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

Luke Ong **Dominik Wagner**

LICS 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

> *“$\overbrace{\textit{Constrained}}^{\textit{1st-order}}$ Horn Clauses provide a suitable basis for automatic $\underbrace{\textit{program}}_{\textit{imperative}}$ 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

[Cathcart Burn, Ong & Ramsay; POPL'18]:

extend approach to higher-orders

```
let add    x y = x + y
```

```
let add     x y = x + y
let twice f x = f (f x)
```

```
let add   x y = x + y
let twice f x = f (f x)
in  λx. assert (x >= 1 ->
                     (twice (add x) 0) > x)
```

```
let add     x y = x + y
let twice f x = f (f x)
in  λx. assert (x >= 1 ->
                    (twice (add x) 0) > x)
```

(over-)approximate graph of functions

```
let add    x y = x + y
let twice f x = f (f x)
in  λx. assert (x >= 1 ->
                  (twice (add x) 0) > x)
```

(over-)approximate graph of functions

$$\forall x,y,z.\left(z = x + y \to \mathrm{Add}\ x\ y\ z\right)$$

```
let add    x y = x + y
let twice f x = f (f x)
in  λx. assert (x >= 1 ->
                    (twice (add x) 0) > x)
```

(over-)approximate graph of functions

$$\forall_{x,y,z}.\big(z = x + y \rightarrow \mathsf{Add}\ x\ y\ z\big)$$

$$\forall_{f,x,z}.\big(\exists y.\,(f\ x\ y) \wedge (f\ y\ z) \rightarrow \mathsf{Twice}\ f\ x\ z\big)$$

```
let add    x y = x + y
let twice f x = f (f x)
in  λx. assert (x >= 1 ->
                   (twice (add x) 0) > x)
```

(over-)approximate graph of functions

$$\forall x,y,z. \big(z = x + y \to \mathsf{Add}\; x\, y\, z\big)$$

$$\forall f,x,z. \big(\exists y. (f\, x\, y) \land (f\, y\, z) \to \mathsf{Twice}\; f\, x\, z\big)$$

$$\forall x,z. \big(x \geq 1 \land \mathsf{Twice}\,(\mathsf{Add}\; x)\, 0\, z \;\to\; z > x\big)$$

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

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

	1st-order logic	*HoCHC* ~~higher-order logic~~	
		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

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

This talk:

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

Part I:
HoCHC

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

$$\begin{aligned} \neg(z = x + y) \vee \mathsf{Add}\ x\, y\, z \\ \neg(f\ x\, y) \vee \neg(f\ y\, z) \vee \mathsf{Twice}\ f\ x\, z \\ \neg(x \geq 1) \vee \neg\,\mathsf{Twice}\,(\mathsf{Add}\ x)\,0\,z \vee \neg(z \leq x) \end{aligned}$$

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

background theory relational extension

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

$$\neg(f\ x\ y) \lor \neg(f\ y\ z) \lor \mathsf{Twice}\ f\ x\ z$$

$$\neg(x \geq 1) \lor \neg\,\mathsf{Twice}\,(\mathsf{Add}\ x)\,0\,z \lor \neg(z \leq x)$$

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

background theory   relational extension

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

$$\neg(f\, x\, y) \vee \neg(f\, y\, z) \vee \mathsf{Twice}\; f\, x\, z$$

$$\neg(x \geq 1) \vee \neg\, \mathsf{Twice}\, (\mathsf{Add}\; x)\, 0\, z \vee \neg(z \leq x)$$

- only *relational* higher-order types

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

definite clauses

$$\begin{aligned} \neg(z = x + y) \lor \mathsf{Add}\ x\,y\,z \\ \neg(f\,x\,y) \lor \neg(f\,y\,z) \lor \mathsf{Twice}\ f\,x\,z \\ \neg(x \geq 1) \lor \neg\,\mathsf{Twice}\,(\mathsf{Add}\ x)\,0\,z \lor \neg(z \leq x) \end{aligned}$$

goal clause

- only *relational* higher-order types

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

distinct variables

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

$$\neg(f\ x\ y) \vee \neg(f\ y\ z) \vee \mathsf{Twice}\ f\ x\ z$$

$$\neg(x \geq 1) \vee \neg\,\mathsf{Twice}\,(\mathsf{Add}\ x)\,0\,z \vee \neg(z \leq x)$$

