Nominal Kleene Coalgebra

Dexter Kozen* Konstantinos Mamouras*
Daniela Petrisan† Alexandra Silva†

February 18, 2015

Abstract

We develop the coalgebraic theory of nominal Kleene algebra, including an alternative language-theoretic semantics, a nominal extension of the Brzozowski derivative, and a bisimulation-based decision procedure for the equational theory.

1 Introduction

Nominal Kleene algebra, introduced by Gabbay and Ciancia [12], is an algebraic formalism for reasoning equationally about imperative programs with statically scoped allocation and deallocation of resources. The system consists of Kleene algebra, the algebra of regular expressions, augmented with a binding operator ν that binds a named resource within a local scope.

Gabbay and Ciancia [12] proposed an axiomatization of the system consisting of the axioms of Kleene algebra plus six equations capturing the behavior of the binding operator ν and its interaction with the Kleene algebra operators. They also defined a family of *nominal languages* consisting of certain sets of strings over an infinite alphabet satisfying certain invariance properties and showed soundness of the axioms over this class of interpretations. Their analysis revealed some surprising subtleties arising from the non-compositionality of the sequential composition and iteration operators.

In our previous work [15] we showed that the Gabbay-Ciancia axioms are not complete for the semantic interpretation of [12], but we identified a slightly wider class of language models over which they are sound and complete. The proof of completeness of [15] consisted of several stages of transformations to bring expressions to a certain normal form. Although the construction was effective, one of the transformations required the intersection of several regular

*Computer Science, Cornell University, Ithaca, New York 14853-7501, USA. http://www.cs.cornell.edu/~kozen/, http://www.cs.cornell.edu/~mamouras/

†Intelligent Systems, Radboud University Nijmegen, Postbus 9010, 6500 GL Nijmegen, The Netherlands. http://alexandrasilva.org, http://www.cs.ru.nl/D.Petrisan/

expressions, an operation known to produce a double-exponential increase in size in the worst case [13], thus the construction is unlikely to give a practical decision method.

In this paper, we investigate the coalgebraic theory of nominal Kleene algebra. The motivation for this investigation is to understand the structure of nominal Kleene algebra from a coalgebraic perspective with an eye toward a more efficient decision procedure for the equational theory in the style of [4, 5, 22, 23] for Kleene algebra and Kleene algebra with tests.

The paper is organized as follows. In §3 we introduce a new class of language models consisting of sets of equivalence classes of *ν-strings*. A ν-string is like a string, except that it may contain binding operators. Two ν-strings are equivalent if they are provably so under the Gabbay-Ciancia axioms and associativity. The equivalence classes of ν-strings over a fixed set of variables form a nominal monoid. These language models are isomorphic to the free language models of [15], thus giving a new characterization of the free models, but more amenable for the development of the coalgebraic theory. The proof of isomorphism is given in Appendix A.

In §4 we introduce nominal versions of the semantic and syntactic Brzozowski derivatives. The derivatives are similar to their non-nominal counterparts, but extended to handle bound variables in such a way as to be invariant with respect to α-conversion. The semantic derivative is defined in terms of the new language model and characterizes the final coalgebra. We conclude the section with a result that relates the algebraic and coalgebraic structure and establishes the existence of minimal automata.

In §5 we describe a data representation for the efficient calculation of the Antimirov derivative. The data representation is similar to that used in [7, 14] and also related to constructions of [2, 25], but extended to handle the binding operator. The advantage of this representation is that data representing repeated derivatives can be calculated once and for all in a preprocessing step; thereafter, the derivatives are easily computed by table lookup. The preprocessing step also gives a bound on the size of automata, and we use this to prove that the equational theory is decidable in exponential space. The data representation also provides a platform for the implementation of a bisimulation-based decision procedure in the style of [4, 5, 22, 23] for Kleene algebra and Kleene algebra with tests.

Related Work The notion of nominal sets goes back to work of Fraenkel and Mostowski in the early part of the twentieth century. The notion was first applied in computer science by Gabbay and Pitts [10] (see [21] for a survey).

Recently, there have been many studies involving nominal automata, automata on infinite alphabets, and regular expressions with binders that are closely related to the work presented here.

Montanari and Pistore [18, 19, 20] and Ferrari et al. [6] develop the theory of *history-dependent (HD) automata*, an operational model for process calculi such as the π-calculus. In these automata, there are mechanisms for explicit

allocation and deallocation of names and for explicitly representing the history of allocated names. They work in a category of *named sets*, which except for presentation is essentially equivalent to the category of nominal sets.

A closely related model is the family of *finite memory automata* of Francez and Kaminski [8, 9]. These are ordinary finite-state automata equipped with a finite set of *registers*. At any point in time, each register is either empty or contains a symbol from an infinite alphabet. In each step, the automaton can copy a symbol to a register, compare the contents of a register with an input symbol, and reset a register to empty. The main result is an extension of the Myhill–Nerode theorem for finite memory automata for languages that are invariant under permutations of the infinite alphabet.

Bojanczyk, Klin, Lasota [3] undertake a comprehensive study of nominal automata and discuss the relationships between previous models. They consider nominal sets for arbitrary symmetries and develop nominal automata theory in this framework. They identify the important notion of *orbit-finiteness* as the appropriate analog of finiteness in the non-nominal case and show that their definitions are equivalent to previous definitions of finite memory automata [8, 9]. They prove a nominal analog of the Myhill-Nerode theorem. Their paper does not consider the relationship with regular expressions.

Kurz, Suzuki, Tuosto [16, 17] present a syntax of regular expressions with binders and consider its relationship with nominal automata. Their syntax departs from that of Gabbay and Ciancia in that they include includes operational mechanisms for the dynamic allocation and deallocation of fresh names and explicit permutations. Their semantics uses a name-independent combinatorial construct reminiscent of De Bruijn indices. They prove Kleene theorems relating the syntax and semantics.

The most important distinguishing characteristic of our approach is that both the algebraic and coalgebraic structure are nominal. Our syntax, based on Kleene algebra with ν-binders as introduced by Gabbay and Ciancia [12], and our final coalgebra semantics based on nominal sets of ν-strings, both carry a nominal coalgebraic structure given by the syntactic and semantic Brzozowski derivatives, and the interpretation map is the unique equivariant morphism to the final coalgebra.

2 Background

This section contains an abbreviated review of basic material on Kleene algebra, nominal sets, and the nominal extension of Kleene algebra (NKA) introduced by Gabbay and Ciancia [12], but prior familiarity with nominal sets, KA, and coalgebra will be helpful. For a more thorough introduction, the reader is referred to [11, 21] for nominal sets, to [24] for Kleene (co)algebra, and to [12, 15] for NKA.

Kleene Algebra (KA) is the algebra of regular expressions. A *Kleene algebra* is any structure $(K, +, \cdot, ^*, 0, 1)$ where K is a set, + and $\cdot$ are binary operations

on K, * is a unary operation on K, and 0 and 1 are constants, satisfying the following axioms:

$$\begin{array}{lll} x+(y+z)=(x+y)+z & x(yz)=(xy)z & x+y=y+x \\ 1x=x1=x & x+0=x+x=x & x0=0x=0 \\ x(y+z)=xy+xz & (x+y)z=xz+yz & 1+xx^*\leq x^* \\ y+xz\leq z \;\Rightarrow\; x^*y\leq z & y+zx\leq z \;\Rightarrow\; yx^*\leq z & 1+x^*x\leq x^* \end{array}$$

where we define $x \leq y$ iff $x+y=y$. The axioms above not involving * are succinctly stated by saying that the structure is an idempotent semiring under $+, \cdot, 0$, and 1, the term *idempotent* referring to the axiom $x+x=x$. Due to this axiom, the ordering relation $\leq$ is a partial order. The axioms for * together say that x^*y is the $\leq$-least z such that $y+xz\leq z$ and yx^* is the $\leq$-least z such that $y+zx\leq z$.

G-Sets A *group action* of a group G on a set X is a map $G\times X\to X$, written as juxtaposition, such that $\pi(\rho x)=(\pi\rho)x$ and $1x=x$ for $\pi,\rho\in G$ and $x\in X$. A *G-set* is a set X equipped with a group action $G\times X\to X$. The *orbit* of an element $x\in X$ is the set $\{\pi x \mid \pi\in G\}\subseteq X$. If X and Y are two G-sets, a function $f: X\to Y$ is called *equivariant* if $f\circ\pi=\pi\circ f$ for all $\pi\in G$.

The G-sets and equivariant functions form an elementary topos G-Set with group action on coproducts, products, and exponentials defined by

$$\pi(\mathsf{in}\,x)=\mathsf{in}(\pi x)\quad \pi(x,y)=(\pi x,\pi y)\quad \pi()=()\quad \pi f=\pi\circ f\circ\pi^{-1}. \quad (1)$$

In particular, for sets, $\pi A=\{\pi x \mid x\in A\}$. For $x\in X$ and $A\subseteq X$, define

$$\mathsf{fix}\,x=\{\pi\in G\mid \pi x=x\} \qquad \mathsf{Fix}\,A=\textstyle\bigcap_{x\in A}\mathsf{fix}\,x.$$

Note that $\mathsf{Fix}\,A$ and $\mathsf{fix}\,A$ are different: they are the subgroups of G that fix A pointwise and setwise, respectively.

Nominal Sets Fix a countably infinite set $\mathbb{A}$ of *atoms* and let $G_{\mathbb{A}}$ be the group of all finite permutations of $\mathbb{A}$ (permutations generated by transpositions $(a\ b)$). The set $\mathbb{A}$ is a $G_{\mathbb{A}}$-set under the group action $\pi a=\pi(a)$. If X is another $G_{\mathbb{A}}$-set, we say that $A\subseteq\mathbb{A}$ *supports* $x\in X$ if $\mathsf{Fix}\,A\subseteq\mathsf{fix}\,x$. An element $x\in X$ has *finite support* if there is a finite set $A\subseteq\mathbb{A}$ that supports x. If x has finite support, then there is a smallest set supporting x, called $\mathsf{supp}\,x$. We write $a\#x$ and say a is *fresh for* x if $a\notin\mathsf{supp}\,x$. A *nominal set* is a $G_{\mathbb{A}}$-set X of which every element has finite support. The nominal sets and equivariant functions form a full subcategory Nom of G-Set.

The following lemma reviews some well known facts about nominal sets and equivariant functions. Let $\wp_{\mathsf{fin}}\,\mathbb{A}$ denote the set of finite subsets of $\mathbb{A}$.

Lemma 2.1

(i) $\mathsf{fix}\,\pi x=\pi(\mathsf{fix}\,x)\pi^{-1}$ *and* $\mathsf{Fix}\,\pi A=\pi(\mathsf{Fix}\,A)\pi^{-1}$.

(ii) $\mathsf{supp} : X \to \wp_{\mathrm{fin}} \mathbb{A}$ *is equivariant:* $\mathsf{supp}\, \pi x = \pi(\mathsf{supp}\, x)$.

(iii) $\mathsf{fix}\, x \subseteq \mathsf{fix}\, \mathsf{supp}\, x$.

(iv) *If* $A, B \in \wp_{\mathrm{fin}} \mathbb{A}$ *and* $\mathsf{Fix}\, B \subseteq \mathsf{fix}\, A$, *then* $A \subseteq B$.

(v) *If* f *is an equivariant function, then* $\mathsf{supp}\, f(x) \subseteq \mathsf{supp}\, x$.

Proof. (i)

$$\rho \in \mathsf{fix}\, \pi x \Leftrightarrow \rho\pi x = \pi x \Leftrightarrow \pi^{-1}\rho\pi x = x \Leftrightarrow \pi^{-1}\rho\pi \in \mathsf{fix}\, x \Leftrightarrow \rho \in \pi(\mathsf{fix}\, x)\pi^{-1},$$
$$\mathsf{Fix}\, \pi A = \bigcap_{x \in A} \mathsf{fix}\, \pi x = \bigcap_{x \in A} \pi(\mathsf{fix}\, x)\pi^{-1} = \pi(\bigcap_{x \in A} \mathsf{fix}\, x)\pi^{-1} = \pi(\mathsf{Fix}\, A)\pi^{-1}.$$

(ii)

$$\mathsf{Fix}\, \mathsf{supp}\, x \subseteq \mathsf{fix}\, x \Rightarrow \pi(\mathsf{Fix}\, \mathsf{supp}\, x)\pi^{-1} \subseteq \pi(\mathsf{fix}\, x)\pi^{-1} \Rightarrow \mathsf{Fix}\, (\pi\, \mathsf{supp}\, x) \subseteq \mathsf{fix}\, \pi x.$$

As $\mathsf{supp}\, \pi x$ is the smallest set supporting πx, we have $\mathsf{supp}\, \pi x \subseteq \pi(\mathsf{supp}\, x)$. For the reverse inclusion, $\mathsf{supp}\, \pi^{-1}\pi x \subseteq \pi^{-1}(\mathsf{supp}\, \pi x) \Rightarrow \pi(\mathsf{supp}\, x) \subseteq \mathsf{supp}\, \pi x$.

(iii) $\pi x = x \Rightarrow \pi(\mathsf{supp}\, x) = \mathsf{supp}\, \pi x = \mathsf{supp}\, x \Rightarrow \pi \in \mathsf{fix}\, \mathsf{supp}\, x$.

(iv) If $a \in A - B$, let $b \notin A \cup B$. Then $(a\ b) \in \mathsf{Fix}\, B - \mathsf{fix}\, A$.

