Nominal Kleene Coalgebra

Dexter Kozen[1] Konstantinos Mamouras[1] Daniela Petrisan[2] Alexandra Silva[2]

[1]Cornell University [2]Radboud University Nijmegen

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 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] 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.

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 §C we describe a data representation for the efficient calculation of the Antimirov derivative and give an exponential-space decision procedure. Unfortunately, we are forced to omit this section from this abstract for lack of space.

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 [18–20] 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.

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.

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. 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]. 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 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.

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 a severely 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 [23] for Kleene (co)algebra, and to [12, 15] for NKA.

Kleene Algebra (KA) is the algebra of regular expressions. A *Kleene algebra* $(K, +, \cdot, ^*, 0, 1)$ is an idempotent semiring with $*$ such that x^*y is the $\leq$-least z such that $y + xz \leq z$ and yx^* is the $\leq$-least z such that $y + zx \leq z$. Explicitly,

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

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

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

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

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

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.

Expressions and ν-Strings NKA expressions are given 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, $(a\ b)\nu a.b = \nu b.a$. 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 2.1. *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.*

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:

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

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);
- 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 KA 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

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

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 3.2 ([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 3.1. *The set $\wp_{\mathrm{ufs}} M$ with group action and KA operations* (3) *forms an NKA.*

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.

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

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

Lemma 3.4.
(i) *If* $ax \equiv by$*, then* $a = b$ *and* $x \equiv y$*.*
(ii) *If* $\nu a.ax \equiv \nu a.ay$*, then* $x \equiv y$*.*

Lemma 3.4(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 $\wp_{\text{fs}}\, M$:

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

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 3.4.

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

Examples 4 and 5 do not arise in our coalgebraic semantics, since $\delta_{\nu a}$ 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 $\nu a.am$, since those free occurrences would be captured.

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

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

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.

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

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

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

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 B.13.

Example 4.2. 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 will not arise in our coalgebraic semantics, since $\mathsf{D}_{\nu a}$ will only be applied to e for which a is fresh and the argument has a free variable a.

4.3 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 4.5.
(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 B.7(v).

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 4.2 (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).

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.

4.4 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 §4.3, 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 4.6. *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 $L(s) = \varnothing$).*

Theorem 4.3 (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 4.2. Orbit-finiteness of the Antimirov representation will follow from the data representation to be developed in §C.2. □

It is interesting that the Antimirov derivatives give an orbit-finite representation, whereas the Brzozowski derivatives do not. A counterexample is given in Example C.3. The orbit-finite representation is underlying the decision procedure of the equational theory, which we omit here for lack of space. Please see Appendix C for a full account.

5 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.

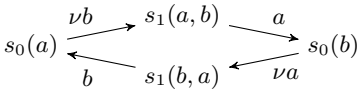
This work raises several intriguing questions. Foremost among them is the complexity of the equational theory. We have given a worst-case exponential-space decision procedure. On the other hand, the best lower bound we have is PSPACE-hardness, which follows from the PSPACE-completeness of the equivalence problem for regular expressions [25].

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], 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 4.3 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



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

for all $a, b \in \mathbb{A}$. All other transitions go to the dead state s_2. 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 4.3.
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], exposing the crucial difference that nondeterminism introduces in the nominal setting (nominal NFA are strictly more expressive that DFA)?
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.

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. Symbolic algorithms for language equivalence and Kleene algebra with tests. POPL 2015, 357–368.
23. A. Silva. *Kleene Coalgebra*. PhD thesis, Radboud University Nijmegen, 2010.
24. A. Silva. Position automata for Kleene algebra with tests. *Scientific Annals of Computer Science*, 22(2):367–394, 2012.
25. L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. STOC 1973, 1–9.