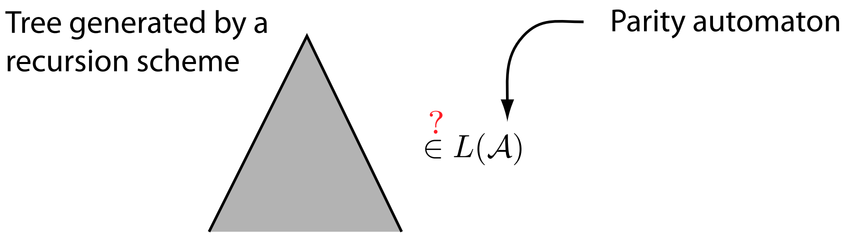
A Model-Theoretic Approach to Model Checking Recursion Schemes

Sylvain Salvati, Balaguru Srivathsan, and Igor Walukiewicz

LaBRI, Bordeaux



- Thm [Ong, LICS'06]: This problem is decidable
- Kobayashi [POPL'09]: Type system for deciding this problem in a special case of automata with trivial acceptance condition.
- Kobayashi & Ong [LICS'09]: Type system for all automata.

Here: Kobayashi's case using models

Recursive schemes

- $\Sigma = \{a, b, \dots\}$ constants (of type $0^n \to 0$ or 0).
- $\mathcal{N} = \{F, G, \dots\}$ nonterminals (typed variables)
- $S \in \mathcal{N}$ starting symbol (of type 0)
- $\mathcal{R} : \mathcal{N} \to \mathit{Terms}$ a rule for every nonterminal

$$\mathcal{R}(F) = \lambda \vec{x}.M$$

its type should be that of F, and its free variables should be included in $\mathcal{N}$.

Example

- $\Sigma = \{a : 0 \to 0 \to 0,\ b : 0 \to 0,\ c : 0\}, \quad \mathcal{N} = \{S : 0,\ F : 0 \to 0\}$
- $R(F) = \lambda x.ax(F(bx)), \quad R(S) = Fc.$

Intuitively the meaning of the scheme is

$$Y(\lambda F.R(F))c.$$

Simply typed λY-calculus with fixpoints

- Types: 0 is a type, and $\alpha \to \beta$ is a type if α, β types.
- Constants: ω^α and $Y^{(\alpha\to\alpha)\to\alpha}$ for every type α.
- Terms: c^α, x^α, MN, $\lambda x^\alpha.M$.

Model: $\mathcal{D} = \langle \{D^\alpha\}_{\alpha\in\mathcal{T}}, \rho\rangle$

- D^0 is a complete lattice;
- $D^{\alpha\to\beta}$ is the complete lattice of monotone functions from D^α to D^β ordered coordinatewise;
- $\rho(\omega^\alpha)$ is the greatest element of D^α.
- $\rho(Y^{(\alpha\to\alpha)\to\alpha})$ is a mapping assigning to a function $f \in D^{\alpha\to\alpha}$ its fixpoint.

- GFP model when Y assigns greatest fixpoints.
- Finitary model when every D^α is finite.

Interpretation of terms in a model

- $[\![c]\!]_{\mathcal{D}}^{\upsilon} = \rho(c)$
- $[\![x^\alpha]\!]_{\mathcal{D}}^{\upsilon} = \upsilon(x^\alpha)$
- $[\![MN]\!]_{\mathcal{D}}^{\upsilon} = [\![M]\!]_{\mathcal{D}}^{\upsilon}[\![N]\!]_{\mathcal{D}}^{\upsilon}$
- $[\![\lambda x^\alpha.M]\!]_{\mathcal{D}}^{\upsilon}$ is a function mapping an element $d \in D^\alpha$ to $[\![M]\!]_{\mathcal{D}}^{\upsilon[d/x^\alpha]}$.
  (this is a monotone function).

β-reduction $(\lambda x.M)N \to_\beta M[N/x]$

η-reduction $(\lambda x.Mx) \to_\eta M$, provided x is not free in M.

δ-reduction $Y(M) \to_\delta M(YM)$.

Fact

For every model $\mathcal{D}$: if $M =_{\beta,\eta,\delta} N$ then $[\![M]\!]^{\mathcal{D}} = [\![N]\!]^{\mathcal{D}}$.