(v) We have $\pi \in \mathsf{fix}\, x \Rightarrow \pi x = x \Rightarrow \pi(f(x)) = f(\pi x) = f(x) \Rightarrow \pi \in \mathsf{fix}\, f(x)$, so $\mathsf{Fix}\, \mathsf{supp}\, x \subseteq \mathsf{fix}\, x \subseteq \mathsf{fix}\, f(x)$, so $\mathsf{supp}\, x$ supports $f(x)$. Since $\mathsf{supp}\, f(x)$ is the smallest set supporting $f(x)$, $\mathsf{supp}\, f(x) \subseteq \mathsf{supp}\, x$. □

The $G_{\mathbb{A}}$-sets $\mathbb{A}$ and $\wp_{\mathrm{fin}} \mathbb{A}$ are nominal sets with $\mathsf{supp}\, a = \{a\}$ and $\mathsf{supp}\, A = A$ for $a \in \mathbb{A}$ and $A \in \wp_{\mathrm{fin}} \mathbb{A}$. By Lemma 2.1(v), the only equivariant function $\mathbb{A} \to \mathbb{A}$ is the identity.

Expressions and ν-Strings NKA expressions are defined by the grammar

$$e ::= a \in \mathbb{A} \mid e + e \mid ee \mid e^* \mid 0 \mid 1 \mid \nu a.e.$$

The scope of the binding νa in $\nu a.e$ is e. As a notational convention, we assign the binding operator νa lower precedence than product but higher precedence than sum; thus in products, scopes extend as far to the right as possible. For example, $\nu a.ab\ \nu b.ba$ should be read as $\nu a.(ab\ \nu b.(ba))$ and not $(\nu a.ab)(\nu b.ba)$. The set of NKA expressions over $\mathbb{A}$ is denoted $\mathsf{Exp}\, \mathbb{A}$.

The free variables $\mathsf{FV}(e)$ of an expression e are defined as usual, and the group $G_{\mathbb{A}}$ acts on $\mathsf{Exp}\, \mathbb{A}$ by permuting the variables in the obvious way. For example, $(a\ b)\nu a.b = \nu b.a$. Formally, $e \mapsto \pi e : \mathsf{Exp}\, \mathbb{A} \to \mathsf{Exp}\, \mathbb{A}$ is the unique homomorphic extension of $\pi : \mathbb{A} \to \mathbb{A}$ with respect to the signature of KA and ν, and $\mathsf{FV} : \mathsf{Exp}\, \mathbb{A} \to \wp_{\mathrm{fin}} \mathbb{A}$ is the unique homomorphic extension of $a \mapsto \{a\}$, where the operations $+, \cdot, ^*, 0, 1$ in $\wp_{\mathrm{fin}} \mathbb{A}$ have meaning $\cup, \cup, \mathsf{id}, \varnothing, \varnothing$, respectively, and $\nu a.A = A - \{a\}$. The relation $\equiv_\alpha$ of *α-equivalence* on $\mathsf{Exp}\, \mathbb{A}$ is defined to be the least congruence containing the pairs $\{e \equiv_\alpha \pi e \mid \pi \in \mathsf{Fix}\, \mathsf{FV}(e)\}$. Let $[e]$ denote the $\equiv_\alpha$-congruence class of e.

Lemma 2.2 *The $\equiv_\alpha$-congruence classes of* $\mathsf{Exp}\,\mathbb{A}$ *form a nominal set with* $\mathsf{supp}\,[e] = \mathsf{FV}(e)$*, and the function* FV *is well defined and equivariant on $\equiv_\alpha$-classes.*

Proof. The function $\mathsf{FV} : \mathsf{Exp}\,\mathbb{A} \to \wp_{\mathrm{fin}}\,\mathbb{A}$ is equivariant, because $\mathsf{FV} \circ \pi$ and $\pi \circ \mathsf{FV}$ are homomorphisms that agree on the generating set $\mathbb{A}$: $\mathsf{FV}(\pi a) = \pi(\mathsf{FV}(a)) = \{\pi a\}$. The function FV is also well defined on $\equiv_\alpha$-classes, because if $\pi \in \mathsf{Fix}\,\mathsf{FV}(e)$, then $\mathsf{FV}(\pi e) = \pi\mathsf{FV}(e) = \mathsf{FV}(e)$, therefore $\equiv_\alpha$ refines the kernel of FV. Thus $\mathsf{FV} : \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha} \to \wp_{\mathrm{fin}}\,\mathbb{A}$ with $\mathsf{FV}([e]) = \mathsf{FV}(e)$. Finally, $\mathsf{supp}[e] \subseteq \mathsf{FV}(e)$ since $\mathsf{Fix}\,\mathsf{FV}(e) \subseteq \mathsf{fix}\,[e]$, and $\mathsf{FV}(e) \subseteq \mathsf{supp}[e]$ by Lemma 2.1(v) and the fact that $\mathsf{supp}\,\mathsf{FV}(e) = \mathsf{FV}(e)$. □

A *ν-string* is a string with νa binders; that is, it is an NKA expression with no occurrence of $+$, *, or 0 modulo multiplicative associativity, and no occurrence of 1 except to denote the null string, in which case we use ε instead.

$$x ::= a \in \Sigma \mid xx \mid \varepsilon \mid \nu a.x$$

The set of ν-strings over $\mathbb{A}$ is denoted $\mathbb{A}^\nu$.

NKA Axioms The axioms proposed by Gabbay and Ciancia [12] are:

$$\begin{array}{lll} \nu a.(d+e) = \nu a.d + \nu a.e & a\#e \Rightarrow \nu b.e = \nu a.(a\ b)e & \nu a.\nu b.e = \nu b.\nu a.e \\ a\#e \Rightarrow (\nu a.d)e = \nu a.de & a\#e \Rightarrow e(\nu a.d) = \nu a.ed & a\#e \Rightarrow \nu a.e = e. \end{array} \tag{2}$$

One can derive a normal form for ν-strings in which each binder νa binds a variable immediately to its right [15].

Nominal ν-Monoids A *nominal ν-monoid over* $\mathbb{A}$ is a structure $(M, \cdot, 1, \mathbb{A}, \nu)$ with binding operation $\nu : \mathbb{A} \times M \to M$ such that

- $(M, \cdot, 1)$ is a monoid with group action $G_\mathbb{A} \times M \to M$ such that M is a nominal set;
- the operation ν satisfies the axioms (2) (omitting the first, which is irrelevant as there is no $+$ operation);
- the monoid operations and ν are equivariant, or equivalently, every $\pi \in G_\mathbb{A}$ is an automorphism of M.

Nominal Kleene algebra (NKA) A *nominal Kleene algebra over* $\mathbb{A}$ is a structure $(K, +, \cdot, ^*, 0, 1, \mathbb{A}, \nu)$ with binding operation $\nu : \mathbb{A} \times K \to K$ such that

- $(K, +, \cdot, ^*, 0, 1)$ is a KA with group action $G_\mathbb{A} \times K \to K$ such that K is a nominal set;
- the operation ν satisfies the axioms (2);

- the KA operations and ν are equivariant in the sense that

$$\begin{array}{lll} \pi(x+y) = \pi x + \pi y & \pi(xy) = (\pi x)(\pi y) & \pi 0 = 0 \\ \pi(x^*) = (\pi x)^* & \pi(\nu a.x) = \nu(\pi a).(\pi x) & \pi 1 = 1, \end{array}$$

  or equivalently, every $\pi \in G_{\mathbb{A}}$ is an automorphism of K.

3 A Nominal Language Model

Let M be a nominal ν-monoid over $\mathbb{A}$. Metasymbols $m, n, \ldots$ denote elements of M. Let $\wp M$ denote the powerset of M. On $\wp M$, define the KA operations and group action

$$\begin{array}{c} A + B = A \cup B \quad AB = \{mn \mid m \in A,\ n \in B\} \quad A^* = \bigcup_k A^k \quad 0 = \varnothing \\ 1 = \{\varepsilon\} \quad \nu a.A = \{\nu a.m \mid m \in A\} \quad \pi A = \{\pi m \mid m \in A\}. \end{array} \tag{3}$$

We say that A is *uniformly finitely supported* if $\bigcup_{m \in A} \mathsf{supp}\, m$ is finite. Let

$$\begin{aligned} \wp_{\mathrm{fs}} M &= \{A \subseteq M \mid A \text{ is finitely supported}\} \\ \wp_{\mathrm{ufs}} M &= \{A \subseteq M \mid A \text{ is uniformly finitely supported}\}. \end{aligned}$$

Lemma 3.1 ([11, Theorem 2.29]) *For $A \subseteq M$, if A is uniformly finitely supported, then A is finitely supported and* $\mathsf{supp}\, A = \bigcup_{m \in A} \mathsf{supp}\, m$.

The converse is false in general. Both $\wp_{\mathrm{fs}} M$ and $\wp_{\mathrm{ufs}} M$ are closed under the operations (3).

Theorem 3.2 *The set $\wp_{\mathrm{ufs}} M$ with group action and KA operations* (3) *forms an NKA.*

Proof. The set $\wp_{\mathrm{ufs}} M$ with the specified group action is evidently a nominal set, and the KA axioms are satisfied because the KA operations are the standard language-theoretic ones. For the axioms of (2),

$$\begin{aligned} \nu a.(A + B) &= \{\nu a.m \mid m \in A \cup B\} \\ &= \{\nu a.m \mid m \in A\} \cup \{\nu a.m \mid m \in B\} = \nu a.A + \nu a.B \end{aligned}$$

$$\begin{aligned} \nu a.\nu b.A &= \{\nu a.m \mid m \in \{\nu b.n \mid n \in A\}\} = \{\nu a.\nu b.n \mid n \in A\} \\ &= \{\nu b.\nu a.n \mid n \in A\} = \nu b.\nu a.A \end{aligned}$$

For the remaining axioms, assume $a \# A$, that is, $a \# m$ for all $m \in A$.

$$\nu a.A = \{\nu a.m \mid m \in A\} = \{m \mid m \in A\} = A$$

$$\begin{aligned} \nu b.A &= \{\nu b.m \mid m \in A\} = \{\nu a.(a\ b)m \mid m \in A\} \\ &= \{\nu a.n \mid n \in (a\ b)A\} = \nu a.(a\ b)A \end{aligned}$$

$$\begin{aligned} (\nu a.B)A &= \{(\nu a.m)n \mid m \in B,\ n \in A\} = \{\nu a.mn \mid m \in B,\ n \in A\} \\ &= \{\nu a.m \mid m \in BA\} = \nu a.BA. \end{aligned}$$

The argument for $A(\nu a.B) = \nu a.AB$ is similar.

Finally, the KA operations are equivariant, as

$$\begin{aligned}\pi(A \cup B) &= \{\pi m \mid m \in A \cup B\} \\ &= \{\pi m \mid m \in A\} \cup \{\pi m \mid m \in B\} = \pi A \cup \pi B\end{aligned}$$

$$\begin{aligned}\pi(AB) &= \{\pi m \mid m \in AB\} = \{\pi(mn) \mid m \in A,\ n \in B\} \\ &= \{(\pi m)(\pi n) \mid m \in A,\ n \in B\} = \{\pi m \mid m \in A\}\{\pi n \mid n \in B\} \\ &= (\pi A)(\pi B)\end{aligned}$$

$$\pi(A^*) = \pi(\bigcup_n A^n) = \bigcup_n (\pi A)^n = (\pi A)^*$$

$$\pi 1 = \pi\{\varepsilon\} = \{\varepsilon\} = 1 \qquad\qquad \pi 0 = \pi\varnothing = \varnothing = 0.$$

□

3.1 Canonical Interpretation over $\mathbb{A}^\nu/{\equiv}$

For $x, y \in \mathbb{A}^\nu$, define $x \equiv y$ if x and y are provably equivalent using the axioms (2) (omitting the first, which is irrelevant as there is no occurrence of $+$ in ν-strings) and the axioms of equality and congruence. Let $[x]$ denote the $\equiv$-congruence class of x and $\mathbb{A}^\nu/{\equiv}$ the ν-monoid of all such congruence classes.

The *length* of $x \in \mathbb{A}^\nu$ is the number of occurrences of symbols of $\mathbb{A}$ in x, excluding binding occurrences νb. If $x \equiv y$, then x and y have the same length, and an occurrence of a symbol in x is free iff the corresponding occurrence in y is free. If both are free, then they are the same symbol. If both are bound, then they can be different symbols due to α-conversion. If two ν-strings are α-equivalent, then they are $\equiv$-equivalent.

Lemma 3.3 *Every $m \in \mathbb{A}^\nu/{\equiv}$ is finitely supported, and $\mathsf{supp}[x] = \mathsf{FV}(x)$.*

Proof. As observed, equivalent ν-strings have the same free variables, thus FV is well defined on $\mathbb{A}^\nu/{\equiv}$. If $\pi \in \mathsf{Fix}\,\mathsf{FV}(x)$, then $\pi x \equiv x$ by α-conversion. As this is true for any string equivalent to x, we have $\pi[x] = [x]$, therefore $\mathsf{FV}(x)$ supports $[x]$ and $\mathsf{supp}[x] \subseteq \mathsf{FV}(x)$.

