

Completeness and Incompleteness in nominal Kleene algebra

Dexter Kozen, Konstantinos Mamouras, **Alexandra Silva**

Cornell University, University College London & HasLab INESC TEC

September 28th, 2015

Ramics 2015

Braga, Portugal

Context

- Names are pervasive in computer science;
- Semantics of programming languages (α -equivalence);

$$f(a) = 2 * a \quad g(b) = 2 * b$$

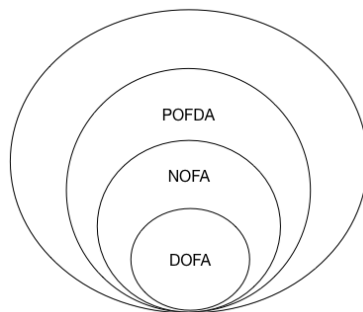
- Range of proposals for sound semantics:
Pistore-Montanari, Gabbay-Pitts, . . .
- Nominal sets (Fraenkel and Mostowski, early twentieth century).

Context

- Francez and Kaminski: finite memory automata.
- Montanari and Pistore: HD-automata.
- Murawski and Tzevelekos: fresh-register automata.
- Bojanczyk, Klin, Lasota: extensive results on nominal automata theory.
- Gabbay and Ciancia: nominal Kleene algebras.
- Kurz, Suzuki, Tuosto: regular expressions for HD-automata.

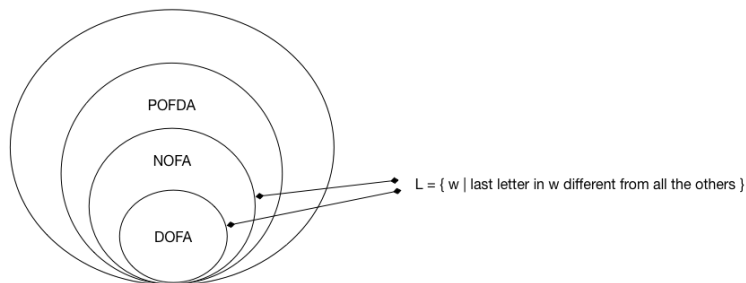
Nominal Chomsky hierarchy

Key point in Polish work: new notion of finiteness, orbit-finiteness.

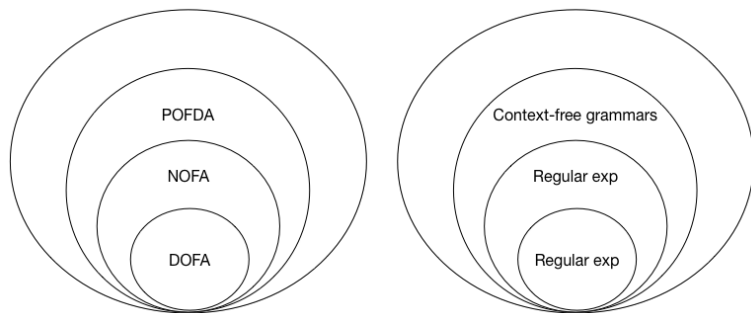


Nominal Chomsky hierarchy

Unexpected things happen with orbit-finiteness.

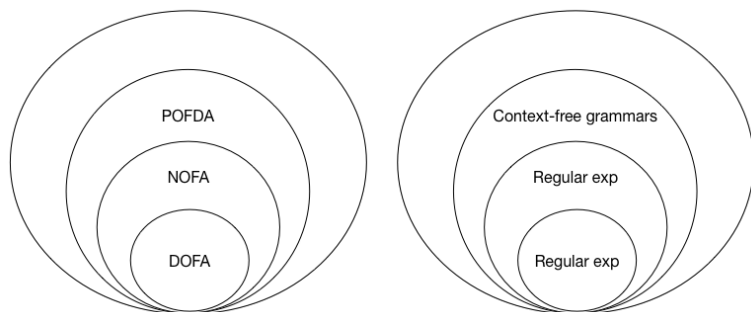


Nominal Chomsky hierarchy



Language hierarchy and correspondence theorems?

Nominal Chomsky hierarchy



Murawski (June 2015): to this day we still do not have a satisfactory notion of nominal regular language.