Böhm trees

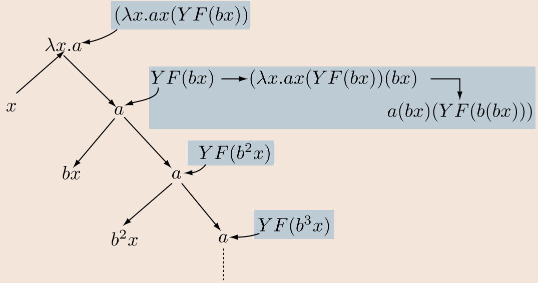
Böhm tree of a term

A Böhm tree of a term M is:

- if $M \to^*_{\beta\delta} \lambda\vec{x}.KN_1 \dots N_i$ with K a variable or a constant then the root of $BT(M)$ is labelled by $\lambda\vec{x}.K$ and has $BT(N_1), \dots, BT(N_i)$ as a sequence of its children.
- If M is not solvable then $BT(M) = \omega^\alpha$, where α is the type of M.

Example

$Y(\lambda F.\lambda x.ax(F(bx))) \ : \ 0 \to 0$



Böhm trees

Böhm tree of a term

A Böhm tree of a term M is:

- if $M \to^*_{\beta\delta} \lambda\vec{x}.KN_1 \ldots N_i$ with K a variable or a constant then the root of $BT(M)$ is labelled by $\lambda\vec{x}.K$ and has $BT(N_1), \ldots, BT(N_i)$ as a sequence of its children.
- If M is not solvable then $BT(M) = \omega^\alpha$, where α is the type of M.

Theorem [?]

For every finitary GFP-model $\mathcal{D}$: if $BT(M) \equiv BT(N)$ then $[\![M]\!]^{\mathcal{D}} = [\![N]\!]^{\mathcal{D}}$.

Approximate Böhm tree

Approximate Böhm tree

$ABT(M)$ is defined by

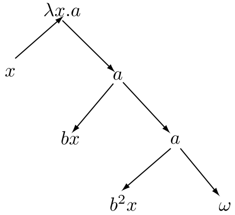
- If M is in head normal form, i.e. $M \equiv \lambda\vec{x}.KN_1 \dots N_k$ with K a constant or a variable then $ABT(M)$ has the root labelled with $\lambda\vec{x}.K$ and $ABT(N_1), \dots, ABT(N_k)$ as its children.
- Otherwise $ABT(M)$ is ω^α; where α is the type of M.

Remark

$ABT(M)$ is a λ-term in a $\beta\delta$-normal form.

Lemma

$BT(M) = \bigsqcup\{ABT(N) : N =_{\beta,\delta} M\}$;
we are taking syntactic limit over trees.



Meanings of Böhm trees

Remainder

$BT(M) = \bigsqcup\{ABT(N) : N =_{\beta,\delta} M\}$; here we are taking syntactic limit over trees.

Semantics

$$[\![BT(M)]\!]^{\mathcal{D}} = \bigwedge\{[\![ABT(N)]\!]^{\mathcal{D}} : N =_{\beta,\delta} M\}$$

Theorem [?]

If $\mathcal{D}$ is a finitary GFP model then: $[\![M]\!]^{\mathcal{D}} = [\![BT(M)]\!]^{\mathcal{D}}$.

Proof of the theorem

Theorem

If $\mathcal{D}$ is a finitary GFP model then: $[\![M]\!]^{\mathcal{D}} = [\![BT(M)]\!]^{\mathcal{D}}$.

Proof $[\![BT(M)]\!] \geq [\![M]\!]$

- $[\![BT(M)]\!]^{\mathcal{D}} = \bigwedge\{[\![ABT(N)]\!]^{\mathcal{D}} : N =_{\beta,\delta} M\}$.
- $[\![ABT(N)]\!]^{\mathcal{D}} \geq [\![N]\!]^{\mathcal{D}} = [\![M]\!]^{\mathcal{D}}$.

Proof $[\![M]\!] \geq [\![BT(M)]\!]$

- Let N be a term of type $\alpha \to \alpha$ without occurrences of Y constants. Define *iterate*$^i(N)$ to be $N(\dots(N\omega^\alpha)\dots)$. In general, define *iterate*$^i(M)$ as result of repeatedly replacing all YN by *iterate*$^i(N)$.
- If $\mathcal{D}$ is a finitary GFP model then there is i such that $[\![M]\!]^{\mathcal{D}} = [\![\textit{iterate}^i(M)]\!]^{\mathcal{D}}$.

$$[\![M]\!] = [\![\textit{iterate}^i(M)]\!] = [\![BT(\textit{iterate}^i(M))]\!] \geq [\![BT(M)]\!]$$

Back to recursion schemes

Recursive schemes

- $\mathcal{R} : \mathcal{N} \to \mathit{Terms}$ a definition rule for every nonterminal

$$\mathcal{R}(F) = \lambda\vec{x}.M$$

its type should be correct, and its free variables should be included in $\mathcal{N}$.

