Higher order model checking meets implicit automata

Abhishek De

University of Birmingham, UK

j.w.w. Charles Grellois, Lê Thành Dung (Tito) Nguyên, Cécilia Pradic

Higher order model checking

Given a tree $\langle G \rangle$ generated by a recursion scheme G and an alternating parity tree automaton $\mathcal{A}$, does $\mathcal{A}$ accept $\langle G \rangle$?

Decidable!

- Automata-theoretic methods [HMOS'14]
- Using intersection types [KO'09]
- Using Krivine machines [SW'16]

Recursion schemes

Typed grammars that generate potentially infinite ranked trees.

Example

Signature $\Sigma = \{a : 2, b : 1, c : 0\}$

$$G = \begin{cases} S \to Fabc \\ Fxyz \to xz(Fxy(yz)) \end{cases}$$

$$\langle G \rangle =$$

```
  a
 / \
c   a
   / \
  b   a
  |  / \
  c b   a
    |    \
    b    ...
    |
    c
```

How do intersection types get in this?

1. HORS = Bohm trees of λY terms
2. Given an alternating parity automaton $\mathcal{A}$ define an intersection-type system of λ^{∞}
3. This type system defines a denotational semantics $\lfloor \bullet \rfloor_{\mathcal{A}}$
4. $\lfloor t \rfloor_{\mathcal{A}}$ is computable for λY term t
5. $\mathcal{A}$ accepts tree defined by HORS corresponding to t iff $q_0 \in \lfloor t \rfloor_{\mathcal{A}}$

Finite words in STLC

Suppose $\Sigma = \{a, b\}$

Fix $a := o \to o, b := o \to o, \epsilon := o$

The word aab is represented by

$$\lambda x^a \lambda y^b \lambda z^\epsilon x(x(yz))$$

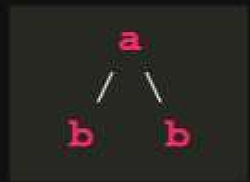
$$\boxed{String_\Sigma : (o \to o) \to (o \to o) \to o \to o}$$

Thm (Hillebrand and Kallenakis'96). L is regular iff it can be represented by a term of type $String_\Sigma[o := \tau] \to Bool$

Finite ranked trees in STLC

Suppose $\Sigma = \{a : 2, b : 0\}$

Fix $a := o \times o \rightarrow o, b := o$

The tree 

is represented by $\lambda x^a \lambda y^b xyy$

$$\boxed{Trees_\Sigma : (o \times o \rightarrow o) \rightarrow o \rightarrow o}$$

Thm (Folklore?). L is regular iff it can be represented by a term of type

$$Trees_\Sigma[o := \tau] \rightarrow Bool$$

HORS in λY

Signature $\Sigma = \{a : 2, b : 1, c : 0\}$

Fix $a := o \times o \to o, b := o \to o, c := o$

$$G = \begin{cases} S \to Fabc \\ Fxyz \to xz(Fxy(yz)) \end{cases}$$

$$M = \lambda u^a \lambda v^b \lambda w^c . Y\,(\lambda F \lambda x \lambda y \lambda z . xz(Fxy(yz)))\,uvw$$

$$\langle G \rangle = BT(M)$$

Simply-typed λ^∞

coterms $t, u ::= x \in \mathsf{Var} \mid \lambda x.t \mid tu$

Finite set of variables

types $\sigma, \tau ::= o \mid \sigma \to \tau$

Regular coterms : coterms with finitely many subterms.

λY as a regular coterm:

$$YM = M(M(\dots))$$

$$\Sigma = \{a, b\}$$

$$a := o \to o, b := o \to o$$

$$(ab)^\omega := \lambda x^a \lambda y^b x(y(x(\dots)))$$

$$\boxed{Stream_\Sigma := (o \to o) \to (o \to o) \to o}$$

Fix $\mathcal{A} = (Q, \Sigma, \Delta, q_0, \kappa : Q \to \{1, 2, \ldots, k\})$

Interpretation of types

$\lfloor o \rfloor_{\mathcal{A}} = Q$

$\lfloor \sigma \to \tau \rfloor_{\mathcal{A}} \subseteq \mathcal{P}(\{0, 1, \ldots, k\} \times \lfloor \sigma \rfloor_{\mathcal{A}}) \times \lfloor \tau \rfloor_{\mathcal{A}}$

Example

$$\lfloor a \rfloor = \{(X, q) \mid X \models \delta(q, a)\}$$

An intersection type system

Sequents of the form

$$x_1 : X_1 :: \tau_1, \ldots, x_n : X_n :: \tau_n \vdash t : \alpha :: \tau$$

$$X_i \subseteq \lfloor \tau_i \rfloor_{\mathcal{A}} \qquad \alpha \in \lfloor \tau \rfloor_{\mathcal{A}}$$

Type derivation coinductively generated by

$$(\text{Var})\ \frac{\exists \alpha' \in X.\, \alpha \leq \alpha'}{x : X :: \tau \vdash x : \alpha :: \tau} \qquad (\text{Abs})\ \frac{\Gamma, x : X :: \sigma \vdash t : \alpha :: \tau}{\Gamma \vdash \lambda x.\, t : X \to \alpha :: \sigma \to \tau}$$

$$(\text{App})\ \frac{\Gamma \vdash t : \{\beta_1, \beta_2\} \to \alpha :: \sigma \to \tau \quad \Gamma \vdash u : \beta_1 :: \sigma \quad \Gamma \vdash u : \beta_2 :: \sigma}{\Gamma, \Gamma, \Gamma \vdash t\, u : \alpha :: \tau}$$

A denotational semantics

$$\boxed{\lfloor t \rfloor_{\mathcal{A}} := \{\alpha \mid \varnothing \vdash t : \alpha :: \tau\}}$$

Prop. $\lfloor t \rfloor_{\mathcal{A}} \subseteq \lfloor \tau \rfloor_{\mathcal{A}}$ and is down-closed.

Theorem. $\lfloor t \rfloor_{\mathcal{A}}$ is computable for a regular coterm t.

Conjecture. If $t \to_{\beta}^{\infty} t'$ then $\lfloor t \rfloor_{\mathcal{A}} = \lfloor t' \rfloor_{\mathcal{A}}$.

Solving HOMC

- Let G be a recursion scheme. Take λY term t such that $BT(t) = \langle G \rangle$
- Unfold t to a λ^∞ term t'
- We have $t' \to_\beta^\infty BT(t)$
- $\mathcal{A}$ accepts $BT(t)$ iff $q_0 \in \lfloor BT(t) \rfloor_\mathcal{A}$
- But $\lfloor t' \rfloor_\mathcal{A} = \lfloor BT(t) \rfloor_\mathcal{A}$ and it is computable.
- Done!

Implicit (ω)-automata

We will go through (sequential) transducers:

Thm. For every s-transducer language $f : \Sigma^\omega \to \Gamma^\omega$ there is a regular coterm $t : Stream_\Sigma \to Stream_\Gamma$ such that

$$t\overline{w} \to_\beta^\infty \overline{f(w)}$$

Easy. Code the matrix of the transducer.

Thm. Every regular coterm $t : Stream_{\Sigma} \to Stream_{\Gamma}$ represents an s-transducer.

Harder. Use the finitary coloured semantics.

Conclusion

Defining a general finitary coloured semantics on the level of infinitary terms.

Makes HOMC easier.

Allows for implicit characterisation of omega-automata.

Future Work

Solve the conjecture.

Import ideas from Mellies' work on higher-order parity automata?

Import ideas from cut elimination proofs of cyclic proof theory?

Deterministic automata?

Thanks!

Questions?