This talk: some results, many problems...

- Nominal Kleene algebra (Ciancia & Gabbay) axioms are not complete.
- Characterisation of the free Nominal Kleene algebra.
- Nominal Kleene algebra does not give a Kleene Theorem.
- New automaton model for one-sided Kleene Theorem.

Kozen, Mamouras, Silva. *Completeness and Incompleteness of Nominal KA*.

Kozen, Mamouras, Petrisan, Silva. *Nominal Kleene coalgebra*.

This talk: some results, many problems...

- Nominal Kleene algebra (Ciancia & Gabbay) axioms are not complete.
- Characterisation of the free Nominal Kleene algebra.
- Nominal Kleene algebra does not give a Kleene Theorem.
- New automaton model for one-sided Kleene Theorem.

Kozen, Mamouras, Silva. *Completeness and Incompleteness of Nominal KA*.

Kozen, Mamouras, Petrisan, Silva. *Nominal Kleene coalgebra*.

Nominal Sets [Gabbay & Pitts, LICS 1999]

Nominal Sets

- a convenient framework for **name generation**, **binding**, **α -conversion**

Applications

- logic: quantifiers
- programming language semantics: references, objects, pointers, function parameters
- XML document processing
- cryptography: nonces

Group Action

- Let G be a group and X a set
- A **group action** of G on X is a map $G \times X \rightarrow X$ such that

$$\pi(\rho x) = (\pi\rho)x \qquad 1x = x$$

- A **G -set** is a set X equipped with a group action $G \times X \rightarrow X$
- $f : X \rightarrow Y$ is **equivariant** if $f \circ \pi = \pi \circ f$ for all $\pi \in G$

Nominal Sets

- Let \mathbb{A} be a countably infinite set of **atoms**
- Let G be the group of all **finite** permutations of \mathbb{A} (permutations generated by transpositions (ab))
- If G acts on X , say that $A \subseteq \mathbb{A}$ **supports** $x \in X$ if

$$\text{Fix } A \subseteq \text{fix } x$$

where $\text{fix } x = \{\pi \in G \mid \pi x = x\}$ and $\text{Fix } A = \bigcap_{x \in A} \text{fix } x$

Nominal Sets

- $x \in X$ has **finite support** if there is a finite $A \subseteq \mathbb{A}$ that supports x
- If $x \in X$ has finite support, then it has a minimum supporting set $\text{supp } x$, the **support** of x
- Write $a\#x$ and say a is **fresh** for x if $a \notin \text{supp } x$
- A **nominal set** is a set X with a group action of G such that every element has finite support

Nominal Sets

Example

- $\mathbb{A} = \{\text{variables}\}$
- $X = \{\lambda\text{-terms over } \mathbb{A}\}$
- If $\pi \in G$ and $\pi a = a$ for $a \in \text{FV}(x)$, then $\pi x = x$
(α -conversion)
- $A \subseteq \mathbb{A}$ supports $x \iff \text{FV}(x) \subseteq A$
- $\text{supp } x = \text{FV}(x)$
- $a \# x$ iff $a \notin \text{FV}(x)$

$$(b\ c) ((\lambda b.a(bb))(\lambda b.a(bb))) = (\lambda c.a(cc))(\lambda c.a(cc))$$

More examples

- The set \mathbb{A} is a G -set under the group action $\pi a = \pi(a)$. It is a nominal set with $\text{supp}(a) = \{a\}$.
- The set $\mathcal{P}\mathbb{A}$ is a G -set, but not a nominal set.
- The set $\mathcal{P}_{fs}\mathbb{A}$ of finite and co-finite subsets of \mathbb{A} is a nominal set.

