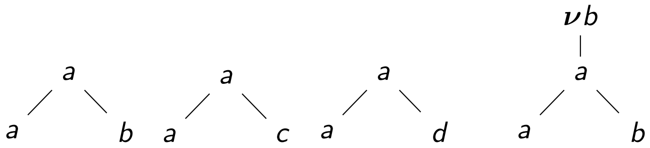
Formalizing Nu-Tree Automata

Leonhard Staut
Advisor: Dominik Kirst
Supervisor: Prof. Dr. Gert Smolka
Saarland University
April 28, 2017

Introduction

- Topic:
  - Language of trees over an infinite alphabet $\mathbb{A}$ undecidable in general
  - Focus on subclass of trees of similar structure obtained by systematically permuting names
- Technique: $\boldsymbol{\nu}$-Trees
  - Trees with binders
  - Using ideas from nominal sets



Nominal sets

Definition (sym action)

Let X be a set, $x \in X$,
A function $(\cdot) : \mathtt{Sym}(\mathbb{A}) \times X \to X$ is called a *sym action*, if

$$\mathrm{id} \cdot x = x \qquad \pi \cdot (\pi' \cdot x) = (\pi \circ \pi') \cdot x, \ \forall \pi, \pi' \in \mathtt{Sym}(\mathbb{A})$$

X is called a *sym set*, if such a function exists.

Definition (nominal set)

Let X be a sym set, $A \subseteq \mathbb{A}$
A *supports* $x \in X$, if $\pi \cdot x = x$ holds $\forall \pi \in \mathtt{Sym}(\mathbb{A})$ with $\forall a \in A.\ \pi(a) = a$.
X is called *nominal*, if every $x \in X$ has a finite support.

Definition (equivariance)

Functions $f \subseteq X \times Y$ for nominal X, Y are called *equivariant* if $\pi \cdot f(x) = f(\pi \cdot x)$.

$\boldsymbol{\nu}$-Tree

Definition ($\boldsymbol{\nu}$-Tree)

The set $\boldsymbol{\nu}$-Tree is defined inductively by

$$n ::= a_k n_1 \dots n_k \mid \boldsymbol{\nu} a_k . n$$

where a_k ranges over the ranked alphabet $\mathbb{A}$

Definition (Names)

Let n be a $\boldsymbol{\nu}$-tree. FN(n) are the *free names* of n.

- $\mathbb{A}$ countably infinite in every rank
  - $\boldsymbol{\nu}$-Tree automata use a ranked alphabet
- Sym action:

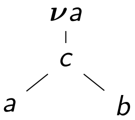
$$\pi \cdot (a_k n_1 \dots n_k) = (\pi a_k)(\pi \cdot n_1) \dots (\pi \cdot n_k)$$
$$\pi \cdot (\boldsymbol{\nu} a_k . n) = \boldsymbol{\nu}(\pi a_k).(\pi \cdot n)$$

- The set $\boldsymbol{\nu}$-Tree is nominal

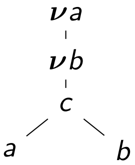
$\boldsymbol{\nu}$-Tree Denotation examples

- 

$\rightsquigarrow \quad \{\, a \mid a \in \mathbb{A} \,\} = \{\, a,\ b,\ c,\ \ldots \,\}$

- 

$\rightsquigarrow \quad \{\, c\ ab \mid a \in \mathbb{A}; a \neq b \,\}$

- 

$\rightsquigarrow \quad \{\, c\ ab \mid a, b \in \mathbb{A}; a \neq b \,\}$

- formalize with $[\![-]\!] : \boldsymbol{\nu}\text{-Tree} \to \mathcal{P}(\mathbb{A}\text{-Tree})$

$\boldsymbol{\nu}$-Tree Denotation

- Parameterize with an accumulator list carrying already used names

Definition ($\boldsymbol{\nu}$-Tree Denotation)

$$\frac{t_i \in [\![n_i]\!]_{a_k::A}}{a_k t_1 \ldots t_k \in [\![a_k n_1 \ldots n_k]\!]_A}$$

$$\frac{t \in [\![(a_k b_k) \cdot n]\!]_{b_k::A} \qquad b_k \notin A \quad b_k \notin FN(\boldsymbol{\nu} a_k.n)}{t \in [\![\boldsymbol{\nu} a_k.n]\!]_A}$$

Denotation equivariance

Theorem (Denotation equivariance)

$\forall\ t\ n\ \pi.\ \pi[\![n]\!]_A = [\![\pi \cdot n]\!]_{\pi \cdot A}$

- Denotation is compatible with permutations

Lemma

- $\forall\ \pi\ n.\ \mathrm{FN}(\pi \cdot n) = \pi \cdot \mathrm{FN}(n)$
- $\forall\ A\ a\ \pi.\ a \in (\pi \cdot A) \leftrightarrow (\pi^{-1} a) \in A$

Denotation renaming

Theorem (Denotation renaming)

$\forall\ n\ \pi.\ \pi \text{ fixes } \mathsf{FN}(n) \rightarrow [\![\pi \cdot n]\!] = [\![n]\!]$

- Similar to α-equivalence

Lemma

$\forall\ n\ \pi\ \pi'.\ \ (\forall a.\ \pi a = \pi' a) \rightarrow \pi \cdot n = \pi' \cdot n$

Corollary

$\forall\ n\ n'.\ n \approx_\alpha n' \rightarrow [\![n]\!] = [\![n']\!]$

ν-Tree Implementation

- `Definition Name :=  nat * nat.`
- `Definition Action := Name -> Name.`
- `Definition Perm p :=`
  `∀a, rk a = rk (p a) ∧ ∃p', Inv p p'.`

Alternatively:

- `Definition Name := nat.`
- `Definition Action := nat -> nat.`
- `Definition Perm p := ∃p', Inv p p'.`

Induction for ν-Trees

- Want tree to be well-ranked
- Tree induction not by default
-
```
Inductive NuTreeWr :  NuTree → Prop :=
  Trwr a l :  (∀n, n el l → NuTreeWr n) →
                rk a = |l| → NuTreeWr (Tr a l)
 |Nuwr a n :  NuTreeWr n → NuTreeWr (Nu a n).
```

Future directions

- Denotation
  - Decidability of $t \in [\![n]\!]$, and $[\![n]\!] = [\![n']\!]$
- Formalize the automata model
  - Decidability of acceptance and emptiness
- Generalize nominal kleene algebra to trees [Kozen et al., 2015]

References

- Gabbay, M. J. and Ciancia, V. (2011). *Freshness and Name-Restriction in Sets of Traces with Names*, pages 365–380. Springer Berlin Heidelberg.
- Grumberg, O., Kupferman, O., and Sheinvald, S. (2010). *Variable Automata over Infinite Alphabets*, pages 561–572. Springer Berlin Heidelberg.
- Jurdzinski, M. and Lazic, R. (2008). Alternating automata on data trees and xpath satisfiability. *CoRR*.
- Kozen, D., Mamouras, K., and Silva, A. (2015). *Completeness and Incompleteness in Nominal Kleene Algebra*, pages 51–66. Springer International Publishing.
- Pitts, A. M. (2013). *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge University Press.