HoCHC: a Refutationally Complete and Semantically Invariant System of Higher-order Logic Modulo Theories

Luke Ong **Dominik Wagner**

UNIVERSITY OF OXFORD

HCVS 2019

7th April 2019

"Constrained Horn Clauses provide a suitable basis for automatic program verification"

[Bjørner et al., 2015]

"Constrained Horn Clauses provide a suitable basis for automatic program verification"

[Bjørner et al., 2015]

- separation of concerns
- good *algorithmic* properties: semi-decidable, highly efficient solvers

1st-order

"Constrained Horn Clauses provide a suitable basis for automatic program verification"

imperative

[Bjørner et al., 2015]

- separation of concerns
- good *algorithmic* properties: semi-decidable, highly efficient solvers

1st-order

"Constrained Horn Clauses provide a suitable basis for automatic program verification"

imperative

[Bjørner et al., 2015]

- separation of concerns
- good *algorithmic* properties: semi-decidable, highly efficient solvers

[Cathcart Burn, Ong & Ramsay; POPL'18]:
extend approach to higher-orders

```
let       add  x y   = x + y
let rec iter f s n = if n <= 0 then s
                     else f n (iter f s (n-1))
in λn. assert (n >= 1 -> (iter add n n > n+n))
```

```
let     add  x y   = x + y
let rec iter f s n = if n <= 0 then s
                     else f n (iter f s (n−1))
in λn. assert (n >= 1 −> (iter add n n > n+n))
```

```
let     add   x y   = x + y
let rec iter f s n = if n <= 0 then s
                     else f n (iter f s (n-1))
in λn. assert (n >= 1 -> (iter add n n > n+n))
```

⬇ (over-)approximate graph of functions ⬇

```
let     add  x y   = x + y
let rec iter f s n = if n <= 0 then s
                     else f n (iter f s (n−1))
in λn. assert (n >= 1 −> (iter add n n > n+n))
```

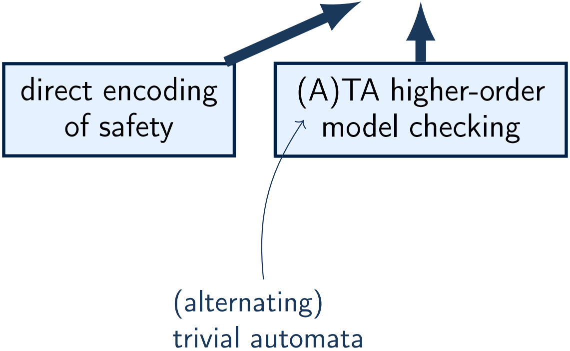
(over-)approximate graph of functions

$$\forall_{x,y,z.} \big(z = x + y \to \mathrm{Add}\ x\, y\, z\big)$$
$$\forall_{f,s,n,x.} \big(n \leq 0 \wedge s = x \to \mathrm{Iter}\ f\, s\, n\, x\big)$$
$$\forall_{f,s,n,x.} \big(n > 0 \wedge \exists y.\,(\mathrm{Iter}\ f\, s\,(n-1)\, y \wedge f\, n\, y\, x) \to \mathrm{Iter}\ f\, s\, n\, x\big)$$
$$\forall_{n,x.} \big(n \geq 1 \wedge \mathrm{Iter}\ \mathrm{Add}\ n\, n\, x \ \to\ x > n + n\big)$$

HoCHC for verification

direct encoding of safety

HoCHC for verification



HoCHC for verification

direct encoding of safety

(A)TA higher-order model checking

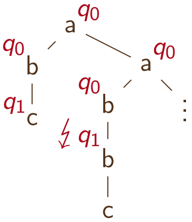
$\mathcal{L}(\mathcal{A}) \ni [\![\mathcal{G}]\!]$

(alternating) trivial automata

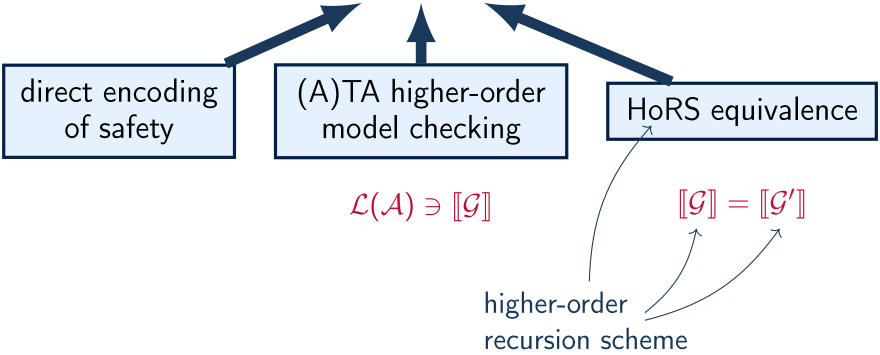
higher-order recursion scheme

$\mathcal{G}$ defined by

$$S = F\, b\, c$$
$$F\, f\, x = a\,(f\, x)\,(F\, f\,(f\, x))$$

$[\![\mathcal{G}]\!] =$ 

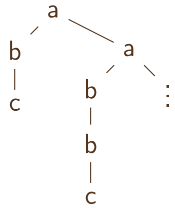
HoCHC for verification



$\mathcal{G}$ defined by

$$S = F\, b\, c$$
$$F\, f\, x = a\,(f\, x)\,(F\, f\,(f\, x))$$

$[\![\mathcal{G}]\!] =$



Is higher-order (Horn) logic modulo theories a sensible algorithmic approach to verification?

Is higher-order (Horn) logic modulo theories a sensible algorithmic approach to verification?

Is it well-founded?

