Reducing Higher-order Recursion Scheme Equivalence to Coinductive Higher-Order Constrained Horn Clauses

Jerome Jochems

Department of Computer Science
University of Bristol

jerome.jochems@bristol.ac.uk

28 March 2021

Motivation

- HoCHC is a recent approach to HO program verification [COR18]
- How does it relate to existing approaches like HoRS model checking?
- Decidability of the HoRS equivalence problem is open

Overview

1. Intro to Higher-order Constrained Horn Clauses (HoCHC + co-HoCHC)
2. Intro to Higher-order Recursion Schemes (HoRS)
3. Reduction
   - Eliminate non-termination from HoRS
   - Encode HoRS into HoCHC logic programs
   - Define two coinductive HoCHC instances
4. Correctness outline

Intro to HoCHC

Higher-order Constrained Horn Clauses

- Fragment of higher-order logic
- Horn clauses of HO logic with constraints from a FO background theory
- Unsolvability/unsatisfiability is semi-decidable for semi-decidable background theories [PRO18, OW19]

Intro to HoCHC

Higher-order Constrained Horn Clauses

- Fragment of higher-order logic
- Horn clauses of HO logic with constraints from a FO background theory
- Unsolvability/unsatisfiability is semi-decidable for semi-decidable background theories [PRO18, OW19]

$$\forall x\, y\, z. \quad z = x + y \Rightarrow \mathit{Add}\, x\, y\, z$$
$$\forall f\, s\, n\, m. \quad n \leq 0 \wedge m = s \Rightarrow \mathit{Iter}\, f\, s\, n\, m$$
$$\forall f\, s\, n\, m. \quad n > 0 \wedge \exists p.\, \mathit{Iter}\, f\, s\, (n-1)\, p \wedge f\, n\, p\, m \Rightarrow \mathit{Iter}\, f\, s\, n\, m$$

$$\exists n\, m.\, \mathit{Iter}\, \mathit{Add}\, 0\, n\, m \wedge n > m$$

Intro to HoCHC

Higher-order Constrained Horn Clauses

- Fragment of higher-order logic
- Horn clauses of HO logic with constraints from a FO background theory
- Unsolvability/unsatisfiability is semi-decidable for semi-decidable background theories [PRO18, OW19]

$$
\begin{array}{lcll}
\mathit{Add} & = & \lambda x\, y\, z. & z = x + y \\
\mathit{Iter} & = & \lambda f\, s\, n\, m. & (n \leq 0 \wedge m = s) \vee \\
 & & & (n > 0 \wedge \exists p.\, \mathit{Iter}\, f\, s\, (n-1)\, p \wedge f\, n\, p\, m)
\end{array}
$$

$$\exists n\, m.\, \mathit{Iter}\, \mathit{Add}\, 0\, n\, m \wedge n > m$$

Intro to HoCHC - definition

Simple sorts: $\sigma ::= o \mid \iota \mid \sigma \to \sigma$

Intro to HoCHC - definition

Simple sorts: $\sigma ::= o \mid \iota \mid \sigma \to \sigma$

Relational sorts: $\rho ::= o \mid \iota \to \rho \mid \rho \to \rho$

Intro to HoCHC - definition

Simple sorts: $\sigma ::= o \mid \iota \mid \sigma \to \sigma$

Relational sorts: $\rho ::= o \mid \iota \to \rho \mid \rho \to \rho$

We consider a monotone semantics

Intro to HoCHC - definition

Simple sorts: $\sigma ::= o \mid \iota \mid \sigma \to \sigma$

Relational sorts: $\rho ::= o \mid \iota \to \rho \mid \rho \to \rho$

We consider a monotone semantics

A higher-order constrained Horn clause problem is given by a pair $\langle P, G \rangle$ in which:

- $\vdash P : \Delta$ is a constrained logic program over relational variables Δ
- $\Delta \vdash G$ is a constrained goal formula over Δ

Intro to HoCHC - definition

Simple sorts: $\sigma ::= o \mid \iota \mid \sigma \to \sigma$

Relational sorts: $\rho ::= o \mid \iota \to \rho \mid \rho \to \rho$

We consider a monotone semantics

A higher-order constrained Horn clause problem is given by a pair $\langle P, G \rangle$ in which:

- $\vdash P : \Delta$ is a constrained logic program over relational variables Δ
- $\Delta \vdash G$ is a constrained goal formula over Δ

Axiomatised over a (first-order) background theory Th (with a fixed/standard model)

Intro to HoCHC - definition

Simple sorts: $\sigma ::= o \mid \iota \mid \sigma \to \sigma$

Relational sorts: $\rho ::= o \mid \iota \to \rho \mid \rho \to \rho$

We consider a monotone semantics

A higher-order constrained Horn clause problem is given by a pair $\langle P, G \rangle$ in which:

- $\vdash P : \Delta$ is a constrained logic program over relational variables Δ
- $\Delta \vdash G$ is a constrained goal formula over Δ

Axiomatised over a (first-order) background theory Th (with a fixed/standard model)

$$G ::= \varphi \mid x \mid G \vee G \mid G \wedge G \mid \exists x.\, G \mid G\, N \mid G\, H \mid \lambda x.\, G$$

Intro to HoCHC - problem instance

A higher-order constrained Horn clause problem is given by a pair $\langle P, G \rangle$ in which:

- $\vdash P : \Delta$ is a constrained logic program over relational variables Δ
- $\Delta \vdash G$ is a constrained goal formula over Δ

Intro to HoCHC - problem instance

A higher-order constrained Horn clause problem is given by a pair $\langle P, G \rangle$ in which:

- $\vdash P : \Delta$ is a constrained logic program over relational variables Δ
- $\Delta \vdash G$ is a constrained goal formula over Δ

A valuation $\beta \in \mathcal{M}[\![\Delta]\!]$ is a model of P, written $\beta \vDash P$, just if $\beta = T^{\mathcal{M}}_{P:\Delta}(\beta)$.

Intro to HoCHC - problem instance

A higher-order constrained Horn clause problem is given by a pair $\langle P, G \rangle$ in which:

- $\vdash P : \Delta$ is a constrained logic program over relational variables Δ
- $\Delta \vdash G$ is a constrained goal formula over Δ

A valuation $\beta \in \mathcal{M}[\![\Delta]\!]$ is a model of P, written $\beta \vDash P$, just if $\beta = T^{\mathcal{M}}_{P:\Delta}(\beta)$.

A problem is solvable just if, for the standard model of the background theory Th, there exists a valuation β of the variables in Δ such that $\beta \vDash P$, and yet $\beta \nvDash G$.

Intro to HoCHC - problem instance

A higher-order constrained Horn clause problem is given by a pair $\langle P, G \rangle$ in which:

- $\vdash P : \Delta$ is a constrained logic program over relational variables Δ
- $\Delta \vdash G$ is a constrained goal formula over Δ

A valuation $\beta \in \mathcal{M}[\![\Delta]\!]$ is a model of P, written $\beta \vDash P$, just if $\beta = T^{\mathcal{M}}_{P:\Delta}(\beta)$.

A problem is solvable just if, for the standard model of the background theory Th, there exists a valuation β of the variables in Δ such that $\beta \vDash P$, and yet $\beta \nvDash G$.

A coinductive problem is is solvable just if, for the standard model of the background theory Th, there exists a valuation β of the variables in Δ such that $\beta \vDash P$ and $\beta \vDash G$.

Intro to HoRS

- *n*th order tree grammars
- Order 0 = regular trees, order = 1 algebraic trees, order 2 = hyperalgebraic trees, etc.