Kleene Algebra

Idempotent Semiring Axioms

$$p + (q + r) = (p + q) + r$$

$$p + q = q + p$$

$$p + 0 = p$$

$$p + p = p$$

$$p(q + r) = pq + pr$$

$$(p + q)r = pr + qr$$

$$p(qr) = (pq)r$$

$$1p = p1 = p$$

$$p0 = 0p = 0$$

$$a \leq b \stackrel{\Delta}{\iff} a + b = b$$

Axioms for *

$$1 + pp^* \leq p^*$$

$$1 + p^*p \leq p^*$$

$$q + px \leq x \Rightarrow p^*q \leq x$$

$$q + xp \leq x \Rightarrow qp^* \leq x$$

Standard Model

Regular sets of strings over Σ

$$A + B = A \cup B$$

$$AB = \{xy \mid x \in A, y \in B\}$$

$$A^* = \bigcup_{n \geq 0} A^n = A^0 \cup A^1 \cup A^2 \cup \dots$$

$$1 = \{\varepsilon\}$$

$$0 = \emptyset$$

This is the **free KA** on generators Σ

Other Models

- Relational models
- Trace models used in semantics
- $(\min, +)$ algebra used in shortest path algorithms
- $(\max, +)$ algebra used in coding
- Convex sets used in computational geometry (Iwano & Steiglitz 90)
- Matrix algebras

Nominal KA [Gabbay & Ciancia 2011]

A **nominal Kleene algebra** (NKA) over atoms \mathbb{A} is a structure

$$(K, +, \cdot, *, 0, 1, \nu)$$

with $\nu : \mathbb{A} \times K \rightarrow K$ such that

- K is a nominal set over \mathbb{A}
- the KA operations and ν are equivariant:

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

equivalently, every $\pi \in G$ is an automorphism of K

- all the KA axioms are satisfied and ν satisfies...

Nominal Axioms [Gabbay & Ciancia 2011]

Odersky style axioms	interaction with KA operators
$a\#e \Rightarrow \nu a.e = e$ $\nu a.\nu b.e = \nu b.\nu a.e$ $a\#e \Rightarrow \nu b.e = \nu a.(a\ b)e$	$\nu a.(d + e) = \nu a.d + \nu a.e$ $a\#e \Rightarrow (\nu a.d)e = \nu a.de$ $a\#e \Rightarrow e(\nu a.d) = \nu a.ed$

Nominal KA [Gabbay & Ciancia 2011]

Expressions

$$e ::= a \in \Sigma \mid e + e \mid ee \mid e^* \mid 0 \mid 1 \mid \nu a.e$$

The operator νa is a **binding operator** whose scope is e

The set of expressions over Σ is denoted Exp_Σ

ν -strings

A ν -string is an expression with no occurrence of $+$, $*$, 0 , or 1 (except to denote the null string, in which case we use ε)

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

The set of ν -strings over Σ is denoted Σ^ν .

Nominal Language Model [Gabbay & Ciancia 2011]

$$NL : \text{Exp}_{\mathbb{A}} \rightarrow \mathcal{P}(\mathbb{A}^*)$$

Example:

$$NL(\nu a.ab) = \{ab \mid a \neq b\}$$

$$NL((\nu a.ab)(\nu a.ab)) = \{abcb \mid a, c \in \mathbb{A} \text{ distinct and different than } b\}$$

Care must be taken when defining product to avoid capture!

Nominal Language Model [Gabbay & Ciancia 2011]

$$NL : \text{Exp}_{\mathbb{A}} \rightarrow \mathcal{P}(\mathbb{A}^*)$$

Example:

$$NL(\nu a.ab) = \{ab \mid a \neq b\}$$

$$NL((\nu a.ab)(\nu a.ab)) = \{abcb \mid a, c \in \mathbb{A} \text{ distinct and different than } b\}$$

Care must be taken when defining product to avoid capture!

Nominal Language Model [Gabbay & Ciancia 2011]

$$NL : \text{Exp}_{\mathbb{A}} \rightarrow \mathcal{P}(\mathbb{A}^*)$$

Example:

$$NL(\nu a.ab) = \{ab \mid a \neq b\}$$

$$NL((\nu a.ab)(\nu a.ab)) = \{abcb \mid a, c \in \mathbb{A} \text{ distinct and different than } b\}$$

Care must be taken when defining product to avoid capture!