Translation to λY-terms

$$\begin{aligned}
T_1 &= Y(\lambda F_1.\mathcal{R}(F_1)) \\
T_2 &= Y(\lambda F_2.\mathcal{R}(F_2)[T_1/F_1]) \\
&\vdots \\
T_n &= Y(\lambda F_n.(\ldots((\mathcal{R}(F_n)[T_1/F_1])[T_2/F_2])\ldots)[T_{n-1}/F_{n-1}])
\end{aligned}$$

Fact

If F_n is the starting symbol of the grammar then $BT(T_n)$ is the tree generated by the scheme.

Half way through

We have

1. Models $\mathcal{D} = ({D^\alpha}_{\alpha \in \mathcal{T}}, \rho)$ interpreting fixpoint operators.
2. Definition of a Böhm tree of a λY-term: $BT(M)$.
3. Models are capable of talking about Böhm trees:

$$[\![M]\!]^{\mathcal{D}} = [\![BT(M)]\!]^{\mathcal{D}}$$

4. Translation from recursive schemes to λY-terms:

   $\mathcal{R} \mapsto M$, such that $BT(M)$ is the meaning of $\mathcal{R}$.

We want

- Models for calculating properties of $BT(M)$.
- In particular a model $\mathcal{D}_\mathcal{A}$ such that $[\![M]\!]^{\mathcal{D}_\mathcal{A}}$ tells us if $BT(M)$ is accepted by $\mathcal{A}$.

Automata

Tree signature

Σ has only constants of types 0 or $0^n \to 0$ (and all the constants ω^α, $Y^{(\alpha\to\alpha)\to\alpha}$).
If M is a closed term of type 0 then $BT(M)$ is a ranked tree.

Automaton

Let $\Sigma = \Sigma_0 \cup \Sigma_2$ with Σ_0 constants of type 0 and Σ_2 of type $0 \to 0 \to 0$.

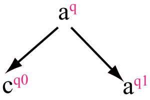
$$\mathcal{A} = \langle Q, \Sigma, q^0 \in Q, \delta_1 : Q \times \Sigma_0 \to \{ff, tt\}, \delta_2 : Q \times \Sigma_2 \to \mathcal{P}(Q^2)\rangle$$

Run of $\mathcal{A}$ on $t : \{0,1\}^* \to \Sigma$

- $r(\varepsilon) = q^0$
- $(r(w0), r(w1)) \in \delta_2(t(w), r(w))$ if w is an internal node.

A run is accepting if:

- $\delta_1(r(w), t(w)) = tt$ for every leaf w.



$(q_0, q_1) \in \delta(q, a)$

c^q

$\delta(q, c) = tt$

A model from an automaton

For an automaton $\mathcal{A} = \langle Q, \Sigma, q^0 \in Q, \delta_1 : Q \times \Sigma_0 \to \{ff, tt\}, \delta_2 : Q \times \Sigma_2 \to \mathcal{P}(Q^2)\rangle$ we define a model $\mathcal{D}_\mathcal{A}$.

- $D^0 = \mathcal{P}(Q)$.
- If $c : 0$ then $[\![c]\!] = \{q : \delta_1(q, c) = tt\}$.
- If $a : 0^2 \to 0$ then $[\![a]\!]$ is a function that for $(S_0, S_1) \in \mathcal{P}(Q)^2$ returns

$$\{q : \delta_2(q, a) \in S_0 \times S_1\}$$

Theorem

For every closed term M of type 0:

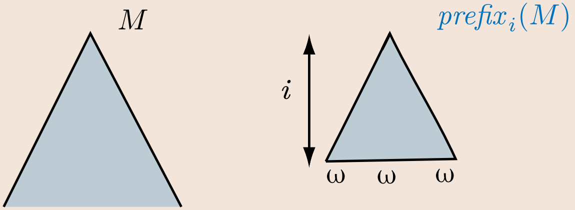
$$BT(M) \in L(\mathcal{A}) \quad \text{iff} \quad q_0 \in [\![M]\!]^{D_\mathcal{A}}$$

If $BT(M) \in L(\mathcal{A})$ then $q_0 \in [\![M]\!]$

Take a run of $\mathcal{A}$ on $BT(M)$ and show that $q^0 \in [\![BT(M)]\!]^{\mathcal{D}_{\mathcal{A}}}$.
This will do as $[\![BT(M)]\!] = [\![M]\!]$.