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

Syntactic Features

signatures $\Sigma \subseteq \Sigma'$

distinct variables

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

$$\neg(f\; x\, y) \lor \neg(f\; y\, z) \lor \mathsf{Twice}\; f\, x\, z$$

$$\neg(x \geq 1) \lor \neg\, \mathsf{Twice}\, (\mathsf{Add}\; x)\, 0\, z \lor \neg(z \leq x)$$

- only *relational* higher-order types
- positive literals are *definitional*
- no logical symbols in atoms: $\xcancel{R\,\neg}$
- 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

$$\mathcal{S}[\![\iota]\!] := \mathsf{dom}(\mathcal{A}) \quad \mathcal{S}[\![o]\!] := \{0, 1\} \quad \mathcal{S}[\![\tau \to \sigma]\!] := \ [\mathcal{S}[\![\tau]\!] \to \mathcal{S}[\![\sigma]\!]]$$

Standard Semantics

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

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

$$\mathcal{S}[\![\iota]\!] := \mathsf{dom}(\mathcal{A}) \quad \mathcal{S}[\![o]\!] := \{0,1\} \quad \mathcal{S}[\![\tau \to \sigma]\!] := [\mathcal{S}[\![\tau]\!] \to \mathcal{S}[\![\sigma]\!]]$$

Structures $\mathcal{B}$, valuations α and denotations $\mathcal{B}[\![M]\!](\alpha)$ as usual

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

Γ: set of HoCHCs

Satisfiability

Γ 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

Γ: set of HoCHCs

Satisfiability

Γ 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 $C \in \Gamma$ and valuation α.

Part II:

Canonical Model Property

Immediate Consequence Operator

$T_\Gamma(\mathcal{B})$ $\mathcal{B}$

Immediate Consequence Operator

Idea: *satisfy what needs to be satisfied*

$T_{\Gamma}(\mathcal{B})$ $\mathcal{B}$

Immediate Consequence Operator

Idea: *satisfy what needs to be satisfied*

$$\neg A_1 \vee \cdots \vee \neg A_n \vee R\,\overline{x} \in \Gamma$$

$$T_\Gamma(\mathcal{B}), \alpha \models R\,\overline{x} \quad \Longleftrightarrow \quad \mathcal{B}, \alpha \not\models \neg A_1 \vee \cdots \vee \neg A_n$$

Immediate Consequence Operator

Idea: *satisfy what needs to be satisfied*

$$T_\Gamma(\mathcal{B}), \alpha \models R\,\overline{x} \quad\iff\quad \mathcal{B}, \alpha \not\models \overset{\neg A_1 \vee \cdots \vee \neg A_n \vee R\,\overline{x} \in \Gamma}{\neg A_1 \vee \cdots \vee \neg A_n}$$

prefixed points of T_Γ = models of *definite* clauses in Γ

1st-order:

$$\begin{tikzcd}
T_{\Gamma} \text{ is monotone} \arrow[d, Rightarrow, "\text{Knaster-Tarski}"] \\
\Gamma \text{ has } \textit{least} \text{ model}
\end{tikzcd}$$

The *least* model property *fails* for standard semantics!

1st-order:

$$\begin{tikzcd}
T_\Gamma \text{ is monotone} \arrow[d, Rightarrow, "\text{Knaster-Tarski}"] \\
\Gamma \text{ has \textit{least} model}
\end{tikzcd}$$

The *least* model property *fails* for standard semantics!

1st-order:

T_Γ is monotone

$$\Big\Downarrow \text{ Knaster-Tarski}$$

Γ has *least* model

higher-order:

T_Γ is *quasi-monotone*

$$\Big\Downarrow \begin{array}{l}\textit{extension} \text{ of}\\ \text{Knaster-Tarski}\end{array}$$

Γ has *canonical* model

Fix: $(L, \leq)$ complete lattice, $F : L \to L$,

Fix: $(L, \leq)$ complete lattice, $F : L \to L$,

$$a_0 := \bot \qquad a_1 := F(a_0) \qquad a_2 := F(a_1) \quad \ldots \quad a_\omega := \bigvee_{n \in \omega} a_n \quad \ldots$$

$$a_F := \bigvee_{\beta \in \mathbf{On}} a_\beta$$

Fix: $(L, \leq)$ complete lattice, $F : L \to L$,

$$a_0 := \bot \qquad a_1 := F(a_0) \qquad a_2 := F(a_1) \quad \ldots \quad a_\omega := \bigvee_{n \in \omega} a_n \quad \ldots$$

$$a_F := \bigvee_{\beta \in \mathbf{On}} a_\beta$$

Proposition ("Extended Knaster-Tarski")