	1st-order logic
complete proof systems	✓
semi-decidable	✓

	1st-order logic	higher-order logic
		standard
complete proof systems	✓	✗
semi-decidable	✓	✗

	1st-order logic	higher-order logic	
		standard	Henkin
complete proof systems	✓	✗	✓
semi-decidable	✓	✗	✓
1st-order translation	–	✗	✓

	1st-order logic	higher-order logic	
		standard	Henkin
complete proof systems	✓	✗	✓
semi-decidable	✓	✗	✓
1st-order translation	–	✗	✓
intuitive	✓	✓	✗

	1st-order logic	~~higher-order logic~~ *HoCHC*	
		standard	Henkin
complete proof systems	✓	~~✗~~ ✓	✓
semi-decidable	✓	~~✗~~ ✓	✓
1st-order translation	–	~~✗~~ ✓	✓
intuitive	✓	✓	✗

	1st-order logic	~~higher-order logic~~ *HoCHC* standard ⇔	Henkin
complete proof systems	✓	✗ ✓	✓
semi-decidable	✓	✗ ✓	✓
1st-order translation	–	✗ ✓	✓
intuitive	✓	✓	✗

Contributions

- A *simple* resolution proof system for HoCHC
  - Completeness even for *standard* semantics
  - HoCHC is *semi-decidable* and compact
- Semantic invariance

Contributions

- A *simple* resolution proof system for HoCHC
  - Completeness even for *standard* semantics
  - HoCHC is *semi-decidable* and compact
- Semantic invariance
- Canonical model property
- 1-st order translation (complete for *standard* semantics)
- *Decidable* fragments

Paper accompanying this talk: [Ong & Wagner, LICS'19]

Contributions

- A *simple* resolution proof system for HoCHC
  - Completeness even for *standard* semantics
  - HoCHC is *semi-decidable* and compact
- Semantic invariance
- Canonical model property
- 1-st order translation (complete for *standard* semantics)
- *Decidable* fragments

Paper accompanying this talk: [Ong & Wagner, LICS'19]

This talk:

- Resolution proof system and its completeness
- Canonical model property
- Semantic invariance

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

$$\neg(z = x + y) \lor \text{Add } x\, y\, z$$
$$\neg(n \leq 0) \lor \neg(s = x) \lor \text{Iter } f\, s\, n\, x$$
$$\neg(n > 0) \lor \neg\, \text{Iter } f\, s\, (n - 1)\, y \lor \neg(f\, n\, y\, x) \lor \text{Iter } f\, s\, n\, x$$
$$\neg(n \geq 1) \lor \neg\, \text{Iter Add } n\, n\, x \lor \neg(x \leq n + n)$$

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

background theory → Σ relational extension → Σ'

$$\neg(z = x + y) \vee \mathsf{Add}\ x\, y\, z$$

$$\neg(n \leq 0) \vee \neg(s = x) \vee \mathsf{Iter}\ f\, s\, n\, x$$

$$\neg(n > 0) \vee \neg\,\mathsf{Iter}\ f\, s\, (n-1)\, y \vee \neg(f\ n\, y\, x) \vee \mathsf{Iter}\ f\, s\, n\, x$$

$$\neg(n \geq 1) \vee \neg\,\mathsf{Iter}\ \mathsf{Add}\ n\, n\, x \vee \neg(x \leq n + n)$$

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

background theory   relational extension

$$\neg(z = x + y) \lor \mathrm{Add}\ x\,y\,z$$

$$\neg(n \leq 0) \lor \neg(s = x) \lor \mathrm{Iter}\ f\,s\,n\,x$$

$$\neg(n > 0) \lor \neg\,\mathrm{Iter}\ f\,s\,(n-1)\,y \lor \neg(f\ n\,y\,x) \lor \mathrm{Iter}\ f\,s\,n\,x$$

$$\neg(n \geq 1) \lor \neg\,\mathrm{Iter}\ \mathrm{Add}\ n\,n\,x \lor \neg(x \leq n + n)$$

- only *relational* higher-order types

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

definite clauses

$$\neg(z = x + y) \lor \mathrm{Add}\ x\, y\, z$$

$$\neg(n \leq 0) \lor \neg(s = x) \lor \mathrm{Iter}\ f\, s\, n\, x$$

$$\neg(n > 0) \lor \neg\, \mathrm{Iter}\ f\, s\, (n - 1)\, y \lor \neg(f\ n\, y\, x) \lor \mathrm{Iter}\ f\, s\, n\, x$$

$$\neg(n \geq 1) \lor \neg\, \mathrm{Iter}\ \mathrm{Add}\ n\, n\, x \lor \neg(x \leq n + n)$$

goal clause

- only *relational* higher-order types

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

distinct variables

$$\neg(z = x + y) \lor \mathrm{Add}\ x\,y\,z$$

$$\neg(n \leq 0) \lor \neg(s = x) \lor \mathrm{Iter}\ f\,s\,n\,x$$

$$\neg(n > 0) \lor \neg\,\mathrm{Iter}\ f\,s\,(n-1)\,y \lor \neg(f\ n\,y\,x) \lor \mathrm{Iter}\ f\,s\,n\,x$$

$$\neg(n \geq 1) \lor \neg\,\mathrm{Iter}\ \mathrm{Add}\ n\,n\,x \lor \neg(x \leq n + n)$$

- only *relational* higher-order types
- positive literals are *definitional*

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

distinct variables

$$\neg(z = x + y) \vee \mathsf{Add}\ x\, y\, z$$

$$\neg(n \leq 0) \vee \neg(s = x) \vee \mathsf{Iter}\ f\, s\, n\, x$$

$$\neg(n > 0) \vee \neg\,\mathsf{Iter}\ f\, s\, (n-1)\, y \vee \neg(f\ n\, y\, x) \vee \mathsf{Iter}\ f\, s\, n\, x$$

$$\neg(n \geq 1) \vee \neg\,\mathsf{Iter}\ \mathsf{Add}\ n\, n\, x \vee \neg(x \leq n + n)$$