For the reverse inclusion, we just note that $\mathsf{FV} : \mathbb{A}^\nu/{\equiv} \to \wp_{\mathrm{fin}}\,\mathbb{A}$ is equivariant and that $\mathsf{supp}\, A = A$ for $A \in \wp_{\mathrm{fin}}\,\mathbb{A}$, and apply Lemma 2.1(v). □

Lemma 3.4 *The structure $\mathbb{A}^\nu/{\equiv}$ is a nominal ν-monoid over $\mathbb{A}$ and satisfies the following universality property: For any other nominal ν-monoid M over $\mathbb{A}$ and any equivariant set function $h : \mathbb{A} \to M$, h extends uniquely to an equivariant homomorphism $h : \mathbb{A}^\nu/{\equiv} \to M$.*

Proof. The structure $\mathbb{A}^\nu/{\equiv}$ is evidently a nominal ν-monoid over $\mathbb{A}$. If M is any other ν-monoid over $\mathbb{A}$ and $h : \mathbb{A} \to M$, then h extends uniquely and homomorphically to $h : \mathbb{A}^\nu \to M$. It remains to show that h is equivariant and well defined on $\equiv$-classes. Equivariance follows from the fact that $h \circ \pi$ and $\pi \circ h$ are homomorphisms that agree on the generating set $\mathbb{A}$. Finally, to show that h is well defined on $\equiv$-classes, we need to show that $\equiv$ refines the kernel of h. This is true because it holds for the axioms of (2), and $\equiv$ is the least congruence containing these pairs. For example,

- $h(\nu a.\nu b.x) = \nu a.\nu b.h(x) = \nu b.\nu a.h(x) = h(\nu b.\nu a.x)$; and
- $a\#y$ implies $a\#h(y)$ by Lemma 2.1(v), thus $h(\nu a.xy) = \nu a.h(x)h(y) = (\nu a.h(x))h(y) = h(\nu a.x)h(y) = h((\nu a.x)y)$.

□

Henceforth, let $M = \mathbb{A}^\nu/{\equiv}$. The map $L : \mathsf{Exp}\,\mathbb{A} \to \wp\, M$ is defined to be the unique homomorphism such that $L(a) = \{[a]\}$ for $a \in \mathbb{A}$. Explicitly,

$$\begin{gathered} L(e_1 + e_2) = L(e_1) \cup L(e_2) \qquad L(e_1 e_2) = \{mn \mid m \in L(e_1),\ n \in L(e_2)\} \\ L(e^*) = L(e)^* = \textstyle\bigcup_k L(e)^k \qquad L(0) = \varnothing \qquad L(1) = \{\varepsilon\} \\ L(a) = \{[a]\},\ a \in \mathbb{A} \qquad L(\nu a.e) = \nu a.L(e) = \{\nu a.m \mid m \in L(e)\}. \end{gathered} \tag{4}$$

The following lemma guarantees the existence of an equivariant homomorphism $L : \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha} \to \wp_{\mathrm{ufs}}\, M$.

Lemma 3.5 *The map L is well defined and equivariant on $\equiv_\alpha$-congruence classes and takes values in $\wp_{\mathrm{ufs}}\, M$.*

Proof. The image of L is contained in $\wp_{\mathrm{ufs}}\, M$ since it holds for $L(a)$ and $\wp_{\mathrm{ufs}}\, M$ is closed under the operations (3). Equivariance follows from the fact that $L \circ \pi$ and $\pi \circ L$ are homomorphisms that agree on the generating set $\mathbb{A}$: $L(\pi a) = \pi L(a) = \{[\pi a]\}$. To show that L is well defined on $\equiv_\alpha$-classes, we first observe that

$$\mathsf{supp}\, L(e) = \bigcup_{m \in L(e)} \mathsf{supp}\, m = \bigcup_{m \in L(e)} \mathsf{FV}(m) \subseteq \mathsf{FV}(e).$$

The first two equalities are from Lemmas 3.3 and 3.1, and the final inclusion is easily shown by induction on the definition of L. Thus

$$\mathsf{Fix}\,\mathsf{FV}(e) \subseteq \mathsf{Fix}\,\mathsf{supp}\, L(e) \subseteq \mathsf{fix}\, L(e).$$

If $\pi \in \mathsf{Fix}\,\mathsf{FV}(e)$, then $L(\pi e) = \pi L(e) = L(e)$. As $\equiv_\alpha$ is the smallest congruence for which this is true, $\equiv_\alpha$ refines the kernel of L. □

The following deconstruction lemma is important for our coalgebraic treatment of §4.

Lemma 3.6

(i) *If* $ax \equiv by$*, then* $a = b$ *and* $x \equiv y$*.*

(ii) *If* $\nu a.ax \equiv \nu a.ay$*, then* $x \equiv y$*.*

Proof. This follows from the normal form of [15]. For (i), the first symbol is free and is uniquely determined, since no axiom of (2) or proof rule can alter it, thus $a = b$. For (ii), the first symbol is bound to the initial νa. It is not uniquely determined due to α-conversion. Once we have ax and ay in (i) or $\nu a.ax$ and $\nu a.ay$ in (ii), the remaining reductions of [15] apply only to x and y, thus reducing to normal form gives $x \equiv y$. □

Lemma 3.6(ii) is somewhat delicate. Note that $\nu a.x \equiv \nu a.y$ does not imply $x \equiv y$ in general: we have $\nu b.ab \not\equiv \nu b.ba$, but $\nu a.\nu b.ab \equiv \nu a.\nu b.ba$ by applying the permutation $(a\ b)$ and reversing the order of the bindings.

4 Coalgebraic Structure

We will presently define syntactic Brzozowski and Antimirov derivatives on NKA expressions over $\mathbb{A}$ and a corresponding semantic derivative on subsets of M. These constructs will be seen to comprise coalgebras for a Nom-endofunctor K defined by

$$KX = 2 \times X^{\mathbb{A}} \times [\mathbb{A}]X, \tag{5}$$

where the nominal set $X^{\mathbb{A}}$ consists of finitely supported functions $\mathbb{A} \to X$ and $[\mathbb{A}]X$ is the abstraction of the nominal set X; see [21] for a detailed account of the abstraction functor on Nom. We recall here that the nominal set $[\mathbb{A}]X$ is defined as the quotient of $\mathbb{A} \times X$ by the equivalence relation given by $(a, x) \sim (b, y)$ if and only if for any fresh c we have $(c\ a)x = (c\ b)y$. Furthermore, the abstraction functor $[\mathbb{A}](-)$ has a left adjoint $\mathbb{A}\#(-)$ defined on objects by

$$\mathbb{A}\#X = \{(a, x) \mid a\#x\}.$$

Hence a K-coalgebra is a tuple of the form $(X, \mathsf{obs}, \mathsf{cont}, \mathsf{cont}_\nu)$, where X is a nominal set and

$$\mathsf{obs} : X \to 2 \qquad \mathsf{cont} : X \to X^{\mathbb{A}} \qquad \mathsf{cont}_\nu : X \to [\mathbb{A}]X \tag{6}$$

are equivariant functions, called the *observation* and *continuation* maps, respectively. Using the cartesian closed structure on Nom and the adjunction $\mathbb{A}\#(-) \dashv [\mathbb{A}](-)$, the continuation maps are in one-to-one correspondence with maps defined on $\mathbb{A} \times X$ and $\mathbb{A}\#X$ respectively.

$$\frac{\mathsf{cont} : X \to X^{\mathbb{A}}}{\mathsf{cont}^\flat : \mathbb{A} \times X \to X} \qquad \frac{\mathsf{cont}_\nu : X \to [\mathbb{A}]X}{\mathsf{cont}_\nu^\flat : \mathbb{A}\#X \to X}$$

To simplify notation, we write

$$\mathsf{cont}_a : X \to X,\ a \in \mathbb{A} \qquad \mathsf{cont}_{\nu a} : \{s \in X \mid a\#s\} \to X,\ a \in \mathbb{A} \tag{7}$$

for the uncurried continuation maps obtained by fixing the first argument to $a \in \mathbb{A}$. Intuitively, cont_a tries to consume a free variable a and $\mathsf{cont}_{\nu a}$ tries to consume a bound variable a bound by νa. We will discuss the intuition behind these constructs more fully and justify the typing (6) in Example 4.1 below.

It follows from (1) that the equivariance of the structure map $(\mathsf{obs}, \mathsf{cont}, \mathsf{cont}_\nu)$ is equivalent to the properties

$$\mathsf{cont}_{\pi a} \circ \pi = \pi \circ \mathsf{cont}_a \qquad \mathsf{cont}_{\nu \pi a} \circ \pi = \pi \circ \mathsf{cont}_{\nu a} \qquad \mathsf{obs} \circ \pi = \mathsf{obs} \tag{8}$$

for all $\pi \in G_{\mathbb{A}}$.

Henceforth, the term *coalgebra* refers specifically to coalgebras for the Nom-functor K in (5).

4.1 Semantic Derivative

Let $M = \mathbb{A}^\nu/{\equiv}$. The semantic derivative is defined as a K-coalgebra with carrier the nominal set $\wp_{\mathrm{fs}}\, M$:

$$(\varepsilon, \delta, \delta_\nu) : \wp_{\mathrm{fs}}\, M \to 2 \times (\wp_{\mathrm{fs}}\, M)^{\mathbb{A}} \times [\mathbb{A}]\, \wp_{\mathrm{fs}}\, M$$

where

$$\varepsilon(A) = \begin{cases} 1, & \varepsilon \in A, \\ 0, & \varepsilon \notin A \end{cases} \qquad \begin{aligned} \delta_a(A) &= \{m \mid am \in A\},\ a \in \mathbb{A} \\ \delta_{\nu a}(A) &= \{m \mid \nu a.am \in A\},\ a \in \mathbb{A}. \end{aligned}$$

The maps δ_a and $\delta_{\nu a}$ are well defined by Lemma 3.6.

Example 4.1 The a in δ_a and $\delta_{\nu a}$ play very different roles. Intuitively, $\delta_a(A)$ tries to consume a free variable a at the front of strings in A. For example, for $b \neq a$,

- $\delta_a(\{aa, bb\}) = \{a\}$
- $\delta_a(\{\nu b.ab\}) = \{\nu b.b\}$
- $\delta_a(\{\nu a.ab\}) = \varnothing$ (since the first letter of $\nu a.ab$ is bound).

On the other hand, $\delta_{\nu a}(A)$ tries to consume a bound variable at the front of strings in A and change the remaining variables bound by the same binder to a. The bound variable need not be a, but it should be possible to change it to a by α-conversion. For example, for $b \neq a$,

1. $\delta_{\nu a}(\{\nu a.aa\}) = \delta_{\nu a}(\{\nu b.bb\}) = \{a\}$ (since $\nu b.bb = \nu a.aa$ in $\mathbb{A}^\nu/{\equiv}$)
2. $\delta_{\nu a}(\{\nu a.ab\}) = \{b\}$

3. $\delta_{va}(\{va.ba\}) = \varnothing$ (since the initial symbol b is not bound)
4. $\delta_{va}(\{vb.ba\}) = \varnothing$ (since $vb.ba \neq va.am$ for any $m \in \mathbb{A}^\nu/\equiv$)
5. $\delta_{va}(\{(va.aa)a\}) = \varnothing$ (since $(va.aa)a \neq va.am$ for any $m \in \mathbb{A}^\nu/\equiv$)
6. $\delta_{va}(\{(vb.bb)b\}) = \{ab\}$ (since $(vb.bb)b = va.aab$ in $\mathbb{A}^\nu/\equiv$).

Examples 4 and 5 do not arise in our coalgebraic semantics, since δ_{va} may only be applied to A for which a is fresh due to the domain restriction in (7). If there are free occurrences of a, one cannot α-convert to obtain a string of the form $va.am$, since those free occurrences would be captured. □

Lemma 4.2

(i) $\delta_a(AB) = \delta_a(A)B \cup \varepsilon(A)\delta_a(B)$

(ii) $\delta_a(A^*) = \delta_a(A)A^*$

(iii) $\delta_a(vb.A) = \begin{cases} \varnothing, & b = a, \\ vb.\delta_a(A), & b \neq a. \end{cases}$

Proof.

$$\begin{aligned}
\text{(i) } \delta_a(AB) &= \{m \mid am \in AB\} \\
&= \{mn \mid am \in A,\ n \in B\} \cup \{m \mid \varepsilon \in A,\ am \in B\} \\
&= \{mn \mid m \in \{m \mid am \in A\},\ n \in B\} \cup \varepsilon(A)\{m \mid am \in B\} \\
&= \delta_a(A)B \cup \varepsilon(A)\delta_a(B). \\
\text{(ii) } \delta_a(A^*) &= \{m \mid am \in A^*\} = \{m \mid am \in AA^*\} \\
&= \{mn \mid am \in A,\ n \in A^*\} = \{mn \mid m \in \{m \mid am \in A\},\ n \in A^*\} \\
&= \delta_a(A)A^*.
\end{aligned}$$

(iii) We have $\delta_a(va.A) = \{m \mid am \in va.A\} = \varnothing$, and for $b \neq a$,

$$\begin{aligned}
\delta_a(vb.A) &= \{m \mid am \in vb.A\} = \{m \mid am \in \{vb.an \mid an \in A\}\} \\
&= \{m \mid am \in \{a(vb.n) \mid an \in A\}\} = \{vb.n \mid an \in A\} \\
&= vb.\{n \mid an \in A\} = vb.\delta_a(A).
\end{aligned}$$

□

Lemma 4.3

(i) *If $a\#AB$, then $\delta_{va}(AB) = \delta_{va}(A)B \cup \varepsilon(A)\delta_{va}(B)$.*

(ii) *If $a\#A^*$, then $\delta_{va}(A^*) = \delta_{va}(A)A^*$.*

(iii) *If $a\#vb.A$, then $\delta_{va}(vb.A) = vb.\delta_{va}(A) \cup \delta_a((a\ b)A)$.*

Proof. (i) If $a\#AB$, then $a\#B$, and

$$\begin{aligned}\delta_{\nu a}(AB) &= \{m \mid \nu a.am \in AB\} \\ &= \{mn \mid \nu a.am \in A,\ n \in B,\ a\#n\} \cup \{n \mid \varepsilon \in A,\ \nu a.an \in B\} \\ &= \{mn \mid m \in \{m \mid \nu a.am \in A\},\ n \in B\} \cup \varepsilon(A)\{n \mid \nu a.an \in B\} \\ &= \delta_{\nu a}(A)B \cup \varepsilon(A)\delta_{\nu a}(B).\end{aligned}$$

The condition $a\#B$ is necessary: for $A = \{\nu a.aa\}$ and $B = \{a\}$, we have $AB = \{(\nu a.aa)a\}$, thus $aa \in \delta_{\nu a}(A)B$ but $aa \notin \delta_{\nu a}(AB)$.