Intro to HoRS

- *n*th order tree grammars
- Order 0 = regular trees, order = 1 algebraic trees, order 2 = hyperalgebraic trees, etc.
- Programs of a simply typed λY-calculus with uninterpreted function symbols

Intro to HoRS

- nth order tree grammars
- Order 0 = regular trees, order = 1 algebraic trees, order 2 = hyperalgebraic trees, etc.
- Programs of a simply typed λY-calculus with uninterpreted function symbols
- HoRS model checking is decidable over MSO [Ong06]

Intro to HoRS - definition

- A tuple $\mathcal{G} = \langle \mathcal{N}, \Sigma, \mathcal{R}, S \rangle$

Intro to HoRS - definition

- A tuple $\mathcal{G} = \langle \mathcal{N}, \Sigma, \mathcal{R}, S \rangle$
- Assume determinism

Intro to HoRS - definition

- A tuple $\mathcal{G} = \langle \mathcal{N}, \Sigma, \mathcal{R}, S \rangle$
- Assume determinism
- Sorts $\sigma ::= \iota \mid \sigma_1 \to \sigma_2$ interpreted by

$$\mathcal{H}[\![\iota]\!] := \langle \mathcal{T}_{\Sigma_\perp}, \sqsubseteq \rangle \qquad \mathcal{H}[\![\sigma_1 \to \sigma_2]\!] := \mathcal{H}[\![\sigma_1]\!] \Rightarrow_c \mathcal{H}[\![\sigma_2]\!]$$

where $\mathcal{T}_{\Sigma_\perp}$ denotes the set of all finite and infinite trees over $\Sigma \cup \{\perp\}$,

Intro to HoRS - definition

- A tuple $\mathcal{G} = \langle \mathcal{N}, \Sigma, \mathcal{R}, S \rangle$
- Assume determinism
- Sorts $\sigma ::= \iota \mid \sigma_1 \to \sigma_2$ interpreted by

$$\mathcal{H}[\![\iota]\!] := \langle \mathcal{T}_{\Sigma_\perp}, \sqsubseteq \rangle \qquad \mathcal{H}[\![\sigma_1 \to \sigma_2]\!] := \mathcal{H}[\![\sigma_1]\!] \Rightarrow_c \mathcal{H}[\![\sigma_2]\!]$$

where $\mathcal{T}_{\Sigma_\perp}$ denotes the set of all finite and infinite trees over $\Sigma \cup \{\perp\}$, and $\sqsubseteq$ is the least partial order such that $C[\perp] \sqsubseteq C[t]$ for every tree context C and $t \in \mathcal{T}_{\Sigma_\perp}$ with $C[t] \in \mathcal{T}_{\Sigma_\perp}$.

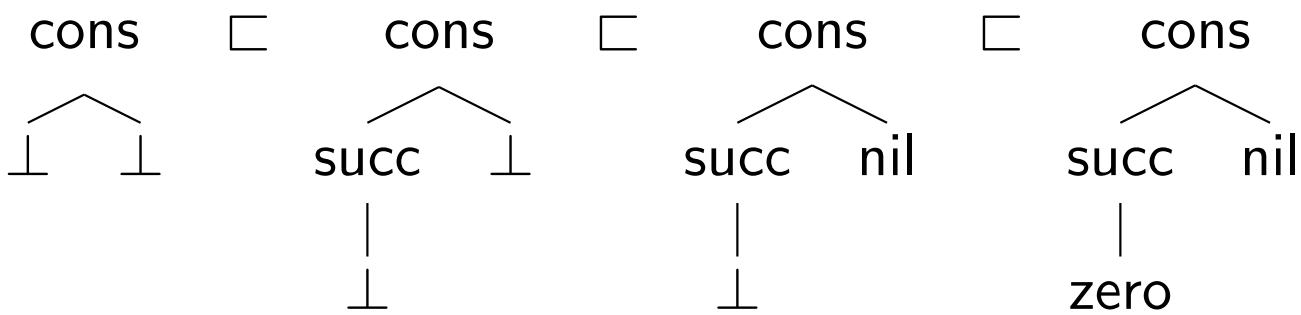
Intro to HoRS - definition

- A tuple $\mathcal{G} = \langle \mathcal{N}, \Sigma, \mathcal{R}, S \rangle$
- Assume determinism
- Sorts $\sigma ::= \iota \mid \sigma_1 \to \sigma_2$ interpreted by

$$\mathcal{H}[\![\iota]\!] := \langle \mathcal{T}_{\Sigma_\perp}, \sqsubseteq \rangle \qquad \mathcal{H}[\![\sigma_1 \to \sigma_2]\!] := \mathcal{H}[\![\sigma_1]\!] \Rightarrow_c \mathcal{H}[\![\sigma_2]\!]$$

where $\mathcal{T}_{\Sigma_\perp}$ denotes the set of all finite and infinite trees over $\Sigma \cup \{\perp\}$, and $\sqsubseteq$ is the least partial order such that $C[\perp] \sqsubseteq C[t]$ for every tree context C and $t \in \mathcal{T}_{\Sigma_\perp}$ with $C[t] \in \mathcal{T}_{\Sigma_\perp}$.

E.g.



Intro to HoRS - example

Order-2 HoRS $\mathcal{G} = \langle \{S_2, F, B\}, \{\mathsf{cons}, \mathsf{succ}, \mathsf{zero}\}, \mathcal{R}, S_2 \rangle$

$$S_2 = F\,\mathsf{succ}$$
$$F = \lambda\varphi.\,\mathsf{cons}\,(\varphi\,\mathsf{zero})\,(F\,(B\,\varphi\,\varphi))$$
$$B = \lambda\varphi\,\psi\,x.\,\varphi\,(\psi\,x)$$

S_2 $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\bot$

Intro to HoRS - example

Order-2 HoRS $\mathcal{G} = \langle \{S_2, F, B\}, \{\mathsf{cons}, \mathsf{succ}, \mathsf{zero}\}, \mathcal{R}, S_2 \rangle$

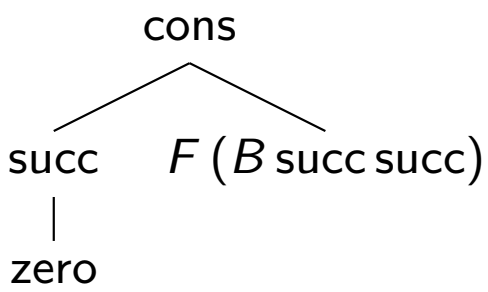
$$S_2 = F\,\mathsf{succ}$$
$$F = \lambda\varphi.\,\mathsf{cons}\,(\varphi\,\mathsf{zero})\,(F\,(B\,\varphi\,\varphi))$$
$$B = \lambda\varphi\,\psi\,x.\,\varphi\,(\psi\,x)$$

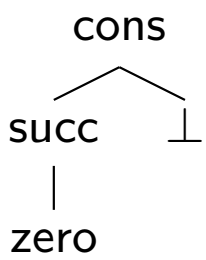
$F\,\mathsf{succ}$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\bot$

Intro to HoRS - example

Order-2 HoRS $\mathcal{G} = \langle \{S_2, F, B\}, \{\text{cons}, \text{succ}, \text{zero}\}, \mathcal{R}, S_2 \rangle$

$$S_2 = F\,\text{succ}$$
$$F = \lambda\varphi.\,\text{cons}\,(\varphi\,\text{zero})\,(F\,(B\,\varphi\,\varphi))$$
$$B = \lambda\varphi\,\psi\,x.\,\varphi\,(\psi\,x)$$

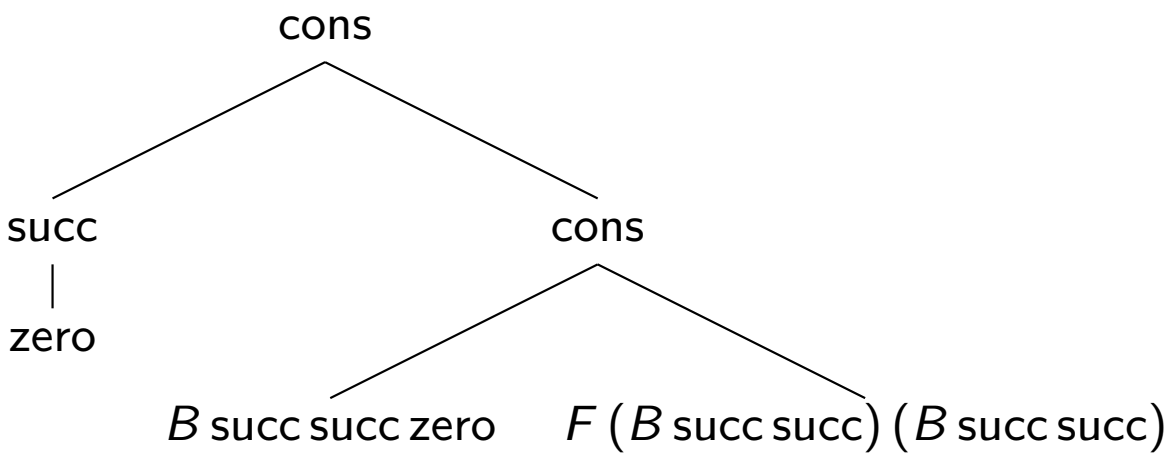


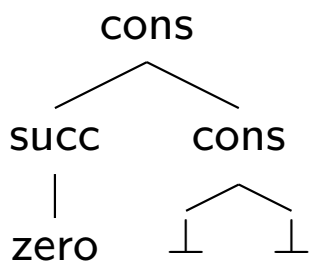


Intro to HoRS - example

Order-2 HoRS $\mathcal{G} = \langle \{S_2, F, B\}, \{\text{cons}, \text{succ}, \text{zero}\}, \mathcal{R}, S_2 \rangle$

$$S_2 = F\ \text{succ}$$
$$F = \lambda\varphi.\ \text{cons}\ (\varphi\ \text{zero})\ (F\ (B\ \varphi\ \varphi))$$
$$B = \lambda\varphi\ \psi\ x.\ \varphi\ (\psi\ x)$$