- only *relational* higher-order types
- positive literals are *definitional*
- no logical symbols in atoms: ~~$R\,\neg$~~
- in paper: + λ-abstractions

Standard Semantics

$\mathcal{A}$: fixed model of the background theory

Standard Semantics

$\mathcal{A}$: fixed model of the background theory

standard interpretation $\mathcal{S}$ of types: *full* function space

$$\mathcal{S}[\![\iota]\!] \coloneqq \mathsf{dom}(\mathcal{A}) \quad \mathcal{S}[\![o]\!] \coloneqq \mathbb{B} \quad \mathcal{S}[\![\tau \to \sigma]\!] \coloneqq [\mathcal{S}[\![\tau]\!] \to \mathcal{S}[\![\sigma]\!]]$$

Standard Semantics

$\mathcal{A}$: fixed model of the background theory

standard interpretation $\mathcal{S}$ of types: *full* function space

$$\mathcal{S}[\![\iota]\!] := \mathsf{dom}(\mathcal{A}) \quad \mathcal{S}[\![o]\!] := \mathbb{B} \quad \mathcal{S}[\![\tau \to \sigma]\!] := [\mathcal{S}[\![\tau]\!] \to \mathcal{S}[\![\sigma]\!]]$$

Structures $\mathcal{B}$, valuations α and denotations $\mathcal{B}[\![M]\!](\alpha)$ as usual *(w.r.t. $\mathcal{S}[\![-]\!]$!)*

e.g. $$\mathcal{B}[\![M_1\, M_2]\!](\alpha) := \mathcal{B}[\![M_1]\!](\alpha)(\mathcal{B}[\![M_2]\!](\alpha))$$

HoCHC Satisfiability Problem

$\mathcal{A}$: fixed model (over Σ) of the background theory

S: set of HoCHCs

Definition (Satisfiability)

S is *$\mathcal{A}$-satisfiable* if there exists a Σ'-structure $\mathcal{B}$ s.t.

1. $\mathcal{B}$ agrees with $\mathcal{A}$ on Σ (background theory) ,

HoCHC Satisfiability Problem

$\mathcal{A}$: fixed model (over Σ) of the background theory

S: set of HoCHCs

Definition (Satisfiability)

S is *$\mathcal{A}$-satisfiable* if there exists a Σ'-structure $\mathcal{B}$ s.t.

1. $\mathcal{B}$ agrees with $\mathcal{A}$ on Σ (background theory) ,
2. $\mathcal{B}, \alpha \models C$ for each $C \in S$ and valuation α.

Proof System

Resolution $$\frac{\neg R\,\overline{M} \vee G \qquad\qquad G' \vee R\,\overline{x}}{G \vee \big(G'[\overline{M}/\overline{x}]\big)}$$

Proof System

Resolution
$$\frac{\neg R\,\overline{M} \vee G \qquad G' \vee R\,\overline{x}}{G \vee \big(G'[\overline{M}/\overline{x}]\big)}$$

variables — background atoms

Constraint Refutation
$$\frac{\neg\, x_1\ \overline{M}_1 \vee \cdots \vee \neg\, x_m\ \overline{M}_m \vee \neg\, \varphi_1 \vee \cdots \vee \neg\, \varphi_n}{\bot}$$

provided that there exists a valuation α such that $\mathcal{A}, \alpha \models \varphi_1 \wedge \cdots \wedge \varphi_n$

Proof System

Resolution

$$\frac{\neg R\,\overline{M} \vee G \qquad G' \vee R\,\overline{x}}{G \vee \big(G'[\overline{M}/\overline{x}]\big)}$$

Constraint Refutation

variables

background atoms

$$\frac{\neg\, x_1\ \overline{M}_1 \vee \cdots \vee \neg\, x_m\ \overline{M}_m \vee \neg\, \varphi_1\ \vee \cdots \vee \neg\, \varphi_n}{\bot}$$

provided that there exists a valuation α such that $\mathcal{A}, \alpha \models \varphi_1 \wedge \cdots \wedge \varphi_n$

$\neg x\, M \vee \neg\varphi$

Proof System

Resolution

$$\frac{\neg R\,\overline{M} \lor G \qquad G' \lor R\,\overline{x}}{G \lor \left(G'[\overline{M}/\overline{x}]\right)}$$

variables — background atoms

Constraint Refutation

$$\frac{\neg\, x_1\ \overline{M}_1 \lor \cdots \lor \neg\, x_m\ \overline{M}_m \lor \neg\, \varphi_1\ \lor \cdots \lor \neg\, \varphi_n}{\bot}$$

provided that there exists a valuation α such that $\mathcal{A}, \alpha \models \varphi_1 \land \cdots \land \varphi_n$

$\neg x\, M \lor \neg\varphi \models \neg\varphi$

Proof System

Resolution

$$\frac{\neg R\,\overline{M} \vee G \qquad G' \vee R\,\overline{x}}{G \vee \big(G'[\overline{M}/\overline{x}]\big)}$$

Constraint Refutation

variables — background atoms

$$\frac{\neg\, x_1\ \overline{M}_1 \vee \cdots \vee \neg\, x_m\ \overline{M}_m \vee \neg\, \varphi_1\ \vee \cdots \vee \neg\, \varphi_n}{\bot}$$

provided that there exists a valuation α such that $\mathcal{A}, \alpha \models \varphi_1 \wedge \cdots \wedge \varphi_n$

$\neg x\, M \vee \neg\varphi \models \neg\varphi$

(modulo renaming of variables)
(+ rule for β-reduction in paper)