1. $F(a_F) \leq a_F$

Fix: $(L, \leq)$ complete lattice, $F : L \to L$, $\precsim \subseteq L \times L$

$$a_0 := \bot \qquad a_1 := F(a_0) \qquad a_2 := F(a_1) \quad \ldots \quad a_\omega := \bigvee_{n \in \omega} a_n \quad \ldots$$

$$a_F := \bigvee_{\beta \in \mathbf{On}} a_\beta$$

Proposition ("Extended Knaster-Tarski")

1. $F(a_F) \leq a_F$

Fix: $(L, \leq)$ complete lattice, $F : L \to L$, $\precsim \subseteq L \times L$

$$a_0 := \bot \qquad a_1 := F(a_0) \qquad a_2 := F(a_1) \quad \dots \quad a_\omega := \bigvee_{n \in \omega} a_n \quad \dots$$

$$a_F := \bigvee_{\beta \in \mathbf{On}} a_\beta$$

Definition

F is *quasi-monotone* if $a \precsim b \implies F(a) \precsim F(b)$.

Proposition ("Extended Knaster-Tarski")

1. $F(a_F) \leq a_F$

Fix: $(L, \leq)$ complete lattice, $F : L \to L$, $\precsim \subseteq L \times L$

$$a_0 := \bot \qquad a_1 := F(a_0) \qquad a_2 := F(a_1) \quad \dots \quad a_\omega := \bigvee_{n \in \omega} a_n \quad \dots$$

$$a_F := \bigvee_{\beta \in \mathbf{On}} a_\beta$$

Definition

F is *quasi-monotone* if $a \precsim b \implies F(a) \precsim F(b)$.

Proposition ("Extended Knaster-Tarski")

1. $F(a_F) \leq a_F$
2. $$\left.\begin{array}{ll} \textit{(i)} & F(b) \leq b \\ \textit{(ii)} & F \textit{ is quasi-monotone} \\ \textit{(iii)} & \precsim \textit{ is compatible with } \leq \end{array}\right\} \implies a_F \precsim b$$

Use: T_Γ

Use: T_Γ

- $\mathcal{A}_\Gamma \models \{D \in \Gamma \mid D \text{ definite}\}$ canonical structure

Use: T_Γ and *logical relations* $\precsim_\sigma \subseteq \mathcal{S}[\![\sigma]\!] \times \mathcal{S}[\![\sigma]\!]$

- $\mathcal{A}_\Gamma \models \{D \in \Gamma \mid D \text{ definite}\}$ canonical structure

Use: T_Γ and *logical relations* $\precsim_\sigma \subseteq \mathcal{S}[\![\sigma]\!] \times \mathcal{S}[\![\sigma]\!]$

Lemma (Fundamental Theorem)

$$\left.\begin{array}{c} \mathcal{B} \precsim \mathcal{B}' \\ \alpha \precsim \alpha' \end{array}\right\} \implies \mathcal{B}[\![M]\!](\alpha) \precsim \mathcal{B}'[\![M]\!](\alpha')$$

- $\mathcal{A}_\Gamma \models \{D \in \Gamma \mid D \text{ definite}\}$ canonical structure

Use: T_Γ and *logical relations* $\precsim_\sigma \subseteq \mathcal{S}[\![\sigma]\!] \times \mathcal{S}[\![\sigma]\!]$

Lemma (Fundamental Theorem)

$$\left.\begin{array}{c} \mathcal{B} \precsim \mathcal{B}' \\ \alpha \precsim \alpha' \end{array}\right\} \implies \mathcal{B}[\![M]\!](\alpha) \precsim \mathcal{B}'[\![M]\!](\alpha')$$

- $\mathcal{A}_\Gamma \models \{D \in \Gamma \mid D \text{ definite}\}$

canonical structure

- T_Γ is quasi-monotone
- $\mathcal{A}_\Gamma \precsim \mathcal{B}$ if $\mathcal{B} \models \Gamma$

Use: T_Γ and *logical relations* $\precsim_\sigma \subseteq \mathcal{S}[\![\sigma]\!] \times \mathcal{S}[\![\sigma]\!]$