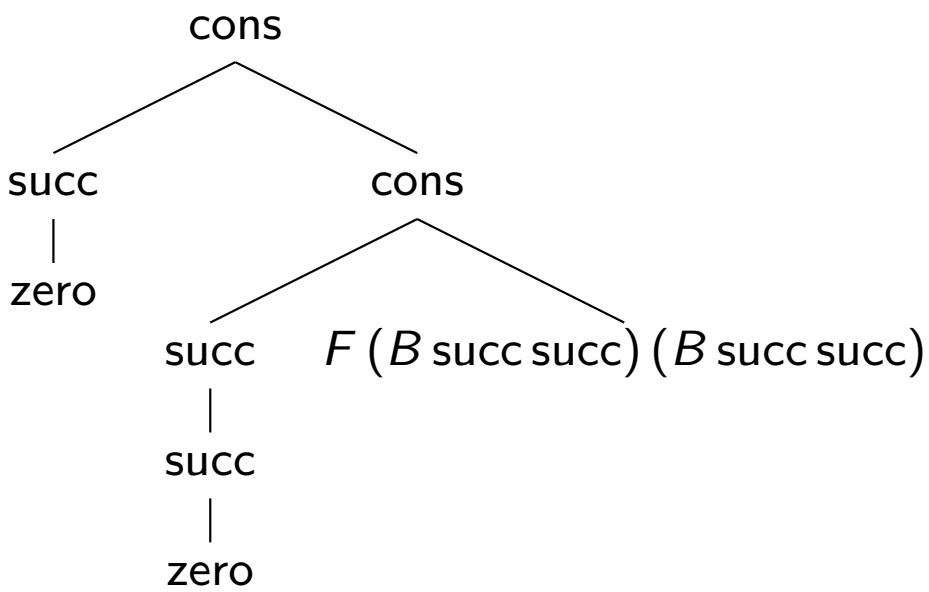


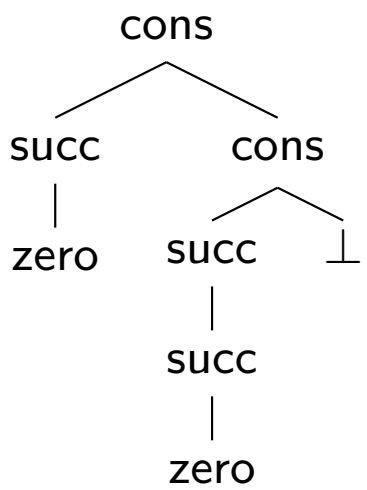


Intro to HoRS - example

Order-2 HoRS $\mathcal{G} = \langle \{S_2, F, B\}, \{\text{cons}, \text{succ}, \text{zero}\}, \mathcal{R}, S_2 \rangle$

$$S_2 = F\,\text{succ}$$
$$F = \lambda\varphi.\,\text{cons}\,(\varphi\,\text{zero})\,(F\,(B\,\varphi\,\varphi))$$
$$B = \lambda\varphi\,\psi\,x.\,\varphi\,(\psi\,x)$$

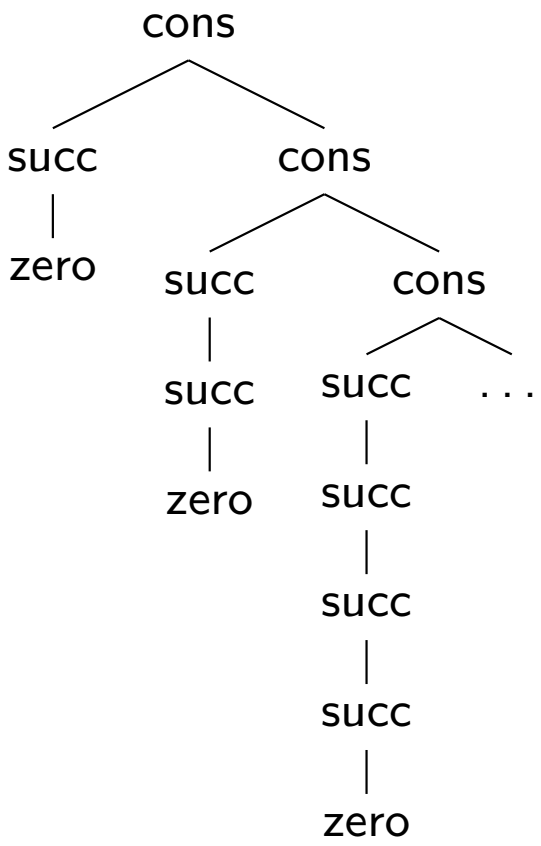




Intro to HoRS - example

Order-2 HoRS $\mathcal{G} = \langle \{S_2, F, B\}, \{\text{cons}, \text{succ}, \text{zero}\}, \mathcal{R}, S_2 \rangle$

$$S_2 = F\,\text{succ}$$
$$F = \lambda\varphi.\,\text{cons}\,(\varphi\,\text{zero})\,(F\,(B\,\varphi\,\varphi))$$
$$B = \lambda\varphi\,\psi\,x.\,\varphi\,(\psi\,x)$$



Intro to HoRS - equivalence problem

Given (deterministic) HoRS $\mathcal{G}_1$ and $\mathcal{G}_2$, does $[\![\mathcal{G}_1]\!] = [\![\mathcal{G}_2]\!]$ hold?

Open problem

Intro to HoRS - equivalence problem

Given (deterministic) HoRS $\mathcal{G}_1$ and $\mathcal{G}_2$, does $[\![\mathcal{G}_1]\!] = [\![\mathcal{G}_2]\!]$ hold?

Open problem

Recursively equivalent to the λY-calculus Böhm tree equivalence problem [CM13]

Reduction - outline

HoRS equivalence problem $\langle \mathcal{G}_1, \mathcal{G}_2 \rangle$

$\Rightarrow$

coinductive HoCHC problems $\mathcal{P}_1$ and $\mathcal{P}_0$

Reduction - outline

HoRS equivalence problem $\langle \mathcal{G}_1, \mathcal{G}_2 \rangle$

$\Rightarrow$

coinductive HoCHC problems $\mathcal{P}_1$ and $\mathcal{P}_0$ (with shared logic programs but different goal clauses)

Reduction - outline

HoRS equivalence problem $\langle \mathcal{G}_1, \mathcal{G}_2 \rangle$

$\Rightarrow$

coinductive HoCHC problems $\mathcal{P}_1$ and $\mathcal{P}_0$ (with shared logic programs but different goal clauses)

1. Eliminate non-termination from HoRS
2. Encode HoRS into HoCHC logic programs
3. Define two coinductive HoCHC instances

Reduction - stage 1: non-termination elimination

Lemma (Computability of $\bot$-free transform of HoRS)

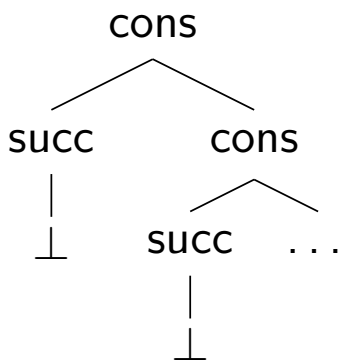
There is an algorithm that, given a HoRS $\mathcal{G}$, returns a HoRS $\mathcal{G}'$ – call it the $\bot$-free transform of $\mathcal{G}$ – that generates the $\bot$-free conversion of $[\![\mathcal{G}]\!]$.