✓

soundness

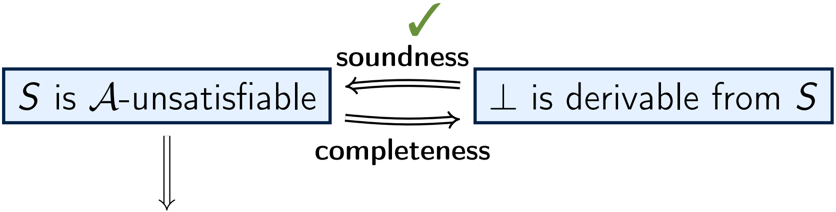
S is $\mathcal{A}$-unsatisfiable $\Longleftarrow$ $\bot$ is derivable from S

✓ soundness

S is $\mathcal{A}$-unsatisfiable $\Longleftarrow$ $\bot$ is derivable from S

S is $\mathcal{A}$-unsatisfiable $\Longrightarrow$ $\bot$ is derivable from S

completeness



1. S is $\mathcal{A}$-*continuous*-unsatisfiable

✓ soundness

S is $\mathcal{A}$-unsatisfiable ⇐⇒ $\bot$ is derivable from S

completeness

⇓

1. S is $\mathcal{A}$-*continuous*-unsatisfiable

⇓

2. $\exists G \in S$ and $n \in \omega$ s.t. $\mathcal{A}^{\mathcal{C}}_n \not\models G$

↑ n-th stage of continuous *canonical* structure

✓
soundness

S is $\mathcal{A}$-unsatisfiable $\Longleftarrow$ $\bot$ is derivable from S

S is $\mathcal{A}$-unsatisfiable $\Longrightarrow$ $\bot$ is derivable from S

completeness

$\Downarrow$

1. S is $\mathcal{A}$-*continuous*-unsatisfiable

$\Downarrow$

2. $\exists G \in S$ and $n \in \omega$ s.t. $\mathcal{A}_n^{\mathcal{C}} \not\models G$ $\Longrightarrow$ **3.** Reason for $\mathcal{A}_n^{\mathcal{C}} \not\models G$ can be captured *syntactically* by *unfolding definitions*

n-th stage of continuous *canonical* structure

✓ **soundness**

S is $\mathcal{A}$-unsatisfiable $\Longleftarrow$ / $\Longrightarrow$ $\bot$ is derivable from S

completeness

1. S is $\mathcal{A}$-*continuous*-unsatisfiable

$\Downarrow$

2. $\exists G \in S$ and $n \in \omega$ s.t. $\mathcal{A}_n^{\mathcal{C}} \not\models G$

$\Longrightarrow$

3. Reason for $\mathcal{A}_n^{\mathcal{C}} \not\models G$ can be captured *syntactically* by *unfolding definitions*

$\Uparrow$

4. Sufficient to *unfold* at *leftmost* positions of atoms

n-th stage of continuous *canonical* structure

1. S is $\mathcal{A}$-*continuous*-unsatisfiable if S is $\mathcal{A}$-unsatisfiable

Continuous Semantics

continuous interpretation $\mathcal{C}$ of types:

$$\mathcal{C}[\![\iota]\!] := \mathsf{dom}(\mathcal{A}) \quad \mathcal{C}[\![o]\!] := \mathbb{B} \quad \mathcal{C}[\![\tau \to \sigma]\!] := [\mathcal{C}[\![\tau]\!] \xrightarrow{c} \mathcal{C}[\![\sigma]\!]]$$

↗ *continuous* function space

Continuous Semantics

continuous interpretation $\mathcal{C}$ of types:

$$\mathcal{C}[\![\iota]\!] \coloneqq \mathsf{dom}(\mathcal{A}) \quad \mathcal{C}[\![o]\!] \coloneqq \mathbb{B} \quad \mathcal{C}[\![\tau \to \sigma]\!] \coloneqq [\mathcal{C}[\![\tau]\!] \xrightarrow{c} \mathcal{C}[\![\sigma]\!]]$$

continuous function space

Definition

A monotone $f : P \to Q$ is *continuous* if for all *directed* $D \subseteq P$,

$$f\left(\bigsqcup D\right) = \bigsqcup_{d \in D} f(d)$$

directed-complete posets

Continuous Semantics

continuous interpretation $\mathcal{C}$ of types:

$$\mathcal{C}[\![\iota]\!] := \mathsf{dom}(\mathcal{A}) \quad \mathcal{C}[\![o]\!] := \mathbb{B} \quad \mathcal{C}[\![\tau \to \sigma]\!] := [\mathcal{C}[\![\tau]\!] \xrightarrow{c} \mathcal{C}[\![\sigma]\!]]$$

continuous function space

Definition

A monotone $f : P \to Q$ is *continuous* if for all *directed* $D \subseteq P$,

$$f\left(\bigsqcup D\right) = \bigsqcup_{d \in D} f(d)$$

directed-complete posets

Structures $\mathcal{B}$, valuations α and denotations $\mathcal{B}[\![M]\!](\alpha)$ still as usual *(but w.r.t. $\mathcal{C}[\![-]\!]$!)*

Theorem

If S is $\mathcal{A}$-continuous-satisfiable then it is $\mathcal{A}$-satisfiable.

Theorem

If S is $\mathcal{A}$-continuous-satisfiable then it is $\mathcal{A}$-satisfiable.

