names and pointers. *Ann. Pure Appl. Logic*, 151(2–3):151–169, 2008.
15. R. E. Møgelberg. A nominal relational model for local variables. Draft, 2009.
16. E. Moggi. Notions of computation and monads. *Inform. and Comput.*, 93(1):55–92.
17. A. S. Murawski and N. Tzevelekos. Full abstraction for Reduced ML. In *Proc. of FoSSaCS'09*, volume 5504 of *Lecture Notes in Comput. Sci.*, pages 32–47, 2009.
18. P. W. O'Hearn and R. D. Tennant. Parametricity and local variables. *J. ACM*, 42(3):658–709, 1995.
19. A. Pitts and I. Stark. Observable properties of higher order functions that dynamically create local names, or: What's new? In *Proc. of MFCS'93*, 1993.
20. G. D. Plotkin and J. Power. Notions of computation determine monads. In *Proc. of FoSSaCS'02*, volume 2303 of *Lecture Notes in Comput. Sci.*, pages 342–356, 2002. Erratum at http://homepages.inf.ed.ac.uk/gdp/publications/.
21. J. Power. Semantics for local computational effects. In *Proc. of MFPS XXII*, 2006.
22. I. Stark. Categorical models for local names. *LISP and Symbolic Computation*, 9(1):77–107, 1996.
23. S. Staton. Two cotensors in one: Presentations of algebraic theories for local state and fresh names. In *Proc. of MFPS XXV*, 2009.
24. E. Sumii. A complete characterization of observational equivalence in polymorphic $\lambda$-calculus with general references. In *Proc. of CSL'09*, pages 455–469, 2009.
25. N. Tzevelekos. Full abstraction for nominal general references. *Logical Methods in Computer Science*, 5(3), 2009.
26. Y. Zhang and D. Nowak. Logical relations for dynamic name creation. In *Proc. of CSL'03*, pages 575–588, 2003.