Reduction - stage 1: non-termination elimination

Lemma (Computability of $\bot$-free transform of HoRS)

There is an algorithm that, given a HoRS $\mathcal{G}$, returns a HoRS $\mathcal{G}'$ – call it the $\bot$-free transform of $\mathcal{G}$ – that generates the $\bot$-free conversion of $[\![\mathcal{G}]\!]$.

$$S_1 = G\,\mathsf{zero}$$
$$G = \lambda x.\,\mathsf{cons}\,(\mathsf{succ}\,(H\,x))\,(G\,(\mathsf{succ}\,x))$$
$$H = \lambda x.\,H\,(\mathsf{succ}\,x)$$

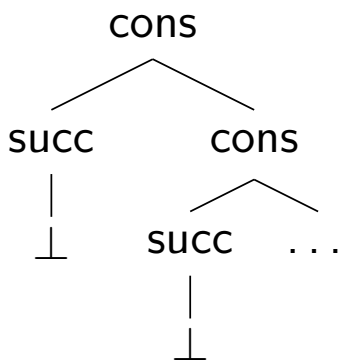


Reduction - stage 1: non-termination elimination

Lemma (Computability of $\bot$-free transform of HoRS)

There is an algorithm that, given a HoRS $\mathcal{G}$, returns a HoRS $\mathcal{G}'$ – call it the $\bot$-free transform of $\mathcal{G}$ – that generates the $\bot$-free conversion of $[\![\mathcal{G}]\!]$.

$$S_1 = b\,(G\,\mathsf{zero})$$
$$G = \lambda x.\,\mathsf{cons}\,(\mathsf{succ}\,(H\,x))\,(G\,(\mathsf{succ}\,x))$$
$$H = \lambda x.\,b\,(H\,(\mathsf{succ}\,x))$$

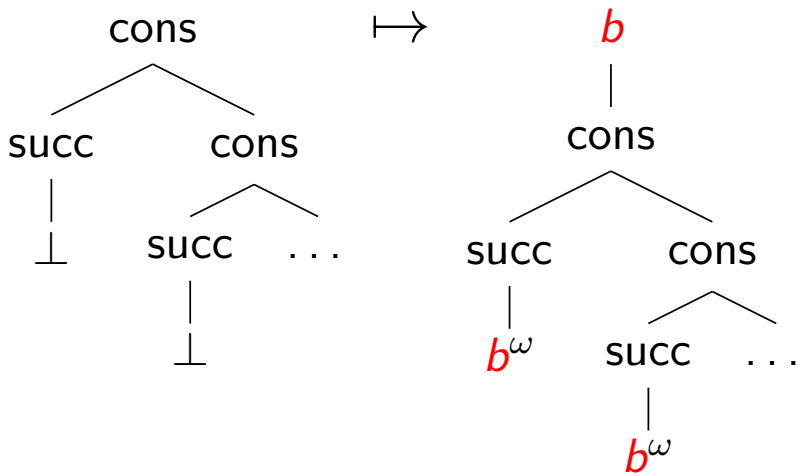


Reduction - stage 1: non-termination elimination

Lemma (Computability of $\bot$-free transform of HoRS)

There is an algorithm that, given a HoRS $\mathcal{G}$, returns a HoRS $\mathcal{G}'$ – call it the $\bot$-free transform of $\mathcal{G}$ – that generates the $\bot$-free conversion of $[\![\mathcal{G}]\!]$.

$$S_1 = b\,(G\,\mathsf{zero})$$
$$G = \lambda x.\,\mathsf{cons}\,(\mathsf{succ}\,(H\,x))\,(G\,(\mathsf{succ}\,x))$$
$$H = \lambda x.\,b\,(H\,(\mathsf{succ}\,x))$$

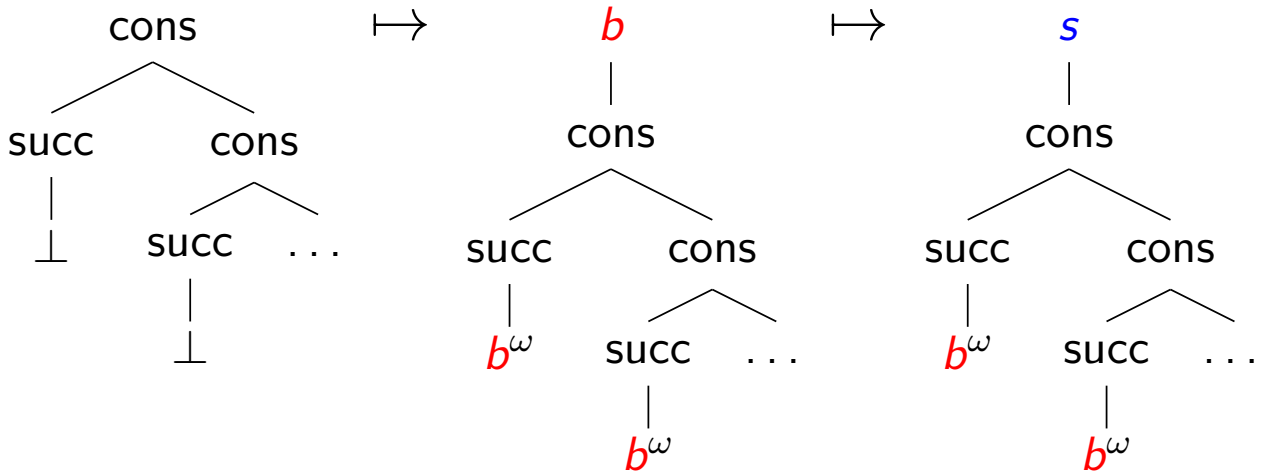


Reduction - stage 1: non-termination elimination

Lemma (Computability of $\bot$-free transform of HoRS)

There is an algorithm that, given a HoRS $\mathcal{G}$, returns a HoRS $\mathcal{G}'$ – call it the $\bot$-free transform of $\mathcal{G}$ – that generates the $\bot$-free conversion of $[\![\mathcal{G}]\!]$.

$$S_1 = s\,(G\,\mathsf{zero})$$
$$G = \lambda x.\,\mathsf{cons}\,(\mathsf{succ}\,(H\,x))\,(G\,(\mathsf{succ}\,x))$$
$$H = \lambda x.\,b\,(H\,(\mathsf{succ}\,x))$$

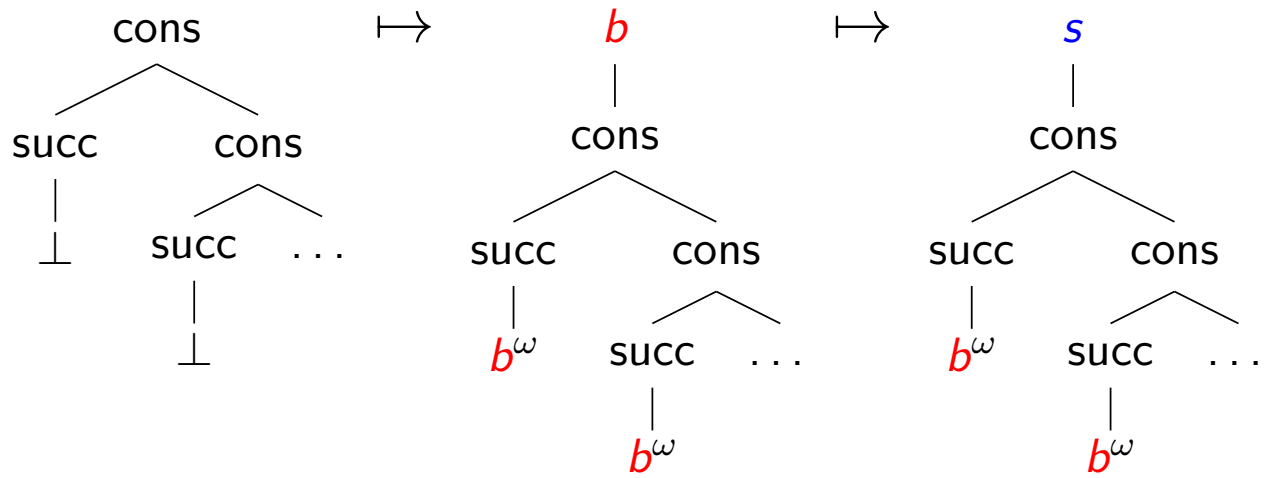


Reduction - stage 1: non-termination elimination

Lemma (Computability of $\bot$-free transform of HoRS)