Proof sketch. Define adjunctions for each type σ:

$$\mathcal{C}[\![\sigma]\!] \underset{L_\sigma}{\overset{I_\sigma}{\rightleftarrows}} \mathcal{S}[\![\sigma]\!] \quad (L_\sigma \dashv I_\sigma)$$

$$\underbrace{I(\mathcal{B}), \alpha \not\models G}_{\text{standard}} \text{ implies } \underbrace{\mathcal{B}, L \circ \alpha \not\models G}_{\text{continuous}}$$

□

2. $\exists G \in S$ and $n \in \omega$ s.t. $\mathcal{A}_n^{\mathcal{C}} \not\models G$

nth-stage of continuous *canonical* structure

Immediate Consequence Operator

Define the *immediate consequence operator* $T_S^{\mathcal{C}}$:

$$R^{T_S^{\mathcal{C}}(\mathcal{B})} := \mathcal{B}\left[\!\!\left[\lambda\overline{x}.\ \bigvee_{G \vee R\,\overline{x} \in S} \neg G\right]\!\!\right]$$

- prefixed points of $T_S^{\mathcal{C}}$ = models of definite clauses in S

Immediate Consequence Operator

Define the *immediate consequence operator* $T_S^{\mathcal{C}}$:

$$R^{T_S^{\mathcal{C}}(\mathcal{B})} := \mathcal{B}\left[\!\!\left[\lambda\overline{x}.\ \bigvee_{G \vee R\,\overline{x} \in S} \neg G\right]\!\!\right]$$

- prefixed points of $T_S^{\mathcal{C}}$ = models of definite clauses in S
- $T_S^{\mathcal{C}}$ is *continuous*

Canonical Structure

Define the *canonical continuous* structure:

$$\mathcal{A}_0^{\mathcal{C}} = \bot_{\Sigma'}$$
$$\mathcal{A}_{n+1}^{\mathcal{C}} = T_H^{\mathcal{C}}(\mathcal{A}_n^{\mathcal{C}}) \qquad n \in \omega$$
$$\mathcal{A}_\omega^{\mathcal{C}} = \bigsqcup_{n\in\omega} \mathcal{A}_n^{\mathcal{C}}$$

Theorem

$T_S^{\mathcal{C}}(\mathcal{A}_\omega^{\mathcal{C}}) \sqsubseteq \mathcal{A}_\omega^{\mathcal{C}}$. Hence, $\mathcal{A}_\omega^{\mathcal{C}} \models D$ for all definite $D \in S$.

Canonical Structure

Define the *canonical continuous* structure:

$$\begin{aligned} \mathcal{A}_0^{\mathcal{C}} &= \bot_{\Sigma'} \\ \mathcal{A}_{n+1}^{\mathcal{C}} &= T_H^{\mathcal{C}}(\mathcal{A}_n^{\mathcal{C}}) \qquad n \in \omega \\ \mathcal{A}_\omega^{\mathcal{C}} &= \bigsqcup_{n \in \omega} \mathcal{A}_n^{\mathcal{C}} \end{aligned}$$

Theorem

$T_S^{\mathcal{C}}(\mathcal{A}_\omega^{\mathcal{C}}) \sqsubseteq \mathcal{A}_\omega^{\mathcal{C}}$. Hence, $\mathcal{A}_\omega^{\mathcal{C}} \models D$ for all definite $D \in S$.

S is $\mathcal{A}$-continuous-unsatisfiable

- $\mathcal{A}_\omega^{\mathcal{C}}, \alpha \not\models G$ for some $G \in S$, valuation α

Canonical Structure

Define the *canonical continuous* structure:

$$\mathcal{A}_0^{\mathcal{C}} = \bot_{\Sigma'}$$
$$\mathcal{A}_{n+1}^{\mathcal{C}} = T_H^{\mathcal{C}}(\mathcal{A}_n^{\mathcal{C}}) \qquad n \in \omega$$
$$\mathcal{A}_\omega^{\mathcal{C}} = \bigsqcup_{n\in\omega} \mathcal{A}_n^{\mathcal{C}}$$

Theorem

$T_S^{\mathcal{C}}(\mathcal{A}_\omega^{\mathcal{C}}) \sqsubseteq \mathcal{A}_\omega^{\mathcal{C}}$. Hence, $\mathcal{A}_\omega^{\mathcal{C}} \models D$ for all definite $D \in S$.

S is $\mathcal{A}$-continuous-unsatisfiable

- $\mathcal{A}_\omega^{\mathcal{C}}, \alpha \not\models G$ for some $G \in S$, valuation α
- $\mathcal{A}_n^{\mathcal{C}}, \alpha \not\models G$ for some $n \in \omega$, $G \in S$, valuation α

✓ **soundness**

S is $\mathcal{A}$-unsatisfiable ⟸ ⟹ $\bot$ is derivable from S

completeness

1. S is $\mathcal{A}$-*continuous*-unsatisfiable

⇓

2. $\exists G \in S$ and $n \in \omega$ s.t. $\mathcal{A}^{\mathcal{C}}_n \not\models G$

⟹

3. Reason for $\mathcal{A}^{\mathcal{C}}_n \not\models G$ can be captured *syntactically* by *unfolding definitions*

⇑

4. Sufficient to *unfold* at *leftmost* positions of atoms

⇑

n-th stage of continuous *canonical* structure

Canonical Model Property and Semantic Invariance

Canonical Model Property

- $\mathcal{A}_\omega^\mathcal{C}$ is the *least continuous* model of S if it is satisfiable (**Kleene's fixed-point theorem**)

Canonical Model Property

- $\mathcal{A}_\omega^{\mathcal{C}}$ is the *least continuous* model of S if it is satisfiable (**Kleene's fixed-point theorem**)
- The *least* model property fails for standard semantics
- $T_S^{\mathcal{S}}$ is *not* even *monotone*

Canonical Model Property

- $\mathcal{A}^{\mathcal{C}}_{\omega}$ is the *least continuous* model of S if it is satisfiable (**Kleene's fixed-point theorem**)
- The *least* model property fails for standard semantics
- $T^{\mathcal{S}}_{S}$ is *not* even *monotone*

Theorem

$\bigsqcup_{\gamma \in \mathbf{On}} \mathcal{A}^{\mathcal{S}}_{\gamma}$ *is a model of S if it is $\mathcal{A}$-satisfiable.*

Canonical Model Property

- $\mathcal{A}_\omega^\mathcal{C}$ is the *least continuous* model of S if it is satisfiable (**Kleene's fixed-point theorem**)
- The *least* model property fails for standard semantics
- $T_S^\mathcal{S}$ is *not* even *monotone*