(ii) If $a\#A^*$, then $a\#A$, and

$$\begin{aligned}\delta_{\nu a}(A^*) &= \{m \mid \nu a.am \in A^*\} = \{m \mid \nu a.am \in AA^*\} \\ &= \{mn \mid \nu a.am \in A,\ n \in A^*,\ a\#n\} \\ &= \{mn \mid m \in \{m \mid \nu a.am \in A\},\ n \in A^*\} = \delta_{\nu a}(A)A^*.\end{aligned}$$

Again, the condition $a\#A$ is necessary: for $A = \{\nu a.aa, a\}$, we have $(\nu a.aa)a \in AA^*$, thus $aa \in \delta_{\nu a}(A)A^*$ but $aa \notin \delta_{\nu a}(AA^*)$.

(iii) Assume $a\#\nu b.A$. By α-conversion, we can assume without loss of generality that $b \neq a$.

First we show the left-to-right inclusion. We have

$$\delta_{\nu a}(\nu b.A) = \{m \mid \nu a.am \in \nu b.A\} = \{m \mid \exists n \in A\ \nu a.am = \nu b.n\}.$$

If $\nu a.am = \nu b.n$, the first letter of $\nu b.n$ is bound, and it is either bound to the initial νb or to something else. In the former case, we have $n = bk$ for some k and $a\#\nu b.n$, so

$$\nu a.am = \nu b.n = \nu b.bk = \nu a.a(a\ b)k,$$

therefore $m = (a\ b)k$ by Lemma 3.6, and

$$am = a(a\ b)k = (a\ b)n \in (a\ b)A.$$

In the latter case, $n = \nu c.c\ell \in A$ for some $c\ell$ and $a\#A$. Let $k = (a\ c)\ell$. Then

$$\nu a.ak = \nu a.a(a\ c)\ell = (a\ c)\nu c.c\ell = (a\ c)n = n \in A$$

and $\nu a.am = \nu b.n = \nu b.\nu a.ak = \nu a.a(\nu b.k)$, thus $m = \nu b.k$ by Lemma 3.6. Putting these together,

$$\begin{aligned}\delta_{\nu a}(\nu b.A) &= \{m \mid \nu a.am \in \nu b.A\} = \{m \mid \exists n \in A\ \nu a.am = \nu b.n\} \\ &\subseteq \{\nu b.k \mid \nu a.ak \in A\} \cup \{m \mid am \in (a\ b)A\} \\ &= \{\nu b.k \mid k \in \delta_{\nu a}(A)\} \cup \{m \mid am \in (a\ b)A\} \\ &= \nu b.\delta_{\nu a}(A) \cup \delta_a((a\ b)A).\end{aligned}$$

For the right-to-left inclusion, if $a\#\nu b.A$ and $b \neq a$, we have

$$\begin{aligned}am \in (a\ b)A \Rightarrow (a\ b)am \in A \Rightarrow \nu a.am = \nu b.(a\ b)am \in \nu b.A \\ \nu a.ak \in A \Rightarrow \nu a.a(\nu b.k) = \nu b.\nu a.ak \in \nu b.A\end{aligned}$$

and hence

$$\begin{aligned}\nu b.\delta_{\nu a}(A) \cup \delta_a((a\ b)A) &= \{\nu b.k \mid k \in \delta_{\nu a}(A)\} \cup \{m \mid am \in (a\ b)A\} \\ &= \{\nu b.k \mid \nu a.ak \in A\} \cup \{m \mid am \in (a\ b)A\} \\ &\subseteq \{m \mid \nu a.am \in \nu b.A\} \cup \{m \mid \nu a.am \in \nu b.A\} \\ &= \delta_{\nu a}(\nu b.A).\end{aligned}$$

□

4.2 Brzozowski Derivative

The syntactic Brzozowski derivative is defined inductively on the set of α-equivalence classes of NKA expressions $\mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha}$. Like the semantic derivative, it can also be defined on a broader domain, but also will only make coalgebraic sense for the domain (6).

$$(\mathsf{E}, \mathsf{D}, \mathsf{D}_\nu) : \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha} \to 2 \times (\mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha})^{\mathbb{A}} \times [\mathbb{A}](\mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha})$$

The continuation maps D and D_ν can be further broken down as

$$\mathsf{D}_a : \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha} \to \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha} \quad \mathsf{D}_{\nu a} : \{e \in \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha} \mid a\#e\} \to \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha}$$

for $a \in \mathbb{A}$. We first define these maps on $\mathsf{Exp}\,\mathbb{A}$, then argue that they are well defined on $\equiv_\alpha$-classes.

$$\begin{aligned}&\mathsf{E}(e_1 + e_2) = \mathsf{E}(e_1) + \mathsf{E}(e_2) && \mathsf{E}(e_1 e_2) = \mathsf{E}(e_1)\mathsf{E}(e_2) && \mathsf{E}(a) = \mathsf{E}(0) = 0 \\ &\mathsf{E}(1) = \mathsf{E}(e^*) = 1 && \mathsf{E}(\nu a.e) = \mathsf{E}(e)\end{aligned}$$

$$\begin{aligned}&\mathsf{D}_a(e_1 + e_2) = \mathsf{D}_a(e_1) + \mathsf{D}_a(e_2) && \mathsf{D}_a(e_1 e_2) = \mathsf{D}_a(e_1)e_2 + \mathsf{E}(e_1)\mathsf{D}_a(e_2) \\ &\mathsf{D}_a(e^*) = \mathsf{D}_a(e)e^* && \mathsf{D}_a(0) = \mathsf{D}_a(1) = 0 \\ &\mathsf{D}_a(b) = \begin{cases}1, & b = a \\ 0, & b \neq a\end{cases} && \mathsf{D}_a(\nu b.e) = \begin{cases}0, & b = a \\ \nu b.\mathsf{D}_a(e), & b \neq a\end{cases}\end{aligned}$$

$$\begin{aligned}&\mathsf{D}_{\nu a}(e_1 + e_2) = \mathsf{D}_{\nu a}(e_1) + \mathsf{D}_{\nu a}(e_2) && \mathsf{D}_{\nu a}(e_1 e_2) = \mathsf{D}_{\nu a}(e_1)e_2 + \mathsf{E}(e_1)\mathsf{D}_{\nu a}(e_2) \\ &\mathsf{D}_{\nu a}(e^*) = \mathsf{D}_{\nu a}(e)e^* && \mathsf{D}_{\nu a}(\nu b.e) = \nu b.\mathsf{D}_{\nu a}(e) + \mathsf{D}_a((a\ b)e),\ b \neq a \\ &\mathsf{D}_{\nu a}(0) = \mathsf{D}_{\nu a}(1) = \mathsf{D}_{\nu a}(b) = 0\end{aligned}$$

We can also define $\mathsf{D}_{\nu a}(\nu a.e) = \mathsf{D}_{\nu a}(\nu b.(a\ b)e)$ for an arbitrary b such that $b\#e$ and $b \neq a$, although strictly speaking this is not a function, since the choice of b is not determined. However, the choice of b does not matter, as we are considering expressions modulo α-equivalence. This will be treated formally in Lemma 4.6.

Example 4.4 For $b \neq a$,

1. $\mathsf{D}_{\nu a}(\nu b.bb) = \nu b.\mathsf{D}_{\nu a}(bb) + \mathsf{D}_a((a\ b)bb) = 0 + a = a$.

2. $\mathsf{D}_{\nu a}(\nu a.aa) = \mathsf{D}_{\nu a}(\nu b.bb) = a$.

3. $\mathsf{D}_{\nu a}(\nu a.ab) = \mathsf{D}_{\nu a}(\nu c.cb) = \nu c.\mathsf{D}_{\nu a}(cb) + \mathsf{D}_a(ab) = 0 + b = b$.

4. $\mathsf{D}_{\nu a}(\nu b.ba) = \nu b.\mathsf{D}_{\nu a}(ba) + \mathsf{D}_a((a\ b)ba) = 0 + b = b$.

Example 4 may seem incorrect, since the argument has a free variable a and the result has a free variable b, but this situation will not arise in our coalgebraic semantics, since $\mathsf{D}_{\nu a}$ will only be applied to e for which a is fresh. □

Lemma 4.5 *The derivatives are equivariant.*

Proof. We show that the derivatives satisfy the properties (8). For the semantic derivative $(\wp M, \varepsilon, \delta)$,

$$\varepsilon(A) = \begin{cases} 1, & \varepsilon \in A \\ 0, & \varepsilon \notin A \end{cases} = \begin{cases} 1, & \varepsilon \in \pi A \\ 0, & \varepsilon \notin \pi A \end{cases} = \varepsilon(\pi A)$$

$$\begin{aligned} \pi(\delta_a(A)) &= \pi(\{m \mid am \in A\}) = \{\pi m \mid \pi(am) \in \pi A\} \\ &= \{\pi m \mid (\pi a)(\pi m) \in \pi A\} = \{n \mid (\pi a)n \in \pi A\} = \delta_{\pi a}(\pi A) \end{aligned}$$

$$\begin{aligned} \pi(\delta_{\nu a}(A)) &= \pi(\{m \mid \nu a.am \in A\}) = \{\pi m \mid \pi(\nu a.am) \in \pi A\} \\ &= \{\pi m \mid \nu(\pi a).(\pi a)(\pi m) \in \pi A\} = \{n \mid \nu(\pi a).(\pi a)n) \in \pi A\} \\ &= \delta_{\nu \pi a}(\pi A). \end{aligned}$$

Intuitively, the syntactic derivative in Brzozowski form $(\mathsf{Exp}\,\mathbb{A}, E, D)$ satisfies (8) as well, since the inductive definitions commute with permutations. The only slightly subtle case is $\mathsf{D}_{\nu a}(\nu b.e)$ for $b \neq a$, which we argue explicitly:

$$\begin{aligned} \pi(\mathsf{D}_{\nu a}(\nu b.e)) &= \pi(\nu b.\mathsf{D}_{\nu a}(e) + \mathsf{D}_a((a\ b)e)) \\ &= \nu(\pi b).\pi(\mathsf{D}_{\nu a}(e)) + \pi(\mathsf{D}_a((a\ b)e)) \\ &= \nu(\pi b).\mathsf{D}_{\nu \pi a}(\pi e) + \mathsf{D}_{\pi a}(\pi(a\ b)\pi^{-1}\pi e) \\ &= \nu(\pi b).\mathsf{D}_{\nu \pi a}(\pi e) + \mathsf{D}_{\pi a}((\pi a\ \pi b)\pi e) \\ &= \mathsf{D}_{\nu \pi a}(\nu(\pi b).\pi e) = \mathsf{D}_{\nu \pi a}(\pi(\nu b.e)). \end{aligned}$$

Thus the semantic and syntactic derivatives are coalgebras over $G_{\mathbb{A}}$-Set of type (6). The syntactic derivative is nominal, as expressions have finite support. □

Lemma 4.6 *The syntactic derivative is well defined modulo* $\equiv_\alpha$.

Proof. This is an inductive argument. The case $\mathsf{D}_{\nu a}(\nu b.e)$ for $b \neq a$ is the only interesting case. Suppose we have an α-conversion $\nu b.e \equiv_\alpha \nu c.(b\ c)e$ with $c\#e$ and $c \neq a$. Since $\mathsf{FV}(\mathsf{D}_{\nu a}(\nu b.e)) \subseteq \{a\} \cup \mathsf{FV}(\nu b.e)$, by Lemma 4.5,

$$\mathsf{D}_{\nu a}(\nu b.e) \equiv_\alpha (b\ c)\mathsf{D}_{\nu a}(\nu b.e) = \mathsf{D}_{\nu(b\ c)a}((b\ c)\nu b.e) = \mathsf{D}_{\nu a}(\nu c.(b\ c)e).$$

□

Theorem 4.7 *For all $e \in \mathsf{Exp}\,\mathbb{A}$ and $a \in \mathbb{A}$,*

(i) $\varepsilon(L(e)) = \mathsf{E}(e)$

(ii) $\delta_a(L(e)) = L(\mathsf{D}_a(e))$

(iii) *If $a\#e$, then $\delta_{\nu a}(L(e)) = L(\mathsf{D}_{\nu a}(e))$.*

Proof. (i) The maps $\varepsilon \circ L$ and E are homomorphisms that agree on the generators, therefore agree everywhere. Intuitively, $\mathsf{E}(e)$ is the value obtained by substituting 0 for all letters $a \in \mathbb{A}$ occurring in e and simplifying.