There is an algorithm that, given a HoRS $\mathcal{G}$, returns a HoRS $\mathcal{G}'$ – call it the $\bot$-free transform of $\mathcal{G}$ – that generates the $\bot$-free conversion of $[\![\mathcal{G}]\!]$.

$$S_1 = s\,(G\,\mathsf{zero})$$
$$G = \lambda x.\,\mathsf{cons}\,(\mathsf{succ}\,(H\,x))\,(G\,(\mathsf{succ}\,x))$$
$$H = \lambda x.\,b\,(H\,(\mathsf{succ}\,x))$$



Theorem ([BCOS10])

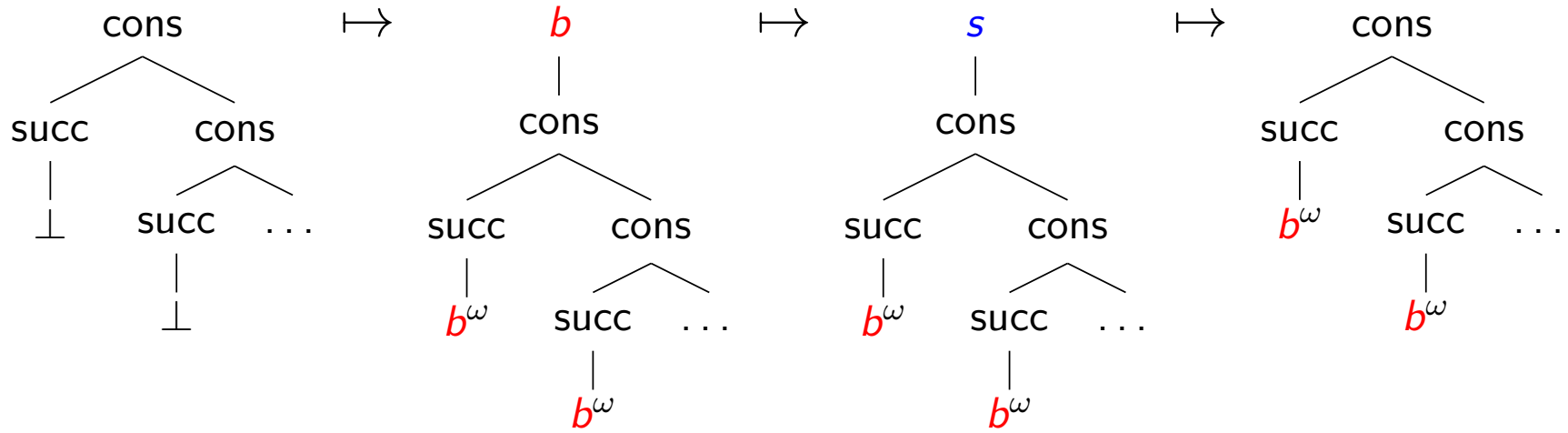
HoRS are reflective w.r.t. modal μ-calculus and MSO.

Reduction - stage 1: non-termination elimination

Lemma (Computability of $\bot$-free transform of HoRS)

There is an algorithm that, given a HoRS $\mathcal{G}$, returns a HoRS $\mathcal{G}'$ – call it the $\bot$-free transform of $\mathcal{G}$ – that generates the $\bot$-free conversion of $[\![\mathcal{G}]\!]$.

$$S_1 = I\,(G\,\mathsf{zero})$$
$$G = \lambda x.\,\mathsf{cons}\,(\mathsf{succ}\,(H\,x))\,(G\,(\mathsf{succ}\,x)) \qquad I = \lambda x.\,x$$
$$H = \lambda x.\,b\,(H\,(\mathsf{succ}\,x))$$



Theorem ([BCOS10])

HoRS are reflective w.r.t. modal μ-calculus and MSO.

Reduction - stage 2: HoRS to HoCHC encoding

Aim: given HoRS $\mathcal{G}$, define a HoCHC logic program $P_{\mathcal{G}}$ such that

Theorem

$\mathcal{M}[\![\Delta_{\mathcal{G}} \vdash R_S]\!](\mathsf{gfp}(T^{\mathcal{M}}_{P_{\mathcal{G}}:\Delta_{\mathcal{G}}}))\, t = 1$ *if and only if* $t = [\![\mathcal{G}]\!]$

Reduction - stage 2: HoRS to HoCHC encoding

Aim: given HoRS $\mathcal{G}$, define a HoCHC logic program $P_{\mathcal{G}}$ such that

Theorem

$\mathcal{M}[\![\Delta_{\mathcal{G}} \vdash R_S]\!](\mathrm{gfp}(T^{\mathcal{M}}_{P_{\mathcal{G}}:\Delta_{\mathcal{G}}}))\, t = 1$ *if and only if* $t = [\![\mathcal{G}]\!]$

Background theory: Maher's theory of finite and infinite trees [Mah88]

- Equational theory
- Complete and decidable
- See Fabian Zaiser's talk at 3pm

Reduction - stage 2: HoRS to HoCHC encoding

We map each:

- HoRS nonterminal $F \in \mathcal{N}$ to a HoCHC relational variable $R_F \in \Delta_{\mathcal{G}}$
- HoRS rewrite rule $F = \mathcal{R}(F)$ to $R_F = \ulcorner \mathcal{R}(F) \urcorner$

Reduction - stage 2: HoRS to HoCHC encoding

We map each:

- HoRS nonterminal $F \in \mathcal{N}$ to a HoCHC relational variable $R_F \in \Delta_{\mathcal{G}}$
- HoRS rewrite rule $F = \mathcal{R}(F)$ to $R_F = \ulcorner \mathcal{R}(F) \urcorner$

This gives us the constrained logic program $\vdash P_{\mathcal{G}} : \Delta_{\mathcal{G}}$, which is essentially the HoRS in continuation passing style.

Reduction - stage 2: HoRS to HoCHC encoding

We map each:

- HoRS nonterminal $F \in \mathcal{N}$ to a HoCHC relational variable $R_F \in \Delta_{\mathcal{G}}$
- HoRS rewrite rule $F = \mathcal{R}(F)$ to $R_F = \ulcorner \mathcal{R}(F) \urcorner$

This gives us the constrained logic program $\vdash P_{\mathcal{G}} : \Delta_{\mathcal{G}}$, which is essentially the HoRS in continuation passing style.

E.g. the FO rewrite rule $S_1 = I\,(G\,\mathsf{zero})$ is mapped to

$$R_{S_1} = \lambda r.\, \exists r_1\, r_2.\, R_I\, r_1\, r \wedge R_G\, r_2\, r_1 \wedge (\mathsf{zero} = r_2)$$

Reduction - stage 2: HoRS to HoCHC encoding

Our second-order HoRS

$$
\begin{aligned}
S_2 &= F\,\mathsf{succ} \\
F &= \lambda\varphi.\,\mathsf{cons}\,(\varphi\,\mathsf{zero})\,(F\,(B\,\varphi\,\varphi)) \\
B &= \lambda\varphi\,\psi\,x.\,\varphi\,(\psi\,x)
\end{aligned}
$$

maps to

Reduction - stage 2: HoRS to HoCHC encoding

Our second-order HoRS

$$
\begin{aligned}
S_2 &= F\,\mathsf{succ} \\
F &= \lambda\varphi.\,\mathsf{cons}\,(\varphi\,\mathsf{zero})\,(F\,(B\,\varphi\,\varphi)) \\
B &= \lambda\varphi\,\psi\,x.\,\varphi\,(\psi\,x)
\end{aligned}
$$

maps to

$$R_{S_2} = \lambda r.\, R_F\,(\lambda y\, r_1.\,\mathsf{succ}\, y = r_1)\, r$$

Reduction - stage 2: HoRS to HoCHC encoding

Our second-order HoRS

$$
\begin{aligned}
S_2 &= F\,\mathsf{succ} \\
F &= \lambda\varphi.\,\mathsf{cons}\,(\varphi\,\mathsf{zero})\,(F\,(B\,\varphi\,\varphi)) \\
B &= \lambda\varphi\,\psi\,x.\,\varphi\,(\psi\,x)
\end{aligned}
$$

maps to

$$
\begin{aligned}
R_{S_2} &= \lambda r.\,R_F\,(\lambda y\,r_1.\,\mathsf{succ}\,y = r_1)\,r \\
R_F &= \lambda\varphi'\,r.\,\exists r_1\,r_2\,r_3.\,(\mathsf{cons}\,r_1\,r_2 = r) \wedge \varphi'\,r_3\,r_1 \wedge (\mathsf{zero} = r_3) \wedge \\
&\qquad R_F\,(\lambda y\,r'.\,R_B\,\varphi'\,\varphi'\,y\,r')\,r_2
\end{aligned}
$$