Theorem

$\bigsqcup_{\gamma \in \mathbf{On}} \mathcal{A}_\gamma^\mathcal{S}$ *is a model of S if it is $\mathcal{A}$-satisfiable.*

Proof sketch. Define a relation $\precsim$ on $\mathcal{S}[\![\sigma]\!]$ s.t. $0 \precsim 1$ and

$$\mathcal{B} \precsim \mathcal{B}' \wedge \alpha \precsim \alpha' \implies \mathcal{B}[\![M]\!](\alpha) \precsim \mathcal{B}'[\![M]\!](\alpha')$$

Canonical Model Property

- $\mathcal{A}_\omega^{\mathcal{C}}$ is the *least continuous* model of S if it is satisfiable (**Kleene's fixed-point theorem**)
- The *least* model property fails for standard semantics
- $T_S^{\mathcal{S}}$ is *not* even *monotone*

Theorem

$\bigsqcup_{\gamma \in \mathbf{On}} \mathcal{A}_\gamma^{\mathcal{S}}$ *is a model of S if it is $\mathcal{A}$-satisfiable.*

Proof sketch. Define a relation $\precsim$ on $\mathcal{S}[\![\sigma]\!]$ s.t. $0 \precsim 1$ and

$$\mathcal{B} \precsim \mathcal{B}' \wedge \alpha \precsim \alpha' \implies \mathcal{B}[\![M]\!](\alpha) \precsim \mathcal{B}'[\![M]\!](\alpha')$$

$\mathcal{A}_\beta^{\mathcal{S}} \precsim \mathcal{B}$ for all $\beta \in \mathbf{On}$ and $\mathcal{B} \models S$. □

Semantic Invariance

$\mathcal{A}$-satisfiable

✓ $\Uparrow$

$\mathcal{A}$-continuous-satisfiable

$$\mathcal{S}[\![\tau \to \sigma]\!] := [\mathcal{S}[\![\tau]\!] \to \mathcal{S}[\![\sigma]\!]]$$

$$\mathcal{C}[\![\tau \to \sigma]\!] := [\mathcal{C}[\![\tau]\!] \xrightarrow{c} \mathcal{C}[\![\sigma]\!]]$$

Semantic Invariance

$\mathcal{A}$-satisfiable

✓ $\Uparrow$ $\Downarrow$ soundness + completeness

$\mathcal{A}$-continuous-satisfiable

$$\mathcal{S}[\![\tau \to \sigma]\!] := [\mathcal{S}[\![\tau]\!] \to \mathcal{S}[\![\sigma]\!]]$$

$$\mathcal{C}[\![\tau \to \sigma]\!] := [\mathcal{C}[\![\tau]\!] \xrightarrow{c} \mathcal{C}[\![\sigma]\!]]$$

Semantic Invariance

$\mathcal{A}$-monotone-satisfiable

$\Updownarrow$ [POPL'18]

$\mathcal{A}$-**satisfiable**

✓ $\Uparrow$ $\Downarrow$ soundness + completeness

$\mathcal{A}$-continuous-satisfiable

$$\mathcal{M}[\![\tau \to \sigma]\!] := [\mathcal{M}[\![\tau]\!] \xrightarrow{m} \mathcal{M}[\![\sigma]\!]]$$

$$\mathcal{S}[\![\tau \to \sigma]\!] := [\mathcal{S}[\![\tau]\!] \to \mathcal{S}[\![\sigma]\!]]$$

$$\mathcal{C}[\![\tau \to \sigma]\!] := [\mathcal{C}[\![\tau]\!] \xrightarrow{c} \mathcal{C}[\![\sigma]\!]]$$

Semantic Invariance

$\mathcal{A}$-monotone-satisfiable $\qquad \mathcal{M}[\![\tau \to \sigma]\!] \coloneqq [\mathcal{M}[\![\tau]\!] \xrightarrow{m} \mathcal{M}[\![\sigma]\!]]$

$\Updownarrow$ [POPL'18]

$\mathcal{A}$-satisfiable $\qquad \mathcal{S}[\![\tau \to \sigma]\!] \coloneqq [\mathcal{S}[\![\tau]\!] \to \mathcal{S}[\![\sigma]\!]]$

✓ $\Uparrow$ $\Downarrow$ soundness + completeness

$\mathcal{A}$-continuous-satisfiable $\qquad \mathcal{C}[\![\tau \to \sigma]\!] \coloneqq [\mathcal{C}[\![\tau]\!] \xrightarrow{c} \mathcal{C}[\![\sigma]\!]]$

$\Downarrow$ ✓

$\mathcal{A}$-Henkin-satisfiable $\qquad \mathcal{H}[\![\tau \to \sigma]\!] \subseteq [\mathcal{H}[\![\tau]\!] \to \mathcal{H}[\![\sigma]\!]]$

Semantic Invariance

$\mathcal{A}$-monotone-satisfiable

$\Updownarrow$ [POPL'18]

$\mathcal{A}$-satisfiable

✓ $\Uparrow$ $\Downarrow$ soundness + completeness

$\mathcal{A}$-continuous-satisfiable

soundness + completeness $\Uparrow$ $\Downarrow$ ✓

$\mathcal{A}$-Henkin-satisfiable

$$\mathcal{M}[\![\tau \to \sigma]\!] := [\mathcal{M}[\![\tau]\!] \xrightarrow{m} \mathcal{M}[\![\sigma]\!]]$$

$$\mathcal{S}[\![\tau \to \sigma]\!] := [\mathcal{S}[\![\tau]\!] \to \mathcal{S}[\![\sigma]\!]]$$

$$\mathcal{C}[\![\tau \to \sigma]\!] := [\mathcal{C}[\![\tau]\!] \xrightarrow{c} \mathcal{C}[\![\sigma]\!]]$$

$$\mathcal{H}[\![\tau \to \sigma]\!] \subseteq [\mathcal{H}[\![\tau]\!] \to \mathcal{H}[\![\sigma]\!]]$$