(ii) For $e \in \{b, 1, 0\}$ with $b \neq a$, we have $\mathsf{D}_a(e) = 0$ and $L(e) \in \{\{b\}, \{\varepsilon\}, \varnothing\}$. In none of the three cases does $L(e)$ contain an element of the form ax for $a \neq b$, thus $\delta_a(L(e)) = \varnothing$. In all three cases,

$$L(\mathsf{D}_a(e)) = L(0) = \varnothing = \delta_a(L(e)).$$

The remaining base case is $\mathsf{D}_a(a)$. Here we have

$$L(\mathsf{D}_a(a)) = L(1) = \{\varepsilon\} = \{x \mid ax \in \{a\}\} = \delta_a(\{a\}) = \delta_a(L(a)).$$

For sums,

$$\begin{aligned} L(\mathsf{D}_a(e_1 + e_2)) &= L(\mathsf{D}_a(e_1)) \cup L(\mathsf{D}_a(e_2)) \\ &= \delta_a(L(e_1)) \cup \delta_a(L(e_2)) = \delta_a(L(e_1 + e_2)). \end{aligned}$$

For products, using (i) and Lemma 4.2(i),

$$\begin{aligned} L(\mathsf{D}_a(e_1e_2)) &= L(\mathsf{D}_a(e_1)e_2 + \mathsf{E}(e_1)\mathsf{D}_a(e_2)) = L(\mathsf{D}_a(e_1))L(e_2) \cup \mathsf{E}(e_1)L(\mathsf{D}_a(e_2)) \\ &= \delta_a(L(e_1))L(e_2) \cup \varepsilon(L(e_1))\delta_a(L(e_2)) = \delta_a(L(e_1)L(e_2)) \\ &= \delta_a(L(e_1e_2)). \end{aligned}$$

For star, using Lemma 4.2(ii),

$$\begin{aligned} L(\mathsf{D}_a(e^*)) &= L(\mathsf{D}_a(e)e^*) = L(\mathsf{D}_a(e))L(e^*) \\ &= \delta_a(L(e))L(e)^* = \delta_a(L(e)^*) = \delta_a(L(e^*)). \end{aligned}$$

For ν, using Lemma 4.2(iii),

$$\begin{aligned} &L(\mathsf{D}_a(\nu a.e)) = L(0) = \varnothing = \delta_a(\nu a.L(e)) = \delta_a(L(\nu a.e)) \\ &L(\mathsf{D}_a(\nu b.e)) = \nu b.L(\mathsf{D}_a(e)) = \nu b.\delta_a(L(e)) = \delta_a(\nu b.L(e)) = \delta_a(L(\nu b.e)). \end{aligned}$$

(iii) The argument for 0,1, and + is the same as in (ii). For b,

$$L(\mathsf{D}_{\nu a}(b)) = L(0) = \varnothing = \delta_{\nu a}(\{b\}) = \delta_{\nu a}(L(b)).$$

In the remaining cases, we use the fact that $a\#e$ implies $a\#L(e)$ (Lemmas 3.5 and 2.1(v)), so that Lemma 4.3 applies. For products, using (i) and Lemma 4.3(i),

$$\begin{aligned} L(\mathsf{D}_{va}(e_1e_2)) &= L(\mathsf{D}_{va}(e_1)e_2 + \mathsf{E}(e_1)\mathsf{D}_{va}(e_2)) \\ &= L(\mathsf{D}_{va}(e_1))L(e_2) \cup \mathsf{E}(e_1)L(\mathsf{D}_{va}(e_2)) \\ &= \delta_{va}(L(e_1))L(e_2) \cup \varepsilon(L(e_1))\delta_{va}L(e_2) \\ &= \delta_{va}(L(e_1)L(e_2)) = \delta_{va}(L(e_1e_2)). \end{aligned}$$

For star, using Lemma 4.3(ii),

$$\begin{aligned} L(\mathsf{D}_{va}(e^*)) &= L(\mathsf{D}_{va}(e)e^*) = L(\mathsf{D}_{va}(e))L(e^*) = \delta_{va}(L(e))L(e^*) \\ &= \delta_{va}(L(e))L(e)^* = \delta_{va}(L(e)^*) = \delta_{va}(L(e^*)). \end{aligned}$$

For ν, using Lemmas 4.3(iii) and 3.5,

$$\begin{aligned} L(\mathsf{D}_{va}(\nu b.e)) &= L(\nu b.\mathsf{D}_{va}(e) + \mathsf{D}_a((a\ b)e)) = \nu b.L(\mathsf{D}_{va}(e)) \cup L(\mathsf{D}_a((a\ b)e)) \\ &= \nu b.\delta_{va}(L(e)) \cup \delta_a(L((a\ b)e)) = \nu b.\delta_{va}(L(e)) \cup \delta_a((a\ b)L(e)) \\ &= \delta_{va}(\nu b.L(e)) = \delta_{va}(L(\nu b.e)). \end{aligned}$$

□

4.3 Antimirov Derivative

There is an analog of the Antimirov derivative for NKA of type

$$\mathcal{A} : \mathsf{Exp}\,\mathbb{A} \to (\wp\,\mathsf{Exp}\,\mathbb{A})^{\mathbb{A}+\mathbb{A}}$$

that corresponds to nondeterministic automata. It is defined inductively as follows: for $a, b \in \mathbb{A}$ and $e, e_1, e_2 \in \mathsf{Exp}\,\mathbb{A}$,

$$\begin{gathered} \mathcal{A}_a(e_1 + e_2) = \mathcal{A}_a(e_1) \cup \mathcal{A}_a(e_2) \\ \mathcal{A}_a(e_1e_2) = \mathcal{A}_a(e_1)\{e_2\} \cup \mathsf{E}(e_1)\mathcal{A}_a(e_2) \\ \mathcal{A}_a(e^*) = \mathcal{A}_a(e)\{e^*\} \qquad \mathcal{A}_a(0) = \mathcal{A}_a(1) = \varnothing \\ \mathcal{A}_a(b) = \begin{cases} \{1\}, & b = a \\ \varnothing, & b \neq a \end{cases} \qquad \mathcal{A}_a(\nu b.e) = \begin{cases} \varnothing, & b = a \\ \nu b.\mathcal{A}_a(e), & b \neq a \end{cases} \end{gathered}$$

$$\begin{gathered} \mathcal{A}_{va}(e_1 + e_2) = \mathcal{A}_{va}(e_1) \cup \mathcal{A}_{va}(e_2) \\ \mathcal{A}_{va}(e_1e_2) = \mathcal{A}_{va}(e_1)\{e_2\} \cup \mathsf{E}(e_1)\mathcal{A}_{va}(e_2) \\ \mathcal{A}_{va}(e^*) = \mathcal{A}_{va}(e)\{e^*\} \qquad \mathcal{A}_{va}(0) = \mathcal{A}_{va}(1) = \mathcal{A}_{va}(b) = \varnothing \\ \mathcal{A}_{va}(\nu b.e) = \nu b.\mathcal{A}_{va}(e) \cup \mathcal{A}_a((a\ b)e),\ b \neq a. \end{gathered}$$

Lemma 4.8 $\mathsf{D}_{va}(e) = \sum \mathcal{A}_{va}(e)$.

4.4 Final Coalgebra

The nominal coalgebra $(\wp_{\mathrm{fs}} M, \varepsilon, \delta, \delta_\nu)$ is final among coalgebras for the Nom-endofunctor K defined in (5). These are the coalgebras $(X, \mathsf{obs}, \mathsf{cont}, \mathsf{cont}_\nu)$ for which X is a nominal set and obs, cont and cont_ν are equivariant. Such a coalgebra can be viewed as an automaton with states X, transitions cont and cont_ν, and acceptance condition obs. The inputs to the automaton are elements of M. Starting from a state $s \in X$, an element $m \in M$ is *accepted* if $\mathsf{Accept}(s, m)$, where

$$\mathsf{Accept}(s, \varepsilon) = \mathsf{obs}(s) \tag{9}$$

$$\mathsf{Accept}(s, am) = \mathsf{Accept}(\mathsf{cont}_a(s), m) \tag{10}$$

$$\mathsf{Accept}(s, \nu a.am) = \mathsf{Accept}(\mathsf{cont}_{\nu a}(s), m),\ a\#s. \tag{11}$$

Clause (11) requires some explanation. We must choose a representative element $\nu a.am$ of the $\equiv$-class such that a is fresh for s, so that $\mathsf{cont}_{\nu a}(s)$ will be defined. It is always possible to find such an a, since the $\equiv$-class is closed under α-conversion and s has finite support. However, the result is independent of the choice of a, as shown in part (ii) of the next lemma, so $\mathsf{Accept}(s, \nu a.am)$ is well defined.

Lemma 4.9

(i) *The acceptance function is equivariant:*

$$\mathsf{Accept}(\pi s, \pi m) = \pi(\mathsf{Accept}(s, m)) = \mathsf{Accept}(s, m).$$

(ii) *If b#s and c#s, then*

$$\mathsf{Accept}(s, \nu b.bm) = \mathsf{Accept}(s, \nu c.c(b\ c)m).$$

We do not explicitly require $c\#\nu b.bx$ in (ii); however, this is a consequence of (i) and Lemma 2.1(v).

Proof. We prove (i) and (ii) by mutual induction on the length of the input string. For the basis (i),

$$\mathsf{Accept}(\pi s, \pi\varepsilon) = \mathsf{obs}(\pi s) = \mathsf{obs}(s) = \mathsf{Accept}(s, \varepsilon).$$

Case (i) for strings of the form bx or $\nu b.bx$ depends on the induction hypothesis (i) for strings x and case (ii) for $\nu b.bx$. Case (ii) for strings $\nu b.bx$ depends on case (i) for strings x.

$$\begin{aligned}
&\mathsf{Accept}(\pi s, \pi(bx)) \\
&\quad = \mathsf{Accept}(\pi s, (\pi b)(\pi x)) = \mathsf{Accept}(\mathsf{cont}_{\pi b}(\pi s), \pi x) \\
&\quad = \mathsf{Accept}(\pi(\mathsf{cont}_b(s)), \pi x) = \mathsf{Accept}(\mathsf{cont}_b(s), x) = \mathsf{Accept}(s, bx).
\end{aligned}$$

The induction hypothesis was applied in the next-to-last step.

$$\begin{aligned}
\mathsf{Accept}(\pi s, \pi(\nu b.bx)) &= \mathsf{Accept}(\pi s, \nu(\pi b).(\pi b)(\pi x)) \\
&= \mathsf{Accept}(\mathsf{cont}_{\nu\pi b}(\pi s), \pi x),\ \pi b \# \pi s && (12) \\
&= \mathsf{Accept}(\pi(\mathsf{cont}_{\nu b}(s)), \pi x),\ \pi b \# \pi s \\
&= \mathsf{Accept}(\mathsf{cont}_{\nu b}(s), x),\ b \# s && (13) \\
&= \mathsf{Accept}(s, \nu b.bx). && (14)
\end{aligned}$$

We used (ii) in steps (12) and (14) for well-definedness, along with the fact that freshness is equivariant, so that $b\#s$ iff $\pi b \# \pi s$. The induction hypothesis (i) was applied in step (13) for the shorter string x.

For case (ii), if $b\#s$ and $c\#s$, apply (i) with $\pi = (b\ c)$ on the shorter string x:

$$\begin{aligned}
\mathsf{Accept}(s, \nu b.bx) &= \mathsf{Accept}(\mathsf{cont}_{\nu b}(s), x) = \mathsf{Accept}((b\ c)\mathsf{cont}_{\nu b}(s), (b\ c)x) \\
&= \mathsf{Accept}(\mathsf{cont}_{\nu(b\ c)b}((b\ c)s), (b\ c)x) \\
&= \mathsf{Accept}(\mathsf{cont}_{\nu c}(s), (b\ c)x) = \mathsf{Accept}(s, \nu c.c(b\ c)x).
\end{aligned}$$

□

The unique coalgebra homomorphism from $(X, \mathsf{obs}, \mathsf{cont}, \mathsf{cont}_\nu)$ to the final coalgebra is just the automata-theoretic language semantics:

Theorem 4.10 (Final coalgebra) *The coalgebra $(\wp_{\mathrm{fs}} M, \varepsilon, \delta, \delta_\nu)$ is a final K-coalgebra. The unique coalgebra homomorphism $(X, \mathsf{obs}, \mathsf{cont}, \mathsf{cont}_\nu)$ to the final coalgebra is given by*

$$L_X : (X, \mathsf{obs}, \mathsf{cont}, \mathsf{cont}_\nu) \to (\wp_{\mathrm{fs}} M, \varepsilon, \delta, \delta_\nu) \quad L_X(s) = \{m \mid \mathsf{Accept}(s, m)\}.$$

Moreover, the coalgebra homomorphism $L_{\mathsf{Exp}\,\mathbb{A}} : \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha} \to \wp_{\mathrm{fs}} M$ coincides with the algebra homomorphism $L : \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha} \to \wp_{\mathrm{fs}} M$ defined in (4)*.*

Proof. We have to check that for all $e \in \mathsf{Exp}\,\mathbb{A}$ and $a \in \mathbb{A}$,

(i) $\varepsilon(L_X(s)) = \mathsf{obs}(s)$

(ii) $\delta_a(L_X(s)) = L_X(\mathsf{cont}_a(s))$

(iii) $\delta_{\nu a}(L_X(s)) = L_X(\mathsf{cont}_{\nu a}(s)),\ a\#s$

and L_X is the unique map for which (i)–(iii) hold. The clauses (i)–(iii) are just the acceptance conditions (9)–(11), respectively:

$$\varepsilon(L_X(s)) = \varepsilon(\{m \mid \mathsf{Accept}(s, m)\}) = \mathsf{Accept}(s, \varepsilon) = \mathsf{obs}(s)$$

$$\begin{aligned}
\delta_a(L_X(s)) &= \delta_a(\{m \mid \mathsf{Accept}(s, m)\}) = \{x \mid ax \in \{m \mid \mathsf{Accept}(s, m)\}\} \\
&= \{x \mid \mathsf{Accept}(s, ax)\} = \{x \mid \mathsf{Accept}(\mathsf{cont}_a(s), x)\} = L_X(\mathsf{cont}_a(s)).
\end{aligned}$$

For (iii), assume $a\#s$.

$$\begin{aligned}\delta_{\nu a}(L_X(s)) &= \delta_{\nu a}(\{m \mid \mathsf{Accept}(s,m)\}) = \{x \mid \nu a.ax \in \{m \mid \mathsf{Accept}(s,m)\}\} \\ &= \{x \mid \mathsf{Accept}(s, \nu a.ax)\} = \{x \mid \mathsf{Accept}(\mathsf{cont}_{\nu a}(s), x)\} \\ &= L_X(\mathsf{cont}_{\nu a}(s)).\end{aligned}$$

One can show by induction on the length of ν-strings that $x \in F(s)$ iff $x \in L_X(s)$ for any other map F satisfying (i)–(iii), therefore L_X is uniquely determined by these properties.