Reduction - stage 2: HoRS to HoCHC encoding

Our second-order HoRS

$$
\begin{aligned}
S_2 &= F\,\mathsf{succ} \\
F &= \lambda\varphi.\,\mathsf{cons}\,(\varphi\,\mathsf{zero})\,(F\,(B\,\varphi\,\varphi)) \\
B &= \lambda\varphi\,\psi\,x.\,\varphi\,(\psi\,x)
\end{aligned}
$$

maps to

$$
\begin{aligned}
R_{S_2} &= \lambda r.\,R_F\,(\lambda y\,r_1.\,\mathsf{succ}\,y = r_1)\,r \\
R_F &= \lambda\varphi'\,r.\,\exists r_1\,r_2\,r_3.\,(\mathsf{cons}\,r_1\,r_2 = r) \wedge \varphi'\,r_3\,r_1 \wedge (\mathsf{zero} = r_3) \wedge \\
&\qquad R_F\,(\lambda y\,r'.\,R_B\,\varphi'\,\varphi'\,y\,r')\,r_2 \\
R_B &= \lambda\varphi'\,\psi'\,x'\,r.\,\exists r_1\,r_2.\,\varphi'\,r_1\,r \wedge \psi'\,r_2\,r_1 \wedge (x' = r_2)
\end{aligned}
$$

Reduction - stage 3: two coinductive HoCHC instances

Let $\mathcal{G}_1 = \langle \mathcal{N}_1, \Sigma, \mathcal{R}_1, S_1 \rangle$ and $\mathcal{G}_2 = \langle \mathcal{N}_2, \Sigma, \mathcal{R}_2, S_2 \rangle$ be deterministic HoRS

Reduction - stage 3: two coinductive HoCHC instances

Let $\mathcal{G}_1 = \langle \mathcal{N}_1, \Sigma, \mathcal{R}_1, S_1 \rangle$ and $\mathcal{G}_2 = \langle \mathcal{N}_2, \Sigma, \mathcal{R}_2, S_2 \rangle$ be deterministic HoRS

The HoCHC goal formulas

$$Eq_1 := \exists r_1\, r_2.\, (R_{S_1}\, r_1 \wedge R_{S_2}\, r_2) \wedge (r_1 = r_2)$$
$$Eq_0 := \exists r_1\, r_2.\, (R_{S_1}\, r_1 \wedge R_{S_2}\, r_2) \wedge (r_1 \neq r_2)$$

Reduction - stage 3: two coinductive HoCHC instances

Let $\mathcal{G}_1 = \langle \mathcal{N}_1, \Sigma, \mathcal{R}_1, S_1 \rangle$ and $\mathcal{G}_2 = \langle \mathcal{N}_2, \Sigma, \mathcal{R}_2, S_2 \rangle$ be deterministic HoRS

The HoCHC goal formulas

$$Eq_1 := \exists r_1\, r_2.\, (R_{S_1}\, r_1 \wedge R_{S_2}\, r_2) \wedge (r_1 = r_2)$$
$$Eq_0 := \exists r_1\, r_2.\, (R_{S_1}\, r_1 \wedge R_{S_2}\, r_2) \wedge (r_1 \neq r_2)$$

give rise to coinductive HoCHC problems

$$\mathcal{P}_1 := \langle P_{\mathcal{G}_1} \cup P_{\mathcal{G}_2}, Eq_1 \rangle$$
$$\mathcal{P}_0 := \langle P_{\mathcal{G}_1} \cup P_{\mathcal{G}_2}, Eq_0 \rangle$$

over the Maher theory as the constraint language and the set $\mathcal{T}_{\Sigma_\perp}$ of finite and infinite trees as the designated model.

Reduction - stage 3: two coinductive HoCHC instances

Let $\mathcal{G}_1 = \langle \mathcal{N}_1, \Sigma, \mathcal{R}_1, S_1 \rangle$ and $\mathcal{G}_2 = \langle \mathcal{N}_2, \Sigma, \mathcal{R}_2, S_2 \rangle$ be deterministic HoRS

The HoCHC goal formulas

$$Eq_1 := \exists r_1\, r_2.\, (R_{S_1}\, r_1 \wedge R_{S_2}\, r_2) \wedge (r_1 = r_2)$$
$$Eq_0 := \exists r_1\, r_2.\, (R_{S_1}\, r_1 \wedge R_{S_2}\, r_2) \wedge (r_1 \neq r_2)$$

give rise to coinductive HoCHC problems

$$\mathcal{P}_1 := \langle P_{\mathcal{G}_1} \cup P_{\mathcal{G}_2}, Eq_1 \rangle$$
$$\mathcal{P}_0 := \langle P_{\mathcal{G}_1} \cup P_{\mathcal{G}_2}, Eq_0 \rangle$$

over the Maher theory as the constraint language and the set $\mathcal{T}_{\Sigma_\perp}$ of finite and infinite trees as the designated model.

$[\![\mathcal{G}_1]\!] = [\![\mathcal{G}_2]\!]$ iff $\mathcal{P}_1$ is solvable, and $[\![\mathcal{G}_1]\!] \neq [\![\mathcal{G}_2]\!]$ iff $\mathcal{P}_0$ is solvable.

Correctness proof - key points

Embed each $\mathcal{H}[\![\sigma]\!]$ into $\mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$ using $\mathsf{i}_\sigma : \mathcal{H}[\![\sigma]\!] \to \mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$, where $\mathsf{i}_\iota(t) := \lambda s.\ t \sqsubseteq s$ for all $t \in \mathcal{H}[\![\iota]\!]$

Correctness proof - key points

Embed each $\mathcal{H}[\![\sigma]\!]$ into $\mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$ using $\mathsf{i}_\sigma : \mathcal{H}[\![\sigma]\!] \to \mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$, where $\mathsf{i}_\iota(t) := \lambda s.\ t \sqsubseteq s$ for all $t \in \mathcal{H}[\![\iota]\!]$

Lemma

For all directed sets $D \subseteq \mathcal{H}[\![\iota]\!]$, $\mathsf{i}_\iota\left(\bigsqcup D\right) = \bigsqcap\{\mathsf{i}_\iota(d) \mid d \in D\}$.

Correctness proof - key points

Embed each $\mathcal{H}[\![\sigma]\!]$ into $\mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$ using $\mathsf{i}_\sigma : \mathcal{H}[\![\sigma]\!] \to \mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$, where $\mathsf{i}_\iota(t) := \lambda s.\ t \sqsubseteq s$ for all $t \in \mathcal{H}[\![\iota]\!]$

Lemma

For all directed sets $D \subseteq \mathcal{H}[\![\iota]\!]$, $\mathsf{i}_\iota\left(\bigsqcup D\right) = \bigsqcap\{\mathsf{i}_\iota(d) \mid d \in D\}$.

Let $\alpha^n := \mathcal{H}[\![\mathcal{G}]\!]^n_{\mathcal{N}}(\bot_{\mathcal{N}})$ and $\beta^n := T^{\mathcal{M}\,n}_{P_{\mathcal{G}}:\Delta_{\mathcal{G}}}(\top_{\Delta_{\mathcal{G}}})$, so that

$$\bot_{\mathcal{N}} = \alpha^0 \sqsubseteq \alpha^1 \sqsubseteq \ldots \qquad \text{and} \qquad \top_{\Delta_{\mathcal{G}}} = \beta_0 \sqsupseteq \beta_1 \sqsupseteq \ldots$$

Correctness proof - key points

Embed each $\mathcal{H}[\![\sigma]\!]$ into $\mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$ using $\mathsf{i}_\sigma : \mathcal{H}[\![\sigma]\!] \to \mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$, where $\mathsf{i}_\iota(t) := \lambda s.\ t \sqsubseteq s$ for all $t \in \mathcal{H}[\![\iota]\!]$

Lemma

For all directed sets $D \subseteq \mathcal{H}[\![\iota]\!]$, $\mathsf{i}_\iota\left(\bigsqcup D\right) = \bigsqcap\{\mathsf{i}_\iota(d) \mid d \in D\}$.