Approximations

One can define *prefix*$_i(M)$:

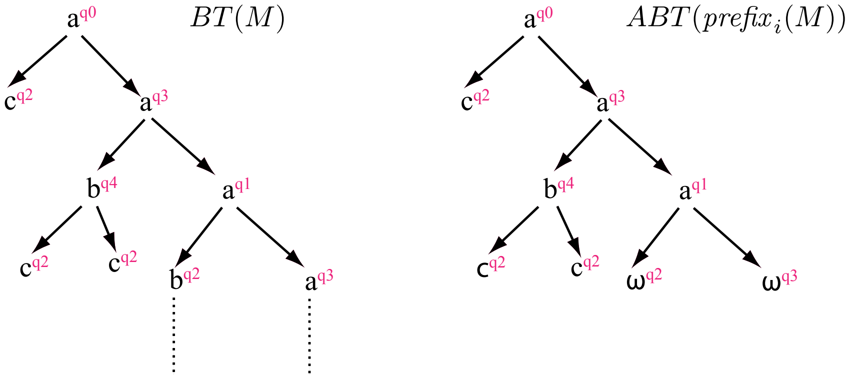


Of course

$$[\![BT(M)]\!] = \bigwedge \{ [\![ABT(\textit{prefix}_i(M))]\!] : i = 1, 2, \dots \}$$

So it is enough to show that $q^0 \in [\![ABT(\textit{prefix}_i(M))]\!]$ for every i.

$q^0 \in [\![ABT(prefix_i(M))]\!]$ FOR EVERY i.



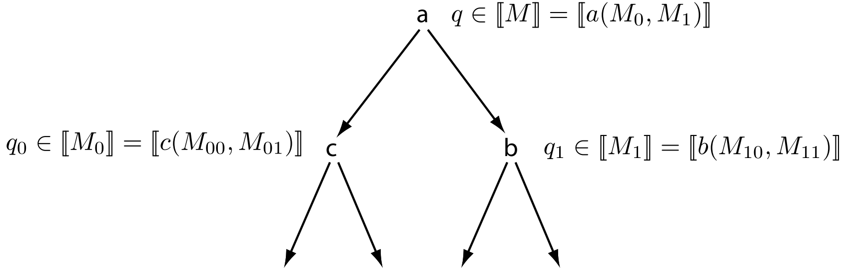
RECALL THAT:

$[\![\omega]\!] = \mathcal{P}(Q) \qquad [\![a]\!](S_0, S_1) = \{q : \delta_2(q, a) \in S_0 \times S_1\}$

If $q_0 \in \llbracket M \rrbracket$ then $BT(M) \in L(\mathcal{A})$

Property of the interpretation

If $q \in \llbracket a(M_0, M_1) \rrbracket$ then there is $(q_0, q_1) \in \delta(q, a)$ such that: $q_0 \in \llbracket M_0 \rrbracket$, and $q_1 \in \llbracket M_1 \rrbracket$.



Putting it all together

Summary

- Given an automaton $\mathcal{A}$ we construct a model $\mathcal{D}_{\mathcal{A}}$.
- For every term of type 0 we have: $q^0 \in [\![M]\!]^{\mathcal{D}_{\mathcal{A}}}$ iff $\mathcal{A}$ accepts $BT(M)$. Here it is important that in the the model $[\![M]\!] = [\![BT(M)]\!]$.
- As the model is finite one can compute $[\![M]\!]^{\mathcal{D}_{\mathcal{A}}}$.
- Recursive schemes can be translated to λY-terms of type 0 (and vice versa).

Remarks

- Standard models are sufficient to do the job.
- This method does not require an induction on the order of the scheme.
- The approach works because the fixpoint defining Böhm tree is the same as the one defining runs of automata.
- It is possible to redo the exercise for LFP models and dual, “prefix”, automata.
- Extension to all parity winning conditions is not obvious as one needs to talk about the winning condition somewhere.