Conclusion

*HoCHC lies at a **"sweet spot"** in higher-order logic, semantically robust and useful for algorithmic verification.*

Conclusion

This talk:

- A *simple* resolution proof system for HoCHC
  - Completeness even for *standard* semantics
- Canonical model property and semantic invariance of HoCHC

Conclusion

This talk:

- A *simple* resolution proof system for HoCHC
  - Completeness even for *standard* semantics
- Canonical model property and semantic invariance of HoCHC

Also in the paper:

- Extension to *compact* theories
- 1st-order translation (complete for *standard* semantics)
- *Decidable* fragments

Conclusion

This talk:

- A *simple* resolution proof system for HoCHC
  - Completeness even for *standard* semantics
- Canonical model property and semantic invariance of HoCHC

Also in the paper:

- Extension to *compact* theories
- 1st-order translation (complete for *standard* semantics)
- *Decidable* fragments

Future directions:

- Implementation
- Improve *robustness* on satisfiable instances

Horus (`http://mjolnir.cs.ox.ac.uk/horus/`)
DefMono (`http://mjolnir.cs.ox.ac.uk/dfhochc/`)

$$\neg(z = x + y) \;\vee\; \mathsf{Add}\; x\, y\, z \;=:\; D_1$$
$$\neg(n \le 0) \;\vee\; \neg(s = x) \;\vee\; \mathsf{Iter}\; f\, s\, n\, x \;=:\; D_2$$
$$\neg(n > 0) \;\vee\; \neg\,\mathsf{Iter}\; f\, s\,(n-1)\, y \;\vee\; \neg(f\; n\, y\, x) \;\vee\; \mathsf{Iter}\; f\, s\, n\, x \;=:\; D_3$$

$$\neg(n \ge 1) \vee \neg\,\mathsf{Iter}\;\mathsf{Add}\; n\, n\, x \;\vee \neg(x \le n + n)$$

$$\neg(z = x + y) \lor \mathsf{Add}\ x\ y\ z =: D_1$$

$$\neg(n \leq 0) \lor \neg(s = x) \lor \mathsf{Iter}\ f\ s\ n\ x =: D_2$$

$$\neg(n > 0) \lor \neg\,\mathsf{Iter}\ f\ s\ (n-1)\ y \lor \neg(f\ n\ y\ x) \lor \mathsf{Iter}\ f\ s\ n\ x =: D_3$$

$$\neg(n \geq 1) \lor \neg\,\mathsf{Iter}\ \mathsf{Add}\ n\ n\ x \lor \neg(x \leq n + n)$$

$$\neg(z = x + y) \lor \text{Add}\ x\ y\ z =: D_1$$
$$\neg(n \leq 0) \lor \neg(s = x) \lor \text{Iter}\ f\ s\ n\ x =: D_2$$
$$\neg(n > 0) \lor \neg\,\text{Iter}\ f\ s\ (n-1)\ y \lor \neg(f\ n\ y\ x) \lor \text{Iter}\ f\ s\ n\ x =: D_3$$

$$\text{Res.}\ \frac{\neg(n \geq 1) \lor \neg\,\text{Iter Add}\ n\ n\ x \lor \neg(x \leq n + n) \qquad D_3}{\neg(n \geq 1) \lor \neg(n > 0) \lor \neg\,\text{Iter Add}\ n\ (n-1)\ y \lor \neg\,\text{Add}\ n\ y\ x \lor \neg(x \leq n + n)}$$

$$\neg(z = x + y) \lor \text{Add } x\,y\,z =: D_1$$
$$\neg(n \leq 0) \lor \neg(s = x) \lor \text{Iter } f\,s\,n\,x =: D_2$$
$$\neg(n > 0) \lor \neg\,\text{Iter } f\,s\,(n-1)\,y \lor \neg(f\,n\,y\,x) \lor \text{Iter } f\,s\,n\,x =: D_3$$

$$\text{Res.}\ \frac{\neg(n \geq 1) \lor \neg\,\text{Iter Add } n\,n\,x \lor \neg(x \leq n + n) \qquad D_3}{\neg(n \geq 1) \lor \neg(n > 0) \lor \neg\,\text{Iter Add } n\,(n-1)\,y \lor \neg\,\text{Add } n\,y\,x \lor \neg(x \leq n + n)}$$

$$
\begin{aligned}
\neg(z = x + y) \lor \text{Add}\ x\,y\,z &=: D_1 \\
\neg(n \le 0) \lor \neg(s = x) \lor \text{Iter}\ f\,s\,n\,x &=: D_2 \\
\neg(n > 0) \lor \neg\,\text{Iter}\ f\,s\,(n-1)\,y \lor \neg(f\,n\,y\,x) \lor \text{Iter}\ f\,s\,n\,x &=: D_3
\end{aligned}
$$

$$
\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\neg(n \ge 1) \lor \neg\,\text{Iter Add}\ n\,n\,x \lor \neg(x \le n+n) \qquad D_3}{\neg(n \ge 1) \lor \neg(n > 0) \lor \neg\,\text{Iter Add}\ n\,(n-1)\,y \lor \neg\,\text{Add}\ n\,y\,x \lor \neg(x \le n+n)} \qquad D_1}{\neg(n \ge 1) \lor \neg(n > 0) \lor \neg\,\text{Iter Add}\ n\,(n-1)\,y \lor \neg(x = n+y) \lor \neg(x \le n+n)}
$$