Lemma (Fundamental Theorem)

$$\left.\begin{array}{c} \mathcal{B} \precsim \mathcal{B}' \\ \alpha \precsim \alpha' \end{array}\right\} \implies \mathcal{B}[\![M]\!](\alpha) \precsim \mathcal{B}'[\![M]\!](\alpha')$$

- $\mathcal{A}_\Gamma \models \{D \in \Gamma \mid D \text{ definite}\}$ canonical structure
- T_Γ is quasi-monotone
- $\mathcal{A}_\Gamma \precsim \mathcal{B}$ if $\mathcal{B} \models \Gamma$
- $\mathcal{A}_\Gamma \models G$ if $\mathcal{B} \models \Gamma$ $G \in \Gamma$ goal clause

Use: T_Γ and *logical relations* $\precsim_\sigma \subseteq \mathcal{S}[\![\sigma]\!] \times \mathcal{S}[\![\sigma]\!]$

Lemma (Fundamental Theorem)

$$\left.\begin{array}{c} \mathcal{B} \precsim \mathcal{B}' \\ \alpha \precsim \alpha' \end{array}\right\} \implies \mathcal{B}[\![M]\!](\alpha) \precsim \mathcal{B}'[\![M]\!](\alpha')$$

- $\mathcal{A}_\Gamma \models \{D \in \Gamma \mid D \text{ definite}\}$ canonical structure
- T_Γ is quasi-monotone
- $\mathcal{A}_\Gamma \precsim \mathcal{B}$ if $\mathcal{B} \models \Gamma$
- $\mathcal{A}_\Gamma \models G$ if $\mathcal{B} \models \Gamma$ $G \in \Gamma$ goal clause

Theorem (Canonical Model Property)

$\mathcal{A}_\Gamma \models \Gamma$ if Γ is $\mathcal{A}$-satisfiable.

Part III:
Resolution Proof System

Proof System

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

Proof System

Resolution

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

background atoms

Constraint Refutation

$$\frac{\neg\,\varphi_1 \;\lor \cdots \lor \neg\,\varphi_n}{\bot}$$

provided there exists a valuation α s.t. $\mathcal{A}, \alpha \not\models \neg\varphi_1 \lor \cdots \lor \neg\varphi_n$

Proof System

Resolution

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

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 there exists a valuation α s.t. $\mathcal{A}, \alpha \not\models \neg\varphi_1 \lor \cdots \lor \neg\varphi_n$

Proof System

Resolution

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

variables

background atoms

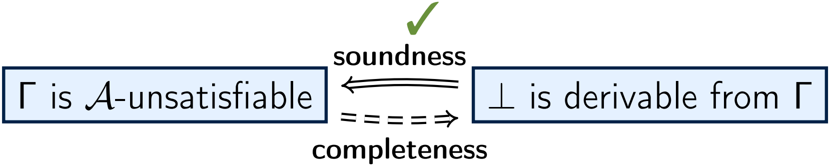
Constraint Refutation

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

provided there exists a valuation α s.t. $\mathcal{A}, \alpha \not\models \neg\varphi_1 \vee \cdots \vee \neg\varphi_n$

(+ rule for β-reduction in paper)

$$\boxed{\Gamma \text{ is } \mathcal{A}\text{-unsatisfiable}} \overset{\checkmark}{\xLeftarrow{\textbf{soundness}}} \boxed{\bot \text{ is derivable from } \Gamma}$$



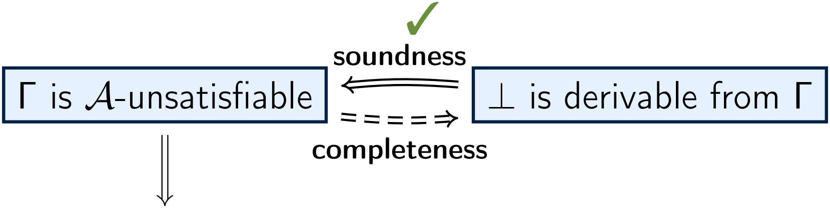
✓

soundness

Γ is $\mathcal{A}$-unsatisfiable

$\perp$ is derivable from Γ

completeness



✓

soundness

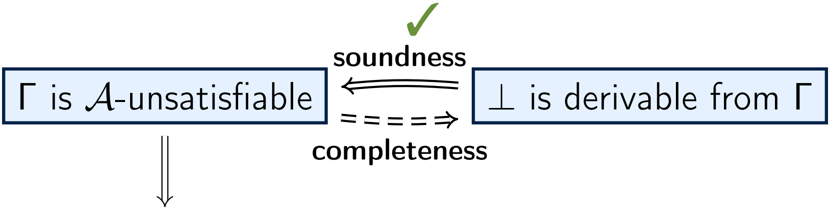
Γ is $\mathcal{A}$-unsatisfiable

$\bot$ is derivable from Γ

completeness

1. $\exists G \in \Gamma$ s.t.