Theorem 4.7 says that the algebra homomorphism $L : \mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha} \to \wp_{\mathsf{fs}}\, M$ defined in (4) satisfies properties (i)–(iii) for $X = \mathsf{Exp}\,\mathbb{A}$, therefore coincides with $L_{\mathsf{Exp}\,\mathbb{A}}$. □

A more standard construction of the final coalgebra computed via the final sequence of the functor K [1] yields an equivalent presentation based on normal forms of ν-strings up to α-equivalence. However, this characterization is more cumbersome algebraically, as it requires explicit α-conversion to define sequential composition.

4.5 Automata Representation: Half of a Kleene Theorem

In this section we prove a theorem for NKA that relates the algebraic and coalgebraic structure. As noted in §4.4, a coalgebra can be regarded as an automaton acceptor with states X, transitions cont, and acceptance condition obs. The inputs to the automaton are elements of M. The state sets are nominal sets and may be formally infinite, but still may be essentially finite in a sense to be described next.

Following [3], we define the *size* of a coalgebra $(X, \mathsf{obs}, \mathsf{cont})$ to be the number of orbits of X under $G_{\mathbb{A}}$, where the *orbit* of $s \in X$ is the set $\{\pi s \mid \pi \in G_{\mathbb{A}}\}$. The orbit of s is the singleton $\{s\}$ if $\mathsf{supp}\, s = \varnothing$, otherwise it is infinite. The orbits partition X and determine an equivalence relation. The coalgebra is called *orbit-finite* if the total number of orbits is finite.

Lemma 4.11 *Let $(X, \mathsf{obs}, \mathsf{cont})$ be a coalgebra, $s \in X$, and $a \in \mathbb{A}$.*

(i) $\mathsf{supp}\,(\mathsf{cont}_{\nu a}(s)) \subseteq \{a\} \cup \mathsf{supp}\, s$.

(ii) *If $a \in \mathsf{supp}\, s$, then $\mathsf{supp}\,(\mathsf{cont}_a(s)) \subseteq \mathsf{supp}\, s$.*

(iii) *If $L(s)$ is uniformly finitely supported and $m \in L(s)$, then $\mathsf{supp}\, m \subseteq \mathsf{supp}\, s$.*

(iv) *If $a\#s$ and $L(s)$ is uniformly finitely supported, then $\mathsf{cont}_a(s)$ is a dead state (one for which $L(s) = \varnothing$).*

Proof. (i) Let $\pi \in \mathsf{Fix}(\{a\} \cup \mathsf{supp}\, s)$. By (8),

$$\pi(\mathsf{cont}_{\nu a}(s)) = \mathsf{cont}_{\nu \pi a}(\pi s) = \mathsf{cont}_{\nu a}(s),$$

thus $\pi \in \mathsf{fix}\,\mathsf{cont}_{\nu a}(s)$. Since $\pi \in \mathsf{Fix}(\{a\} \cup \mathsf{supp}\, s)$ was arbitrary, $\{a\} \cup \mathsf{supp}\, s$ supports $\mathsf{cont}_{\nu a}(s)$. Since $\mathsf{supp}\,(\mathsf{cont}_{\nu a}(s))$ is the least set supporting $\mathsf{cont}_{\nu a}(s)$, $\mathsf{supp}\,(\mathsf{cont}_{\nu a}(s)) \subseteq \{a\} \cup \mathsf{supp}\, s$.

(ii) We have $\mathsf{supp}\,(\mathsf{cont}_a(s)) \subseteq \{a\} \cup \mathsf{supp}\, s \subseteq \mathsf{supp}\, s$, the first inclusion by the same argument as (i) and the second by the assumption $a \in \mathsf{supp}\, s$.

(iii) By Lemma 3.1, $\mathsf{supp}\, m \subseteq \mathsf{supp}\, L(s)$, and by Lemmas 2.1(v) and 4.9, $\mathsf{supp}\, L(s) \subseteq \mathsf{supp}\, s$.

(iv) If $m \in L(\mathsf{cont}_a(s))$, then $am \in L(s)$, contradicting (iii). □

Theorem 4.12 (Half Kleene) *For every NKA expression e, there is a coalgebra X with designated start state s such that $L_X(s) = L(e)$. The coalgebra has an orbit-finite nondeterministic representation given by the Antimirov representation of the Brzozowski derivatives of e.*

Proof. The desired coalgebra is the subcoalgebra of $(\mathsf{Exp}\,\mathbb{A}/{\equiv_\alpha}, E, D)$ generated by e. The designated start state is e. That this is correct is immediate from Theorem 4.10. Orbit-finiteness of the Antimirov representation will follow from the data representation to be developed in §5.1. □

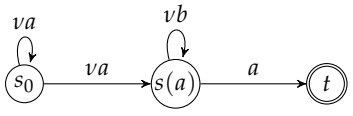
It is interesting that the Antimirov derivative gives an orbit-finite representation, whereas the Brzozowski derivative in general does not:

Example 4.13 Here is an example showing that the Antimirov derivatives of an expression give an orbit-finite nondeterministic automaton, whereas the deterministic automaton given by the Brzozowski derivatives is not necessarily orbit-finite. Consider the expression

$$e = (\nu a.a)^*(\nu a.a(\nu b.b)^* a).$$

This yields a nondeterministic automaton with

- states s_0, $s(a)$ for $a \in \mathbb{A}$, t, and r corresponding to the subexpressions e, $(\nu b.b)^* a$, 1, and 0, respectively;
- group action $\pi s_0 = s_0$, $\pi s(a) = s(\pi a)$, $\pi t = t$, and $\pi r = r$;
- supports $\mathsf{supp}\, s_0 = \mathsf{supp}\, t = \mathsf{supp}\, r = \varnothing$ and $\mathsf{supp}\, s(a) = \{a\}$;
- observations $\mathsf{obs}(s_0) = \mathsf{obs}(s(a)) = \mathsf{obs}(r) = 0$ and $\mathsf{obs}(t) = 1$;
- nondeterministic transitions $\mathsf{cont}_{\nu a}(s_0) = \{s_0, s(a)\}$ for $a \in \mathbb{A}$, $\mathsf{cont}_{\nu b}(s(a)) = s(a)$ for $b \# a$, $\mathsf{cont}_a(s(a)) = t$, and all other transitions going to a dead state r (not shown).



This automaton is orbit-finite with four orbits, but the standard subset construction would yield a non-orbit-finite deterministic automaton with a transition sequence

$$\{s_0\} \xrightarrow{\nu a} \{s_0, s(a)\} \xrightarrow{\nu b} \{s_0, s(a), s(b)\} \xrightarrow{\nu c} \{s_0, s(a), s(b), s(c)\} \xrightarrow{\nu d} \cdots$$

corresponding to the Brzozowski derivative sequence

$$\begin{aligned} e &\xrightarrow{\nu a} e + (\nu b.b)^* a \\ &\xrightarrow{\nu b} e + (\nu b.b)^* b + (\nu b.b)^* a \\ &\xrightarrow{\nu c} e + (\nu b.b)^* c + (\nu b.b)^* b + (\nu b.b)^* a \xrightarrow{\nu d} \cdots \end{aligned}$$

The successive elements of these sequences have unboundedly large supports, thus represent infinitely many orbits. □

5 A Decision Procedure

5.1 Data Representation

We represent the Antimirov derivative in terms of *spines*. This representation was used previously in [7, 14] and is also related to constructions of [2, 25]. Here we adapt it to the nominal setting by including the binding operator ν.

Roughly speaking, the spine related to an occurrence of a letter a in e is obtained by collecting all the expressions appearing in e to the right of that occurrence of a and all binders in whose scope that occurrence of a occurs. Intuitively, the expressions appearing in e to the left of that occurrence of a must be consumed before that occurrence of a can be consumed itself. The spine represents the residual term that remains after that occurrence of a is consumed.

The spines are defined inductively as follows:

$$\begin{gathered} R(e_1 + e_2) = R(e_1) \cup R(e_2) \qquad R(e_1 e_2) = R(e_1)\{e_2\} \cup R(e_2) \\ R(e^*) = R(e)\{e^*\} \qquad R(1) = R(0) = \varnothing \qquad R(b) = \{b\} \\ R(\nu b.e) = \nu b.R(e). \end{gathered}$$

The function R' is defined in the same way except for the base case $R'(b) = \{1\}$:

$$\begin{gathered} R'(e_1 + e_2) = R'(e_1) \cup R'(e_2) \qquad R'(e_1 e_2) = R'(e_1)\{e_2\} \cup R'(e_2) \\ R'(e^*) = R'(e)\{e^*\} \qquad R'(1) = R'(0) = \varnothing \qquad R'(b) = \{1\} \\ R'(\nu b.e) = \nu b.R'(e). \end{gathered}$$

Lemma 5.1 *The cardinalities of $R(e)$ and $R'(e)$ are at most the number of occurrences of letters $b \in \mathbb{A}$ in e.*

Proof. Easy induction on the definition. □

Every element of $R(e)$ is of the form

$$\nu a_1.(\nu a_2.(\cdots(\nu a_{n-1}.(\nu a_n.ae_n)e_{n-1})\cdots)e_2)e_1 \tag{15}$$

and every element of $R'(e)$ is of the form

$$\nu a_1.(\nu a_2.(\cdots(\nu a_{n-1}.(\nu a_n.1e_n)e_{n-1})\cdots)e_2)e_1 \tag{16}$$

where $n \geq 0$ and e_i is either null or a subexpression of e. The occurrence of a shown in (15) is the leftmost occurrence of a letter in the expression. This letter is either bound by one of the νa_i or free. By α-renaming, we can assume without loss of generality that the bound variables in each element of $R(e)$ and $R'(e)$ are distinct and different from the free variables.

We now define two families of related sets $S(e)$ and $S'(e)$. These sets contain all expressions obtained from some element of $R(e)$ (respectively, $R'(e)$) by deleting zero or more binders νb after α-conversion. Formally,

$$S(e_1 + e_2) = S(e_1) \cup S(e_2) \qquad S(e_1e_2) = S(e_1)\{e_2\} \cup S(e_2)$$
$$S(e^*) = S(e)\{e^*\} \qquad S(1) = S(0) = \varnothing \qquad S(b) = \{b\}$$
$$S(\nu b.e) = \nu b.S(e) \cup \textstyle\bigcup_a (a\ b)S(e).$$

$$S'(e_1 + e_2) = S'(e_1) \cup S'(e_2) \qquad S'(e_1e_2) = S'(e_1)\{e_2\} \cup S'(e_2)$$
$$S'(e^*) = S'(e)\{e^*\} \qquad S'(1) = S'(0) = \varnothing \qquad S'(b) = \{1\}$$
$$S'(\nu b.e) = \nu b.S'(e) \cup \textstyle\bigcup_a (a\ b)S'(e).$$

Thus every element of $S(e)$ is of the form

$$\nu \overline{a}_1.(\nu \overline{a}_2.(\cdots(\nu \overline{a}_{n-1}.(\nu \overline{a}_n.ae_n)e_{n-1})\cdots)e_2)e_1 \tag{17}$$

and every element of $S'(e)$ is of the form

$$\nu \overline{a}_1.(\nu \overline{a}_2.(\cdots(\nu \overline{a}_{n-1}.(\nu \overline{a}_n.1e_n)e_{n-1})\cdots)e_2)e_1 \tag{18}$$

where $\nu \overline{a}_i$ is either the binder νa_i or nothing, and e_i is either null or a subexpression of e with some variables renamed. The occurrence of a shown in (17) is the leftmost occurrence of a letter in the expression and would be the next letter consumed by the derivative. This letter is either bound by one of the $\nu \overline{a}_i$ or free. If free, the derivative $\mathcal{A}_a$ would simply delete that letter, giving (18). If bound, the derivative $\mathcal{A}_{\nu c}$ would first α-convert the expression by applying $(a\ c)$, then delete the first letter c and its binder νc.

Unlike $R(e)$ and $R'(e)$, the sets $S(e)$ and $S'(e)$ are infinite. However, they are orbit-finite, as each element of $S(e)$ and $S'(e)$ is obtained from an element of $R(e)$ or $R'(e)$ by deleting some some subset of binders after α-conversion. There are at most exponentially many subsets of binders that could be removed, and every orbit contains a representative from the resulting set of expressions. Thus the number of orbits is at most exponential in the size of e.