$$\begin{aligned}
\neg(z = x + y) \;\vee\; \mathrm{Add}\; x\, y\, z \; &=: D_1 \\
\neg(n \le 0) \;\vee\; \neg(s = x) \;\vee\; \mathrm{Iter}\; f\, s\, n\, x \; &=: D_2 \\
\neg(n > 0) \;\vee\; \neg\,\mathrm{Iter}\; f\, s\,(n-1)\, y \;\vee\; \neg(f\, n\, y\, x) \;\vee\; \mathrm{Iter}\; f\, s\, n\, x \; &=: D_3
\end{aligned}$$

$$\text{Res.}\;\dfrac{\text{Res.}\;\dfrac{\neg(n \ge 1) \vee \neg\,\mathrm{Iter\ Add}\; n\, n\, x \vee \neg(x \le n + n) \qquad D_3}{\neg(n \ge 1) \vee \neg(n > 0) \vee \neg\,\mathrm{Iter\ Add}\; n\,(n-1)\, y \vee \neg\,\mathrm{Add}\; n\, y\, x \vee \neg(x \le n + n)} \qquad D_1}{\neg(n \ge 1) \vee \neg(n > 0) \vee \neg\,\mathrm{Iter\ Add}\; n\,(n-1)\, y \vee \neg(x = n + y) \vee \neg(x \le n + n)}$$

$$\begin{aligned} \neg(z = x + y) \lor \mathrm{Add}\; x\, y\, z &=: D_1 \\ \neg(n \leq 0) \lor \neg(s = x) \lor \mathrm{Iter}\; f\, s\, n\, x &=: D_2 \\ \neg(n > 0) \lor \neg\,\mathrm{Iter}\; f\, s\,(n-1)\, y \lor \neg(f\, n\, y\, x) \lor \mathrm{Iter}\; f\, s\, n\, x &=: D_3 \end{aligned}$$

$$\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\neg(n \geq 1) \lor \neg\,\mathrm{Iter\ Add}\; n\, n\, x \lor \neg(x \leq n+n) \qquad D_3}{\neg(n \geq 1) \lor \neg(n > 0) \lor \neg\,\mathrm{Iter\ Add}\; n\,(n-1)\, y \lor \neg\,\mathrm{Add}\; n\, y\, x \lor \neg(x \leq n+n)} \quad D_1}{\neg(n \geq 1) \lor \neg(n > 0) \lor \neg\,\mathrm{Iter\ Add}\; n\,(n-1)\, y \lor \neg(x = n+y) \lor \neg(x \leq n+n)} \quad D_2}{\neg(n \geq 1) \lor \neg(n > 0) \lor \neg(n-1 \leq 0) \lor \neg(n = y) \lor \neg(x = n+y) \lor \neg(x \leq n+n)}$$

$$\neg(z = x + y) \;\vee\; \text{Add}\; x\,y\,z \;=:\; D_1$$
$$\neg(n \leq 0) \;\vee\; \neg(s = x) \;\vee\; \text{Iter}\; f\,s\,n\,x \;=:\; D_2$$
$$\neg(n > 0) \;\vee\; \neg\,\text{Iter}\; f\,s\,(n-1)\,y \;\vee\; \neg(f\,n\,y\,x) \;\vee\; \text{Iter}\; f\,s\,n\,x \;=:\; D_3$$

$$\text{Res.}\;\frac{\neg(n \geq 1) \vee \neg\,\text{Iter Add}\; n\,n\,x \;\vee \neg(x \leq n+n) \qquad D_3}{\neg(n \geq 1) \vee \neg(n > 0) \vee \neg\,\text{Iter Add}\; n\,(n-1)\,y \;\vee\; \neg\,\text{Add}\; n\,y\,x \;\vee \neg(x \leq n+n)} \quad D_1$$

$$\text{Res.}\;\frac{\cdots}{\neg(n \geq 1) \vee \neg(n > 0) \vee \neg\,\text{Iter Add}\; n\,(n-1)\,y \;\vee\; \neg(x = n+y) \;\vee \neg(x \leq n+n)} \quad D_2$$

$$\text{Res.}\;\frac{\cdots}{\neg(n \geq 1) \;\vee\; \neg(n > 0) \;\vee\; \neg(n-1 \leq 0) \;\vee\; \neg(n = y) \vee \neg(x = n+y) \vee \neg(x \leq n+n)}$$

$\alpha(n) = 1$ $\qquad$ $\alpha(x) = 2$ $\qquad$ $\alpha(y) = 1$

$$
\begin{aligned}
\neg(z = x + y) \;\vee\; \text{Add}\ x\, y\, z \;&=:\; D_1 \\
\neg(n \le 0) \;\vee\; \neg(s = x) \;\vee\; \text{Iter}\ f\, s\, n\, x \;&=:\; D_2 \\
\neg(n > 0) \;\vee\; \neg\,\text{Iter}\ f\, s\,(n-1)\, y \;\vee\; \neg(f\, n\, y\, x) \;\vee\; \text{Iter}\ f\, s\, n\, x \;&=:\; D_3
\end{aligned}
$$

$$
\text{Const. Ref.}\ \dfrac{\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\neg(n \ge 1) \vee \neg\,\text{Iter Add}\ n\, n\, x \vee \neg(x \le n + n) \qquad D_3}{\neg(n \ge 1) \vee \neg(n > 0) \vee \neg\,\text{Iter Add}\ n\,(n-1)\, y \vee \neg\,\text{Add}\ n\, y\, x \vee \neg(x \le n + n)} \quad D_1}{\neg(n \ge 1) \vee \neg(n > 0) \vee \neg\,\text{Iter Add}\ n\,(n-1)\, y \vee \neg(x = n + y) \vee \neg(x \le n + n)} \quad D_2}{\neg(n \ge 1) \vee \neg(n > 0) \vee \neg(n - 1 \le 0) \vee \neg(n = y) \vee \neg(x = n + y) \vee \neg(x \le n + n)}}{\bot}
$$

$\alpha(n) = 1 \qquad \alpha(x) = 2 \qquad \alpha(y) = 1$