Intermediate interpretation as sets of ν -strings over \mathbb{A}

$$I : \text{Exp}_{\mathbb{A}} \rightarrow \mathcal{P}(\mathbb{A}^{\nu})$$

$+$, \cdot , $*$, 0 , and 1 have their usual set-theoretic interpretations, and

$$I(\nu a.e) = \{\nu a.x \mid x \in I(e)\} \qquad I(a) = \{a\}.$$

Examples

$$I(\nu a.a) = \{\nu a.a\}$$

$$I(\nu a.\nu b.(a + b)) = \{\nu a.\nu b.a, \nu a.\nu b.b\}$$

$$I(\nu a.(\nu b.ab)(a + b)) = \{\nu a.(\nu b.ab)a, \nu a.(\nu b.ab)b\}$$

$$I(\nu a.(ab)^*) = \{\nu a.\varepsilon, \nu a.ab, \nu a.abab, \nu a.ababab, \dots\}$$

$$I((\nu a.ab)^*) = \{\varepsilon, \nu a.ab, (\nu a.ab)(\nu a.ab), (\nu a.ab)(\nu a.ab)(\nu a.ab), \dots\}$$

Intermediate interpretation as sets of ν -strings over \mathbb{A}

$$I : \text{Exp}_{\mathbb{A}} \rightarrow \mathcal{P}(\mathbb{A}^{\nu})$$

$+$, \cdot , $*$, 0 , and 1 have their usual set-theoretic interpretations, and

$$I(\nu a.e) = \{\nu a.x \mid x \in I(e)\} \qquad I(a) = \{a\}.$$

Examples

$$I(\nu a.a) = \{\nu a.a\}$$

$$I(\nu a.\nu b.(a + b)) = \{\nu a.\nu b.a, \nu a.\nu b.b\}$$

$$I(\nu a.(\nu b.ab)(a + b)) = \{\nu a.(\nu b.ab)a, \nu a.(\nu b.ab)b\}$$

$$I(\nu a.(ab)^*) = \{\nu a.\varepsilon, \nu a.ab, \nu a.abab, \nu a.ababab, \dots\}$$

$$I((\nu a.ab)^*) = \{\varepsilon, \nu a.ab, (\nu a.ab)(\nu a.ab), (\nu a.ab)(\nu a.ab)(\nu a.ab), \dots\}$$

Nominal Language Model [Gabbay & Ciancia 2011]

$$NL : \mathbb{A}^\nu \rightarrow \mathcal{P}(\mathbb{A}^*)$$

- α -convert so that all bindings in x are distinct and different from free variables in x
- delete all binding operators νa to obtain $x' \in \mathbb{A}^*$
- $NL(x) = \{\pi(x') \mid \pi \in \text{fix FV}(x)\}$
- $NL(e) = \bigcup_{x \in I(e)} NL(x)$

Example

$$\begin{aligned} &NL((\nu a.ab)(\nu a.ab)(\nu a.ab)) \\ &= \{abcdbdb \mid a, c, d \in \mathbb{A} \text{ distinct and different from } b\} \end{aligned}$$

Completeness and Incompleteness

Lemma

For $x, y \in \mathbb{A}^\nu$, $\vdash x = y$ implies $NL(x) = NL(y)$.

Incompleteness

$\nexists a \leq \nu a.a$ but $NL(a) = \{a\} \subseteq \mathbb{A} = NL(\nu a.a)$

Alternative Nominal Language Model

Let Σ and \mathbb{A} be countably infinite **disjoint** sets, $a, b, c, \dots \in \mathbb{A}$, $x, y, z, \dots \in \Sigma$, and $u, v, w, \dots \in (\Sigma \cup \mathbb{A})^*$. Quantification is only over Σ .

A **language** is a subset $A \subseteq (\Sigma \cup \mathbb{A})^*$ such that $\pi A = A$ for all $\pi \in G$. The set of languages is denoted \mathcal{L} .

$$AB = \{uv \mid u \in A, v \in B, \text{FV}(u) \cap \text{FV}(v) \cap \mathbb{A} = \emptyset\}$$

$$\nu x.A = \{w[a/x] \mid w \in A, a \in \mathbb{A} - \text{FV}(w)\}, x \in \Sigma$$

Completeness

Theorem

The axioms of nominal Kleene algebra are sound and complete for the equational theory of nominal Kleene algebras and for the equational theory of the alternative language model:

$$\vdash e_1 = e_2 \iff AL(e_1) = AL(e_2)$$

The alternative language model is the **free nominal KA**.