Lemma 5.2 *The maps S, S', R, and R' are equivariant.*

Proof. Induction on the definition. The only slightly nontrivial case is $\nu b.e$.

$$\begin{aligned} S(\pi(\nu b.e)) &= S(\nu(\pi b).(\pi e)) \\ &= \nu(\pi b).S(\pi e) \cup \textstyle\bigcup_a (\pi a\ \pi b) S(\pi e) \\ &= \nu(\pi b).S(\pi e) \cup \textstyle\bigcup_a \pi(a\ b)\pi^{-1}\pi(S(e)) \\ &= \pi(\nu b.S(e)) \cup \textstyle\bigcup_a \pi((a\ b)S(e)) \\ &= \pi(S(\nu b.e)). \end{aligned}$$

□

Lemma 5.3 *All derivatives of e are contained in $S'(e)$, and $S'(e)$ is closed under taking derivatives; that is,*

(i) $\mathcal{A}_a(e) \subseteq S'(e)$;

(ii) $\mathcal{A}_{\nu a}(e) \subseteq S'(e)$;

(iii) *if $r \in S'(e)$, then $\mathcal{A}_a(r) \subseteq S'(e)$; and*

(iv) *if $r \in S'(e)$, then $\mathcal{A}_{\nu a}(r) \subseteq S'(e)$.*

Proof. (i) The proof is by induction on the structure of e.

$$\begin{gathered} \mathcal{A}_a(e_0 + e_1) = \mathcal{A}_a(e_0) \cup \mathcal{A}_a(e_1) \subseteq S'(e_0) \cup S'(e_1) = S'(e_0 + e_1) \\ \mathcal{A}_a(e_0 e_1) = \mathcal{A}_a(e_0)\{e_1\} \cup \mathcal{A}_a(e_1) \subseteq S'(e_0)\{e_1\} \cup S'(e_1) = S'(e_0 e_1) \\ \mathcal{A}_a(e^*) = \mathcal{A}_a(e)\{e^*\} \subseteq S'(e)\{e^*\} = S'(e^*) \\ \mathcal{A}_a(0) = \mathcal{A}_a(1) = \varnothing = S'(0) = S'(1) \\ \mathcal{A}_a(b) \subseteq \{1\} = S'(b) \\ \mathcal{A}_a(\nu b.e) = \nu b.\mathcal{A}_a(e) \subseteq \nu b.S'(e) \subseteq S'(\nu b.e). \end{gathered}$$

(ii) All cases are the same as in (i) except for b and $\nu b.e$. For b,

$$\mathcal{A}_{\nu a}(b) = \varnothing \subseteq \{1\} = S'(b).$$

For $\nu b.e$ with $b \neq a$,

$$\begin{aligned} \mathcal{A}_{\nu a}(\nu b.e) &= \nu b.\mathcal{A}_{\nu a}(e) \cup \mathcal{A}_a((a\ b)e) \\ &\subseteq \nu b.S'(e) \cup S'((a\ b)e) \\ &= \nu b.S'(e) \cup (a\ b)S'(e) \\ &\subseteq S'(\nu b.e). \end{aligned}$$

(iii) If $r \in S'(e_0 + e_1) = S'(e_0) \cup S'(e_1)$, then either $r \in S'(e_0)$ or $r \in S'(e_1)$. In either case, $\mathcal{A}_a(r) \subseteq S'(e_0) \cup S'(e_1) = S'(e_0 + e_1)$.

If $r \in S'(e_0e_1) = S'(e_0)\{e_1\} \cup S'(e_1)$, then either $r \in S'(e_0)\{e_1\}$ or $r \in S'(e_1)$. In the first case,

$$\begin{aligned} r = se_1 \wedge s \in S'(e_0) &\Rightarrow \mathcal{A}_a(r) = \mathcal{A}_a(s)\{e_1\} \cup \mathcal{A}_a(e_1) \wedge \mathcal{A}_a(s) \subseteq S'(e_0) \\ &\Rightarrow \mathcal{A}_a(r) = \mathcal{A}_a(s)\{e_1\} \cup \mathcal{A}_a(e_1) \wedge \mathcal{A}_a(s)\{e_1\} \subseteq S'(e_0)\{e_1\} \\ &\Rightarrow \mathcal{A}_a(r) = \mathcal{A}_a(s)\{e_1\} \cup \mathcal{A}_a(e_1) \subseteq S'(e_0)\{e_1\} \cup S'(e_1) = S'(e_0e_1). \end{aligned}$$

In the second case, $\mathcal{A}_a(r) \subseteq S'(e_1) \subseteq S'(e_0e_1)$.

If $r \in S'(e^*) = S'(e)S'(e^*)$, then

$$\begin{aligned} r = se^* \wedge s \in S'(e) &\Rightarrow \mathcal{A}_a(r) = \mathcal{A}_a(s)\{e^*\} \wedge \mathcal{A}_a(s) \subseteq S'(e) \\ &\Rightarrow \mathcal{A}_a(r) = \mathcal{A}_a(s)\{e^*\} \subseteq S'(e)\{e^*\} = S'(e^*). \end{aligned}$$

The cases 0 and 1 cannot occur. For b,

$$r \in S'(b) \Rightarrow r = 1 \Rightarrow \mathcal{A}_a(r) = \varnothing \subseteq \{1\} = S'(b).$$

For $\nu b.e$, suppose $r \in S'(\nu b.e) = \nu b.S'(e) \cup \bigcup_c (c\ b)S'(e)$, with $a \neq b$. Either $r \in \nu b.S'(e)$ or $r \in (c\ b)S'(e)$. In the first case,

$$\begin{aligned} r = \nu b.s \wedge s \in S'(e) &\Rightarrow \mathcal{A}_a(r) = \mathcal{A}_a(\nu b.s) = \nu b.\mathcal{A}_a(s) \wedge \mathcal{A}_a(s) \subseteq S'(e) \\ &\Rightarrow \mathcal{A}_a(r) = \nu b.\mathcal{A}_a(s) \subseteq \nu b.S'(e) = S'(\nu b.e). \end{aligned}$$

In the second case, if $a = c$,

$$\begin{aligned} r = (a\ b)s \wedge s \in S'(e) &\Rightarrow r = (a\ b)s \wedge \mathcal{A}_b(s) \subseteq S'(e) \\ &\Rightarrow \mathcal{A}_a(r) = (a\ b)\mathcal{A}_b(s) \subseteq (a\ b)S'(e) \subseteq S'(\nu b.e) \end{aligned}$$

and if $a \neq c$,

$$\begin{aligned} r = (c\ b)s \wedge s \in S'(e) &\Rightarrow r = (c\ b)s \wedge \mathcal{A}_a(s) \subseteq S'(e) \\ &\Rightarrow \mathcal{A}_a(r) = (c\ b)\mathcal{A}_a(s) \subseteq (c\ b)S'(e) \subseteq S'(\nu b.e). \end{aligned}$$

(iv) All cases are the same as in (iii) except for the case $\nu b.e$. For this case, suppose $r \in S'(\nu b.e) = \nu b.S'(e) \cup \bigcup_c (c\ b)S'(e)$, where $a \neq b$. Either $r \in \nu b.S'(e)$ or $r \in (c\ b)S'(e)$. In the first case,

$$\begin{aligned} r = \nu b.s \wedge s \in S'(e) &\Rightarrow \mathcal{A}_{\nu a}(r) = \mathcal{A}_{\nu a}(\nu b.s) = \nu b.\mathcal{A}_{\nu a}(s) \cup \mathcal{A}_a((a\ b)s) \\ &\qquad \wedge\ \mathcal{A}_{\nu a}(s) \subseteq S'(e) \wedge \mathcal{A}_b(s) \subseteq S'(e) \\ &\Rightarrow \mathcal{A}_{\nu a}(r) = \nu b.\mathcal{A}_{\nu a}(s) \cup \mathcal{A}_a((a\ b)s) \\ &\qquad \wedge\ \nu b.\mathcal{A}_{\nu a}(s) \subseteq \nu b.S'(e) \wedge \mathcal{A}_a((a\ b)s) \subseteq (a\ b)S'(e) \\ &\Rightarrow \mathcal{A}_{\nu a}(r) \subseteq \nu b.S'(e) \cup (a\ b)S'(e) \subseteq S'(\nu b.e). \end{aligned}$$

In the second case, we have $r = (c\ b)s$ and $s \in S'(e)$. If $c \neq a$,

$$\begin{aligned} &r = (c\ b)s \wedge s \in S'(e) \\ &\quad \Rightarrow r = (c\ b)s \wedge \mathcal{A}_{\nu a}(s) \subseteq S'(e) \\ &\quad \Rightarrow \mathcal{A}_{\nu a}(r) = \mathcal{A}_{\nu a}((c\ b)s) = (c\ b)\mathcal{A}_{\nu a}(s) \subseteq (c\ b)S'(e) \subseteq S'(\nu b.e). \end{aligned}$$

If $c = a$ and $\mathcal{A}_{va}(r)$ is defined at all, then $a\#r$ and $r = (a\ b)s$, so $b\#s$.

$$\begin{aligned} r = (a\ b)s \wedge s \in S'(e) \\ \Rightarrow r = (a\ b)s \wedge \mathcal{A}_{vb}(s) \subseteq S'(e) \\ \Rightarrow \mathcal{A}_{va}(r) = \mathcal{A}_{va}((a\ b)s) = (a\ b)\mathcal{A}_{vb}(s) \subseteq (a\ b)S'(e) \subseteq S'(vb.e). \end{aligned}$$

□

5.2 A Decision Procedure

Theorem 5.4 *Equivalence of NKA expressions is decidable in deterministic exponential space.*

Proof. Given an expression e, the Antimirov derivative determines a nominal nondeterministic coalgebra whose states are e and the spines $S'(e)$. This is an infinite set of states, but each element of $S'(e)$ is obtained from an element of $S(e)$ by deleting the first occurrence of a letter, and each element of $S(e)$ is obtained from an element of $R(e)$ by renaming and deleting some binders, thus the set is orbit-finite.

As observed in §4.4, we can regard the Antimirov derivative structure as represented by sets of spines. The Brzozowski derivative is a sum of elements of the Antimirov derivative, each of which is a spine. Two expressions are equivalent if the corresponding nondeterministic automata accept the same language.

We now describe a nondeterministic procedure that looks for a violation of bisimilarity between the two coalgebras corresponding to e_1 and e_2. The procedure guesses an input $x \in \mathbb{A}^\nu$ and verifies that it is accepted by e_1 and not by e_2 (or vice versa, but assume the former without loss of generality). At any point in time, it has a current spine of e_1 (an element of $S'(e_1)$) and a set of spines of e_2 (a subset of $S'(e_2)$). The current spine s of e_1 represents the current state that the nondeterministic automaton corresponding to e_1 would be in after scanning an initial portion of the guessed input string x, and the set of current spines r of e_2 represent all possible states that the nondeterministic automaton corresponding to e_2 could be in after scanning that same initial portion of x. For each of the spines r of e_2, the procedure maintains a partial matching between the variables of s and those of r. The matching is determined by the free variables of the original expressions, which must be paired in s and r, and the free variables that have appeared by the elimination of a bound variable in a cont_{va} step, which are both a and which also must be paired. The nondeterministic procedure accepts if ever the current state s of e_1 is an accepting state, that is, if $\mathsf{obs}(s) = 1$, and all current states r of e_2 are rejecting states, that is, $\mathsf{obs}(r) = 0$.

The exponential space bound on this procedure comes from the observation that each spine can be represented in linear space, and if two current states r_1, r_2 of e_2 have the same partial matching with the current state s of e_1 but are otherwise the same up to renaming of free variables not paired with any variable

in s, only one of r_1, r_2 need be kept. The procedure will never take a transition cont_a on one of those variables, because otherwise the machine corresponding to e_1 would reject by Lemma 4.11(iv). Thus there are only exponentially many states of e_2 that can correspond to the current state of e_1 at any time. The actual free variables themselves are not important, but only the partial matchings between the current state s of e_1 and the current states r of e_2. By renaming when necessary, a pool of linearly many variables suffices.

The nondeterministic algorithm can be converted to a deterministic algorithm using Savitch's theorem. □

The naive bound of exponential space may seem like a gross overestimate, especially in light of Lemma 5.1. However, the following example shows that it may be difficult to do better. Consider the behavior of the nondeterministic decision procedure on two copies of the expression

$$\nu a_1 \ldots \nu a_n.(a_1 + \cdots + a_n)^n a_1 a_2 \cdots a_n.$$

Say the procedure guesses the prefix $(\nu a.a)^n$ to try to separate the two copies of the automaton. After scanning $(\nu a.a)^n$, the first automaton will be in some state $b_1 b_2 \cdots b_n$, having applied $\mathcal{A}_{\nu b_i}$ for $1 \leq i \leq n$. But the second automaton can be in any one of $n!$ inequivalent states $b_{\sigma(1)} \cdots b_{\sigma(n)}$, one for each permutation σ of $\{1, \ldots, n\}$, corresponding to the order in which the binders in the second copy of the expression were eliminated. All these states of the second automaton must be represented in the state of the nondeterministic procedure.

6 Conclusion and Open Problems

In this paper we have explored the coalgebraic theory of nominal Kleene algebra. We have introduced a new family of semantic models consisting of sets of nominal monoids and extended the coalgebraic structure of Kleene algebra to the nominal setting using these models. We have developed nominal versions of the Brzozowski and Antimirov derivatives that accommodate bound variables and are invariant with respect to α-conversion. We have proved a theorem relating the algebraic and coalgebraic structure, namely that every expression gives rise to an equivalent automaton. We have used this relationship to show that the equational theory can be decided in exponential space and described an efficient data representation that is amenable to implementation.