   $\mathcal{A}_\Gamma \not\models G$


✓

soundness

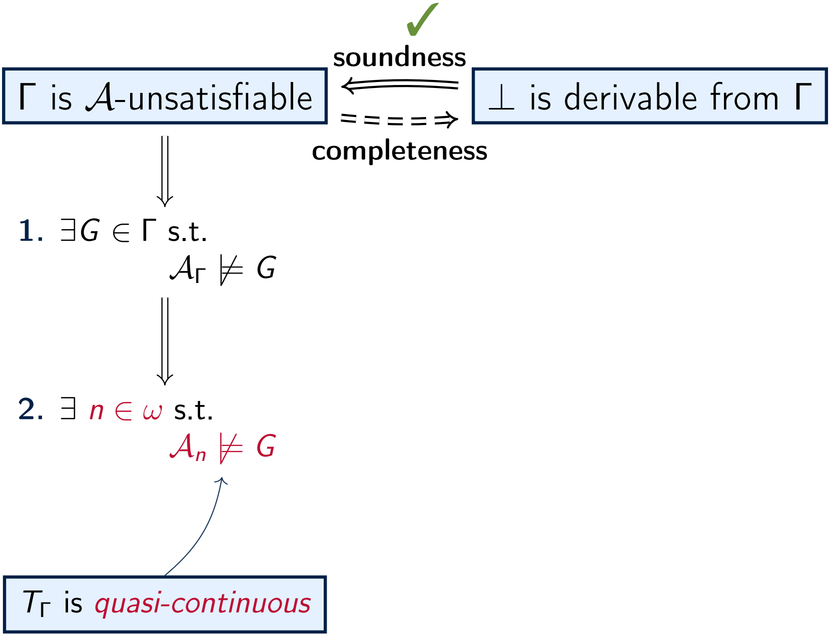
Γ is $\mathcal{A}$-unsatisfiable

$\bot$ is derivable from Γ

completeness

1. $\exists G \in \Gamma$ s.t.
   $\mathcal{A}_\Gamma \not\models G$

T_Γ is *quasi-continuous*



✓

soundness

Γ is $\mathcal{A}$-unsatisfiable $\Longleftarrow$ ⊥ is derivable from Γ

completeness

$\Downarrow$

1. $\exists G \in \Gamma$ s.t.

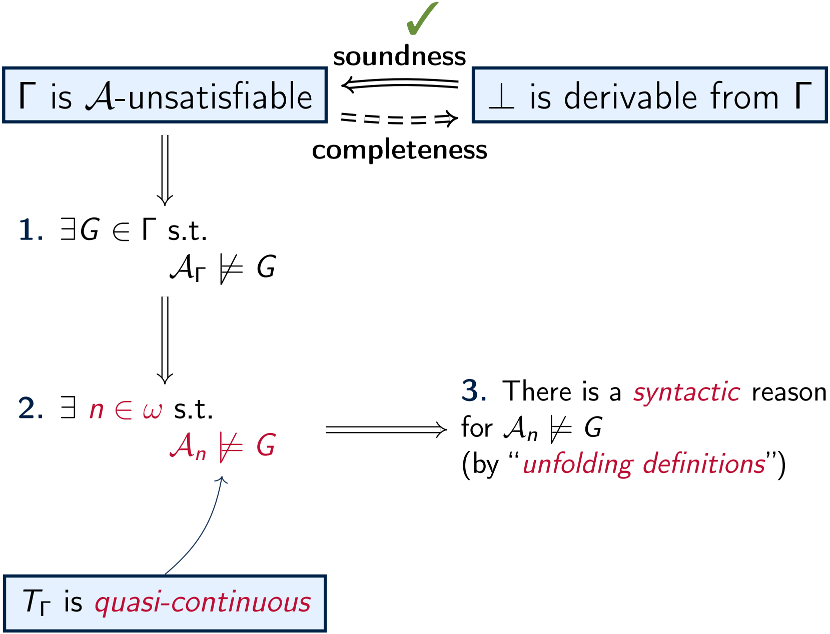
$\mathcal{A}_\Gamma \not\models G$

$\Downarrow$

2. $\exists\ n \in \omega$ s.t.