Let $\alpha^n := \mathcal{H}[\![\mathcal{G}]\!]^n_{\mathcal{N}}(\bot_{\mathcal{N}})$ and $\beta^n := T^{\mathcal{M}\,n}_{P_{\mathcal{G}}:\Delta_{\mathcal{G}}}(\top_{\Delta_{\mathcal{G}}})$, so that

$$\bot_{\mathcal{N}} = \alpha^0 \sqsubseteq \alpha^1 \sqsubseteq \ldots \qquad \text{and} \qquad \top_{\Delta_{\mathcal{G}}} = \beta_0 \sqsupseteq \beta_1 \sqsupseteq \ldots$$

Corollary (Inclusion)

For all $n \geq 0$, $\mathcal{M}[\![\Delta_{\mathcal{G}} \vdash \ulcorner S \urcorner]\!](\beta^n) \sqsubseteq \mathsf{i}_\iota(\mathcal{H}[\![\mathcal{N} \vdash S]\!](\alpha^n))$.

Correctness proof - key points

Embed each $\mathcal{H}[\![\sigma]\!]$ into $\mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$ using $\mathsf{i}_\sigma : \mathcal{H}[\![\sigma]\!] \to \mathcal{M}[\![\mathsf{Rel}^+(\sigma)]\!]$, where $\mathsf{i}_\iota(t) := \lambda s.\ t \sqsubseteq s$ for all $t \in \mathcal{H}[\![\iota]\!]$

Lemma

For all directed sets $D \subseteq \mathcal{H}[\![\iota]\!]$, $\mathsf{i}_\iota\left(\bigsqcup D\right) = \bigsqcap\{\mathsf{i}_\iota(d) \mid d \in D\}$.

Let $\alpha^n := \mathcal{H}[\![\mathcal{G}]\!]^n_{\mathcal{N}}(\bot_{\mathcal{N}})$ and $\beta^n := T^{\mathcal{M}\,n}_{P_{\mathcal{G}}:\Delta_{\mathcal{G}}}(\top_{\Delta_{\mathcal{G}}})$, so that

$$\bot_{\mathcal{N}} = \alpha^0 \sqsubseteq \alpha^1 \sqsubseteq \ldots \qquad \text{and} \qquad \top_{\Delta_{\mathcal{G}}} = \beta_0 \sqsupseteq \beta_1 \sqsupseteq \ldots$$

Corollary (Inclusion)

For all $n \geq 0$, $\mathcal{M}[\![\Delta_{\mathcal{G}} \vdash \ulcorner S \urcorner]\!](\beta^n) \sqsubseteq \mathsf{i}_\iota(\mathcal{H}[\![\mathcal{N} \vdash S]\!](\alpha^n))$.

Lemma (Nonemptiness)

There exists a $\bot$-free tree $t \in \mathcal{M}[\![\iota]\!]$ such that $\mathcal{M}[\![\Delta_{\mathcal{G}} \vdash \ulcorner S \urcorner]\!]\left(\bigsqcap_n \beta^n\right)\ t$.

Contributions

- Coinductive HoCHC framework
- Eliminating non-termination from HoRS
- Encoding from HoRS into HoCHC logic program
- Reduction of HoRS equivalence to semi-decidability of co-HoCHC
- (and therefore also the λY-calculus Böhm tree equivalence problem)

Further directions

- Semi-decidability of coinductive HoCHC over Maher's theory of trees

Further directions

- Semi-decidability of coinductive HoCHC over Maher's theory of trees
- (specifically: of the image of the the HoRS-to-HoCHC encoding)

References

Christopher H. Broadbent, Arnaud Carayol, C.-H. Luke Ong, and Olivier Serre, Recursion schemes and logical reflection, Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom, IEEE Computer Society, 2010, pp. 120–129.

Pierre Clairambault and Andrzej S. Murawski, Böhm Trees as Higher-Order Recursive Schemes, IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2013) (Dagstuhl, Germany) (Anil Seth and Nisheeth K. Vishnoi, eds.), Leibniz International Proceedings in Informatics (LIPIcs), vol. 24, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2013, pp. 91–102.

Toby Cathcart Burn, C.-H. Luke Ong, and Steven J. Ramsay, Higher-order constrained Horn clauses for verification, PACMPL **2** (2018), no. POPL, 11:1–11:28.

Michael J. Maher, Complete axiomatizations of the algebras of finite, rational and infinite trees, LICS, 1988, pp. 348–357.

C.-H. Luke Ong, On model-checking trees generated by higher-order recursion schemes, 21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings, 2006, pp. 81–90.

C.-H. Luke Ong and Dominik Wagner, HoCHC: A refutationally complete and semantically invariant system of higher-order logic modulo theories, 2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2019, pp. 1–14.

Long Pham, Steven J. Ramsay, and C.-H. Luke Ong, Defunctionalization of higher-order constrained Horn clauses, CoRR **abs/1810.03598** (2018).

Why do we need coinduction?

HoCHC individuals (datatypes) are unordered in $\mathcal{M}[\![\iota]\!]$, so that any trees t_1, t_2 are incomparable

Why do we need coinduction?

HoCHC individuals (datatypes) are unordered in $\mathcal{M}[\![\iota]\!]$, so that any trees t_1, t_2 are incomparable

Let $\chi_t \in \mathcal{M}[\![\iota \to o]\!]$ be the characteristic function of some tree $t \in \mathcal{M}[\![\iota]\!]$.

Why do we need coinduction?

HoCHC individuals (datatypes) are unordered in $\mathcal{M}[\![\iota]\!]$, so that any trees t_1, t_2 are incomparable

Let $\chi_t \in \mathcal{M}[\![\iota \to o]\!]$ be the characteristic function of some tree $t \in \mathcal{M}[\![\iota]\!]$.

This implies that:

Lemma

If $f \in \mathcal{M}[\![\iota \to o]\!]$ and $f \sqsubset \chi_t$, then f is the least (constant false) element of $\mathcal{M}[\![\iota \to o]\!]$.

Why do we need coinduction?

HoCHC individuals (datatypes) are unordered in $\mathcal{M}[\![\iota]\!]$, so that any trees t_1, t_2 are incomparable

Let $\chi_t \in \mathcal{M}[\![\iota \to o]\!]$ be the characteristic function of some tree $t \in \mathcal{M}[\![\iota]\!]$.

This implies that:

Lemma

If $f \in \mathcal{M}[\![\iota \to o]\!]$ and $f \sqsubset \chi_t$, then f is the least (constant false) element of $\mathcal{M}[\![\iota \to o]\!]$.

Thus, the ascending Kleene chain $f = \alpha^0 \sqsubseteq \alpha^1 \sqsubseteq \alpha^2 \sqsubseteq \ldots$ must reach χ_t in α^1 or not at all (if α^0 is a fixpoint).

Why do we need coinduction?

HoCHC individuals (datatypes) are unordered in $\mathcal{M}[\![\iota]\!]$, so that any trees t_1, t_2 are incomparable

Let $\chi_t \in \mathcal{M}[\![\iota \to o]\!]$ be the characteristic function of some tree $t \in \mathcal{M}[\![\iota]\!]$.

This implies that:

Lemma

If $f \in \mathcal{M}[\![\iota \to o]\!]$ and $f \sqsubset \chi_t$, then f is the least (constant false) element of $\mathcal{M}[\![\iota \to o]\!]$.

Thus, the ascending Kleene chain $f = \alpha^0 \sqsubseteq \alpha^1 \sqsubseteq \alpha^2 \sqsubseteq \ldots$ must reach χ_t in α^1 or not at all (if α^0 is a fixpoint).

Clearly, the chain cannot reach χ_t for an infinite t after a single step.