This work raises several intriguing questions. Foremost among them is the complexity of the equational theory. We have given a worst-case exponential-space decision procedure. On the other hand, the best lower bound we have is PSPACE-hardness, which follows from the PSPACE-completeness of the equivalence problem for regular expressions [26]. We have not succeeeded in proving any tighter bounds, and we have no conjecture regarding the true complexity of the problem.

Despite the high complexity of the worst-case upper bound, much like the bisimulation-based algorithms for other KA-based systems [4, 5, 7, 22, 23], the

situation may not be so bad in practice. To actually attain the worst-case bound would seem to require highly pathological examples that would be unlikely to arise in practice. However, only implementation and experimentation can confirm or refute this view. This would be an interesting direction for future work.

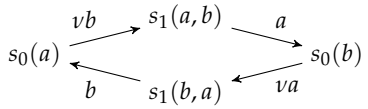
Theorem 4.12 gives one direction of a Kleene theorem: expressions to automata. The converse is false, as the following example shows. Consider the nominal coalgebra with states and group action

- $s_0(a)$ for all $a \in \mathbb{A}$ with $\pi(s_0(a)) = s_0(\pi a)$,
- $s_1(a,b)$ for all $a, b \in \mathbb{A}$, $a \neq b$ with $\pi(s_1(a,b)) = s_1(\pi a, \pi b)$, and
- s_2 with $\pi s_2 = s_2$.

The transitions and observations are

$$\begin{aligned} &\mathsf{cont}_{\nu b}(s_0(a)) = s_1(a,b) && \mathsf{obs}(s_0(a)) = 1 \\ &\mathsf{cont}_a(s_1(a,b)) = s_0(b) && \mathsf{obs}(s_1(a,b)) = \mathsf{obs}(s_2) = 0 \end{aligned}$$

for all $a, b \in \mathbb{A}$. All other transitions go to the dead state s_2.



The coalgebra is orbit-finite with three orbits $\{s_0(a) \mid a \in \mathbb{A}\}$, $\{s_1(a,b) \mid a, b \in \mathbb{A},\ a \neq b\}$, and $\{s_2\}$. The supports are $\mathsf{supp}\, s_0(a) = \{a\}$, $\mathsf{supp}\, s_1(a,b) = \{a,b\}$, and $\mathsf{supp}\, s_2 = \varnothing$. The set of ν-strings accepted from state $s_0(a)$ is

$$\{\varepsilon,\ \nu b.ba,\ \nu b.ba(\nu a.ab),\ \nu b.ba(\nu a.ab(\nu b.ba)),\ \nu b.ba(\nu a.ab(\nu b.ba(\nu a.ab))), \ldots\}$$

It can be shown using the normal form theorem of [15] that this set is not represented by any NKA expression, because it requires unbounded ν-depth.

Given that orbit-finite nominal automata are strictly more expressive than NKA expressions, two questions arise:

1. Can we characterize the subclass of orbit-finite nominal automata that are equivalent to NKA expressions? We conjecture that they are exactly those automata accepting sets of ν-strings of bounded ν-depth, although we are not sure how to characterize this class formally in a way that would lead to a converse of Theorem 4.12.

2. Can we extend the syntax of expressions to capture sets of unbounded ν-depth? The answer is yes: It is not difficult to show that orbit-finite nominal automata are equivalent to orbit-finite systems of right-linear equations. For example, the system corresponding to the automaton above would be

$$X_a = \varepsilon + \nu b.bY_{ab} \qquad\qquad Y_{ab} = aX_b.$$

The set accepted by the automaton is the least solution of the system. This gives a full Kleene theorem, but of course we are now left with the open question of deriving proof rules for this new calculus and extending the completeness result of [15].

3. Can we prove a Kleene theorem for the nominal DFA and NFA models of Bojanczyk, Klin and Lasota [3]?

4. Can we use the coalgebraic setting to systematically develop a nominal Chomsky hierarchy and (semi-)decision procedures for different classes of languages?

The first two questions have an interesting interpretation in terms of the intended application of NKA, which was originally proposed in [12] as a framework for reasoning about *dynamic* allocation of resources. However, the ν-operators in NKA expressions are statically scoped, so *static* may be the more accurate adjective. The more expressive automata of [3, 8, 17, 19] and of this paper may be the more appropriate vehicle for the study of dynamic allocation.

Acnowledgments

Thanks to Filippo Bonchi, Jamie Gabbay, Helle Hvid Hansen, Bart Jacobs, Tadeusz Litak, Damien Pous, and Ana Sokolova for many stimulating discussions, comments, and suggestions. This research was performed at Radboud University Nijmegen and supported by the Dutch Research Foundation (NWO), project numbers 639.021.334 and 612.001.113, and by the National Security Agency.

References

[1] Jiří Adámek. On final coalgebras of continuous functors. *Theor. Comput. Sci.*, 294(12):3–29, February 2003.

[2] C. Allauzen and M. Mohri. A unified construction of the Glushkov, follow, and Antimirov automata. MFCS 2006, LNCS 4162, 110–121.

[3] M. Bojanczyk, B. Klin, and S. Lasota. Automata theory in nominal sets. LMCS 10(3), 2014.

[4] F. Bonchi and D. Pous. Checking NFA equivalence with bisimulations up to congruence. POPL 2013, 457–468.

[5] T. Braibant and D. Pous. Deciding Kleene algebras in Coq. LMCS 8(1:16):1–42, 2012.

[6] G. L. Ferrari, U. Montanari, E. Tuosto, B. Victor, and K. Yemane. Modelling fusion calculus using HD-automata. CALCO 2005, LNCS 3629, 142–156.

[7] N. Foster, D. Kozen, M. Milano, A. Silva, and L. Thompson. A coalgebraic decision procedure for NetKAT. POPL 2015, 343–355.

[8] N. Francez and M. Kaminski. Finite-memory automata. TCS 134(2):329–363, 1994.

[9] N. Francez and M. Kaminski. An algebraic characterization of deterministic regular languages over infinite alphabets. TCS 306(1–3):155–175, 2003.

[10] M. Gabbay and A. M. Pitts. A new approach to abstract syntax involving binders. LICS 1999, 214–224.

[11] M. Gabbay. Foundations of nominal techniques: logic and semantics of variables in abstract syntax. *Bull. Symbolic Logic*, 17(2):161–229, 2011.

[12] M. Gabbay and V. Ciancia. Freshness and name-restriction in sets of traces with names. FoSSaCS 2011, LNCS 6604, 365–380.

[13] W. Gelade and F. Neven. Succinctness of the Complement and Intersection of Regular Expressions. TACS 2008, Dagstuhl LIPIcs 1, 325–336.

[14] D. Kozen. On the coalgebraic theory of Kleene algebra with tests. Tech. Rep. `http://hdl.handle.net/1813/10173`, Cornell, March 2008.

[15] D. Kozen, K. Mamouras, and A. Silva. Completeness and incompleteness in nominal Kleene algebra. Tech. Rep. `http://hdl.handle.net/1813/38143`, Cornell, November 2014.

[16] A. Kurz, T. Suzuki, and E. Tuosto. A characterisation of languages on infinite alphabets with nominal regular expressions. IFIP TCS 2012, LNCS 7604, 193–208.

[17] A. Kurz, T. Suzuki, and E. Tuosto. On nominal regular languages with binders. FoSSaCS 2012, LNCS 7213, 255–269.

[18] U. Montanari and M. Pistore. History dependent automata. Tech. Rep. TR-11-98, Computer Science, Università di Pisa, 1998.

[19] U. Montanari and M. Pistore. History-dependent automata: An introduction. SFM 2005, LNCS 3465, 1–28.

[20] M. Pistore. *History Dependent Automata*. PhD thesis, Università di Pisa, 1999.

[21] A. M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*, *Cambridge Tracts in Theoretical Computer Science* 57, Cambridge University Press, 2013.

[22] D. Pous. Relational algebra and KAT in Coq, February 2013. Available at `http://perso.ens-lyon.fr/damien.pous/ra`.

[23] D. Pous. Symbolic algorithms for language equivalence and Kleene algebra with tests. POPL 2015, 357–368.

[24] A. Silva. *Kleene Coalgebra*. PhD thesis, Radboud University Nijmegen, 2010.

[25] A. Silva. Position automata for Kleene algebra with tests. *Scientific Annals of Computer Science*, 22(2):367–394, 2012.

[26] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. STOC 1973, 1–9.

A Isomorphism of AL and L

In this section we show that the interpretation L defined in §3.1 is isomorphic to the alternative language interpretation AL presented in [15], which is a minor variant of the language interpretation of [12]. It was shown in [15] that the

axiomatization of nominal Kleene algebra presented in [12] is sound and complete for AL; thus $\vdash e_1 = e_2$ iff $AL(e_1) = AL(e_2)$. The interpretation L gives an alternative characterization of this model.

The interpretation AL was defined as follows. Let $\mathcal{B} = \{p, q, \ldots\}$ be a countably infinite set of atoms disjoint from $\mathbb{A} = \{a, b, \ldots\}$, and let $G_{\mathcal{B}}$ be the group of finite permutations of $\mathcal{B}$. Metasymbols $u, v, w, \ldots$ represent strings in $(\mathbb{A} \cup \mathcal{B})^*$ and $\mathsf{FV}(u)$ denotes the set of variables occurring in u (all of which occur freely, as there are no ν-binders in $(\mathbb{A} \cup \mathcal{B})^*$).

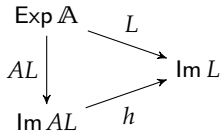
Let $\mathcal{L}$ denote the set of equivariant subsets of $(\mathbb{A} \cup \mathcal{B})^*$ with respect to $G_{\mathcal{B}}$; that is, sets $A \subseteq (\mathbb{A} \cup \mathcal{B})^*$ such that $\pi A = A$ for all $\pi \in G_{\mathcal{B}}$. The operations of NKA are defined on $\mathcal{L}$ as follows:

$$AB = \{uv \mid u \in A,\ v \in B,\ \mathsf{FV}(u) \cap \mathsf{FV}(v) \cap \mathcal{B} = \varnothing\}$$
$$A + B = A \cup B \qquad A^* = \bigcup_k A^k \qquad 0 = \varnothing \qquad 1 = \{\varepsilon\}$$
$$\nu a.A = \{(a\ p)u \mid u \in A,\ p \in \mathcal{B} - \mathsf{FV}(u)\},\ a \in \mathbb{A}.$$

As shown in [15], the set $\mathcal{L}$ is closed under these operations.

We can interpret NKA expressions over $\mathbb{A}$ as elements of $\mathcal{L}$. The interpretation map $AL : \mathsf{Exp}\,\mathbb{A} \to \mathcal{L}$ is the unique homomorphism with respect to the above language operations such that $AL(a) = \{a\}$ for $a \in \mathbb{A}$. Note that in this context, atoms $p \in \mathcal{B}$ do not appear in expressions, although they do appear in their images under AL. For example, $AL(\nu a.a) = \mathcal{B}$ and $AL(\nu a.\nu c.abc) = \{pbq \mid p, q \in \mathcal{B},\ p \neq q\}$.

Let $\mathsf{Im}\,AL \subseteq \mathcal{L}$ and $\mathsf{Im}\,L \subseteq \wp_{\mathrm{ufs}}\, M$ denote the images of $\mathsf{Exp}\,\mathbb{A}$ under AL and L, repectively. We wish to show that $\mathsf{Im}\,AL$ and $\mathsf{Im}\,L$ are isomorphic. The completeness result of [15] says that $L : \mathsf{Exp}\,\mathbb{A} \to \mathsf{Im}\,L$ factors through $\mathsf{Im}\,AL$ via AL and an epimorphism $h : \mathsf{Im}\,AL \to \mathsf{Im}\,L$:



We need only show that h is injective; that is, $AL(e)$ is uniquely determined by $L(e)$. We show this in the next lemma.

Lemma A.1 $AL(e) = \bigcup_{[x] \in L(e)} AL(x)$.

Proof. The proof is by induction on e. The cases of $+$, *, 0, 1, and $a \in \mathbb{A}$ are straightforward. For multiplication,

$$\begin{aligned} AL(e_1 e_2) &= \{uv \mid u \in AL(e_1),\ v \in AL(e_2),\ \mathsf{FV}(u) \cap \mathsf{FV}(v) \cap \mathcal{B} = \varnothing\} \\ &= \bigcup_{[x] \in L(e_1)} \bigcup_{[y] \in L(e_2)} \{uv \mid u \in AL(x),\ v \in AL(y),\ \mathsf{FV}(u) \cap \mathsf{FV}(v) \cap \mathcal{B} = \varnothing\} \\ &= \bigcup_{[x] \in L(e_1)} \bigcup_{[y] \in L(e_2)} AL(xy) = \bigcup_{[z] \in L(e_1 e_2)} AL(z). \end{aligned}$$

For ν,

$$\begin{aligned} AL(\nu a.e) &= \{(a\ p)u \mid u \in AL(e),\ p \in \mathcal{B} - \mathsf{FV}(u)\} \\ &= \bigcup_{[x] \in L(e)} \{(a\ p)u \mid u \in AL(x),\ p \in \mathcal{B} - \mathsf{FV}(u)\} \\ &= \bigcup_{[x] \in L(e)} AL(\nu a.x) = \bigcup_{[y] \in L(\nu a.e)} AL(y). \end{aligned}$$

□

We have shown

Theorem A.2 *The structures* $\mathsf{Im}\, AL$ *and* $\mathsf{Im}\, L$ *are isomorphic.*

It follows from the completeness result of [15] that $\vdash e_1 = e_2$ iff $L(e_1) = L(e_2)$.