Completeness

- ▶ exposing bound variables
- ▶ scope configuration
- ▶ canonical choice of bound variables
- ▶ semilattice identities

Determining Semilattice Identities

- Any substring of the form $\nu a.x$ of a ν -string generated by e_1 or e_2 must be generated by a subexpression $\nu a.d$
- There may be several different subexpressions of this form
- The sets of ν -strings generated by the ν -subexpressions could satisfy various semilattice identities, and we may have to know these identities to prove equivalence

Determining Semilattice Identities

Example

Consider $c_1 + c_2$ and $d_1 + d_2 + d_3$, where

$$c_1 = \nu a.a(aa)^*$$

$$c_2 = \nu a.aa(aa)^*$$

$$d_1 = \nu a.a(aaa)^*$$

$$d_2 = \nu a.aa(aaa)^*$$

$$d_3 = \nu a.aaa(aaa)^*$$

- c_i generates strings with $i \bmod 2$ a 's
- d_i generates strings with $i \bmod 3$ a 's
- Both $c_1 + c_2$ and $d_1 + d_2 + d_3$ generate all nonempty strings of a 's, but in different ways

Determining Semilattice Identities

Express every ν -subexpression in e_1 or e_2 as a **sum of atoms** of the Boolean algebra on sets of ν -strings generated by these ν -subexpressions.

In the example above, the atoms are

$$b_i = \nu a.a^i(a^6)^*, \quad 1 \leq i \leq 6$$

so b_i generates strings with $i \bmod 6$ a 's.

Determining Semilattice Identities

Rewriting as sums of atoms,

$$c_1 = b_1 + b_3 + b_5$$

$$c_2 = b_2 + b_4 + b_6$$

$$d_1 = b_1 + b_4$$

$$d_2 = b_2 + b_5$$

$$d_3 = b_3 + b_6.$$

The equivalences are provable in pure KA plus the nominal axiom $\nu a.(d + e) = \nu a.d + \nu a.e$. This gives

$$c_1 + c_2 = (b_1 + b_3 + b_5) + (b_2 + b_4 + b_6)$$

$$d_1 + d_2 + d_3 = (b_1 + b_4) + (b_2 + b_5) + (b_3 + b_6)$$

Determining Semilattice Identities

Now observe

- any ν -string $\nu a.x$ generated by e_1 or e_2 is generated by exactly one atom $\nu a.e$
- we can treat $\nu a.e$ as atomic!
- we can even replace each atom $\nu a.e$ by a single letter $a_{\nu a.e}$ in e_1 and e_2 , and the resulting expressions are equivalent, therefore provable
- for expressions of ν -depth greater than one, perform the construction inductively, innermost scopes first

Conclusions

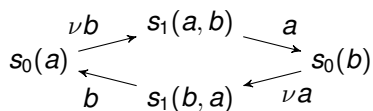
- free language model consisting of regular sets of ν -strings modulo the Gabbay–Ciancia axioms
- new techniques in the completeness proof, e.g.
 - The Boolean algebra generated by finitely many regular sets consists of regular sets and is atomic.
 - Crucial for the normal form: every expression can be written as a sum of atoms.

Conclusions

- nominal versions of the syntactic and semantic Brzozowski derivative
- finitely supported sets of ν -strings modulo the Gabbay–Ciancia axioms form the final coalgebra
- half a Kleene theorem (expressions \Rightarrow automata)
- exponential space decision procedure

Open Problems

- Complexity?
- Other half of the Kleene theorem is false:



The set of ν -strings accepted from state $s_0(a)$ is

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

Requires unbounded ν -depth!

Open Problems

- Can we characterize bounded ν -depth automata in a way that would lead to a converse of the Kleene theorem?
- Can we extend the syntax of expressions to capture sets of unbounded ν -depth? **Yes:**

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

$$Y_{ab} = aX_b$$

...but this leaves us with the task of providing proof rules and proving completeness