Why do we need to eliminate $\perp$ from HoRS?

Correctness requires us to distinguish ‘finished’ from ‘unfinished’ trees

Why do we need to eliminate $\bot$ from HoRS?

Correctness requires us to distinguish 'finished' from 'unfinished' trees

This is due to the embedding $\mathsf{i}_\iota(t) := \lambda s.\ t \sqsubseteq s$ for all $t \in \mathcal{H}[\![\iota]\!]$

Why do we need to eliminate $\bot$ from HoRS?

Correctness requires us to distinguish 'finished' from 'unfinished' trees

This is due to the embedding $\mathsf{i}_\iota(t) := \lambda s.\ t \sqsubseteq s$ for all $t \in \mathcal{H}[\![\iota]\!]$

If t is $\bot$-free, then $\mathsf{i}_\iota(t) = \lambda s.\,(t = s)$ is the characteristic function of t.

Why do we need to eliminate $\bot$ from HoRS?

Correctness requires us to distinguish 'finished' from 'unfinished' trees

This is due to the embedding $\mathsf{i}_\iota(t) := \lambda s.\ t \sqsubseteq s$ for all $t \in \mathcal{H}[\![\iota]\!]$

If t is $\bot$-free, then $\mathsf{i}_\iota(t) = \lambda s.\,(t = s)$ is the characteristic function of t.

Essentially, at the end of the ascending Kleene chain of HoRS semantics, our inclusion in i_ι becomes equality.

Why do we need to eliminate $\bot$ from HoRS?

Correctness requires us to distinguish 'finished' from 'unfinished' trees

This is due to the embedding $\mathsf{i}_\iota(t) := \lambda s.\ t \sqsubseteq s$ for all $t \in \mathcal{H}[\![\iota]\!]$

If t is $\bot$-free, then $\mathsf{i}_\iota(t) = \lambda s.\,(t = s)$ is the characteristic function of t.

Essentially, at the end of the ascending Kleene chain of HoRS semantics, our inclusion in i_ι becomes equality.

This is key to our main theorem:

Theorem

$\mathcal{M}[\![\Delta_\mathcal{G} \vdash R_S]\!](\mathsf{gfp}(T^{\mathcal{M}}_{P_\mathcal{G}:\Delta_\mathcal{G}}))\, t = 1$ *if and only if* $t = [\![\mathcal{G}]\!]$

Intro to HoCHC - syntax

$$\text{(GConstr)}\ \frac{}{\Delta \vdash \varphi : o}\ \Delta \vdash \varphi : o \in Fm \qquad \text{(GVar)}\ \frac{}{\Delta_1, x : \rho, \Delta_2 \vdash x : \rho}$$

$$\text{(GCst)}\ \frac{\Delta \vdash G : o \qquad \Delta \vdash H : o}{\Delta \vdash G * H : o}\ * \in \{\wedge, \vee\}$$

$$\text{(GEx)}\ \frac{\Delta, x : \sigma \vdash G : o}{\Delta \vdash \exists x{:}\sigma.G : o}\ \sigma = \iota \text{ or } \rho$$

$$\text{(GAppI)}\ \frac{\Delta \vdash G : \iota \to \rho}{\Delta \vdash G\ N : \rho}\ \Delta \vdash N : \iota \in Tm$$

$$\text{(GAppR)}\ \frac{\Delta \vdash G : \rho_1 \to \rho_2 \qquad \Delta \vdash H : \rho_1}{\Delta \vdash G\ H : \rho_2}$$

$$\text{(GAbs)}\ \frac{\Delta, x : \sigma \vdash G : \rho}{\Delta \vdash \lambda x.\, G : \sigma \to \rho}\ x \notin \mathrm{dom}(\Delta)$$

Intro to HoCHC - semantics

We consider a monotone semantics with sort frame:

$$\mathcal{M}[\![o]\!] := \mathbb{B} \qquad \mathcal{M}[\![\iota]\!] := A_\iota \qquad \mathcal{M}[\![\sigma_1 \to \sigma_2]\!] := \mathcal{M}[\![\sigma_1]\!] \Rightarrow_m \mathcal{M}[\![\sigma_2]\!]$$

Goal terms are interpreted as follows:

$$\begin{aligned}
\mathcal{M}[\![\Delta \vdash \varphi : o]\!](\beta) &:= \mathit{Th}[\![\varphi]\!](\beta) \\
\mathcal{M}[\![\Delta_1, x : \rho, \Delta_2 \vdash x : \rho]\!](\beta) &:= \beta(x) \\
\mathcal{M}[\![\Delta \vdash G \wedge H : o]\!](\beta) &:= \min\{\mathcal{M}[\![\Delta \vdash G : o]\!](\beta), \mathcal{M}[\![\Delta \vdash H : o]\!](\beta)\} \\
\mathcal{M}[\![\Delta \vdash G \vee H : o]\!](\beta) &:= \max\{\mathcal{M}[\![\Delta \vdash G : o]\!](\beta), \mathcal{M}[\![\Delta \vdash H : o]\!](\beta)\} \\
\mathcal{M}[\![\Delta \vdash \exists x : \sigma.\, G : o]\!](\beta) &:= \max\{\mathcal{M}[\![\Delta, x : \sigma \vdash G : o]\!](\beta[x \mapsto x']) \mid x' \in \mathcal{M}[\![\sigma]\!]\} \\
\mathcal{M}[\![\Delta \vdash \lambda x : \sigma.\, G : \sigma \to \rho]\!](\beta) &:= \lambda x' \in \mathcal{M}[\![\sigma]\!].\, \mathcal{M}[\![\Delta, x : \sigma \vdash G : \rho]\!](\beta[x \mapsto x']) \\
\mathcal{M}[\![\Delta \vdash G\, H : \rho_2]\!](\beta) &:= \mathcal{M}[\![\Delta \vdash G : \rho_1 \to \rho_2]\!](\beta)(\mathcal{M}[\![\Delta \vdash H : \rho_1]\!][\![H]\!](\beta)) \\
\mathcal{M}[\![\Delta \vdash G\, N : \rho]\!](\beta) &:= \mathcal{M}[\![\Delta \vdash G : \iota \to \rho]\!](\beta)(\mathit{Th}[\![N]\!](\beta))
\end{aligned}$$

Reduction - stage 2: HoRS to HoCHC encoding

We map each HoRS nonterminal $F : \sigma \in \mathcal{N}$ to HoCHC relational variable $R_F : \mathsf{Rel}^+(\sigma)$,

Reduction - stage 2: HoRS to HoCHC encoding

We map each HoRS nonterminal $F : \sigma \in \mathcal{N}$ to HoCHC relational variable $R_F : \mathsf{Rel}^+(\sigma)$, where

$$\begin{aligned} \mathsf{Rel}^-(\iota) &:= \iota \\ \mathsf{Rel}^+(\iota) &:= \iota \to o \\ \mathsf{Rel}^+(\sigma_1 \to \sigma_2) := \mathsf{Rel}^-(\sigma_1 \to \sigma_2) &:= \mathsf{Rel}^-(\sigma_1) \to \mathsf{Rel}^+(\sigma_2), \end{aligned}$$

Reduction - stage 2: HoRS to HoCHC encoding

We map each HoRS nonterminal $F : \sigma \in \mathcal{N}$ to HoCHC relational variable $R_F : \mathsf{Rel}^+(\sigma)$, where

$$\begin{aligned} \mathsf{Rel}^-(\iota) &:= \iota \\ \mathsf{Rel}^+(\iota) &:= \iota \to o \\ \mathsf{Rel}^+(\sigma_1 \to \sigma_2) := \mathsf{Rel}^-(\sigma_1 \to \sigma_2) &:= \mathsf{Rel}^-(\sigma_1) \to \mathsf{Rel}^+(\sigma_2), \end{aligned}$$

This gives us the constrained logic program $\vdash P_{\mathcal{G}} : \Delta_{\mathcal{G}}$ defined by:

$$\Delta_{\mathcal{G}} := \left\{ R_F : \mathsf{Rel}^+(\sigma) \mid F : \sigma \in \mathcal{N} \right\}$$

$$P_{\mathcal{G}} := \left\{ R_F : \mathsf{Rel}^+(\sigma) = \ulcorner \mathcal{R}(F) \urcorner \mid F : \sigma \in \mathcal{N} \right\}$$