$\mathcal{A}_n \not\models G$

T_Γ is *quasi-continuous*



✓

soundness

Γ is $\mathcal{A}$-unsatisfiable

$\perp$ is derivable from Γ

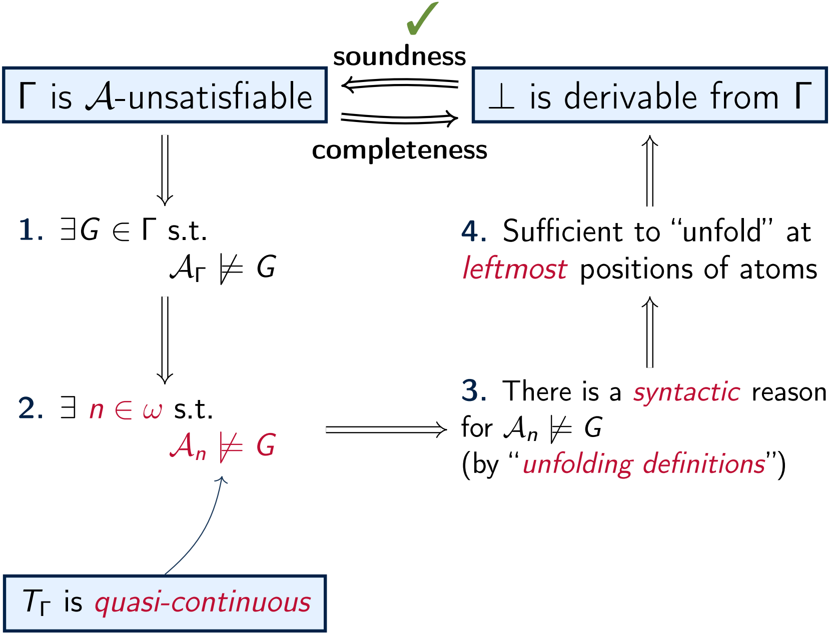
completeness

1. $\exists G \in \Gamma$ s.t.
$\mathcal{A}_\Gamma \not\models G$

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

3. There is a *syntactic* reason
for $\mathcal{A}_n \not\models G$
(by "*unfolding definitions*")

T_Γ is *quasi-continuous*



Γ is $\mathcal{A}$-unsatisfiable

✓ **soundness**

completeness

$\bot$ is derivable from Γ

1. $\exists G \in \Gamma$ s.t. $\mathcal{A}_\Gamma \not\models G$

2. $\exists\, n \in \omega$ s.t. $\mathcal{A}_n \not\models G$

3. There is a *syntactic* reason for $\mathcal{A}_n \not\models G$ (by "*unfolding definitions*")

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

T_Γ is *quasi-continuous*

Part IV:

Semantic Invariance

Semantic Invariance

$\boxed{\mathcal{A}\text{-}\mathbf{satisfiable}}$

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

Semantic Invariance

$\mathcal{A}$-monotone-satisfiable

$\Updownarrow$ [POPL'18]

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

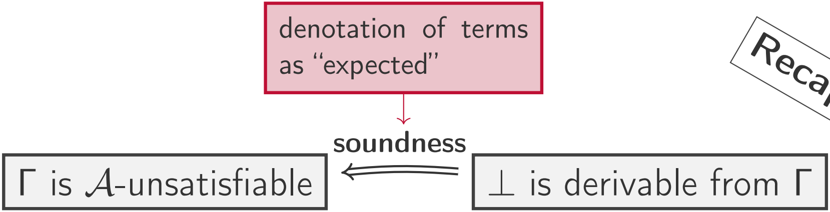
$$\mathcal{M}[\![\tau \to \sigma]\!] \coloneqq [\mathcal{M}[\![\tau]\!] \xrightarrow{m} \mathcal{M}[\![\sigma]\!]]$$

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

Recap

$$\boxed{\Gamma \text{ is } \mathcal{A}\text{-unsatisfiable}} \xLeftarrow{\textbf{soundness}} \boxed{\bot \text{ is derivable from } \Gamma}$$

Recap



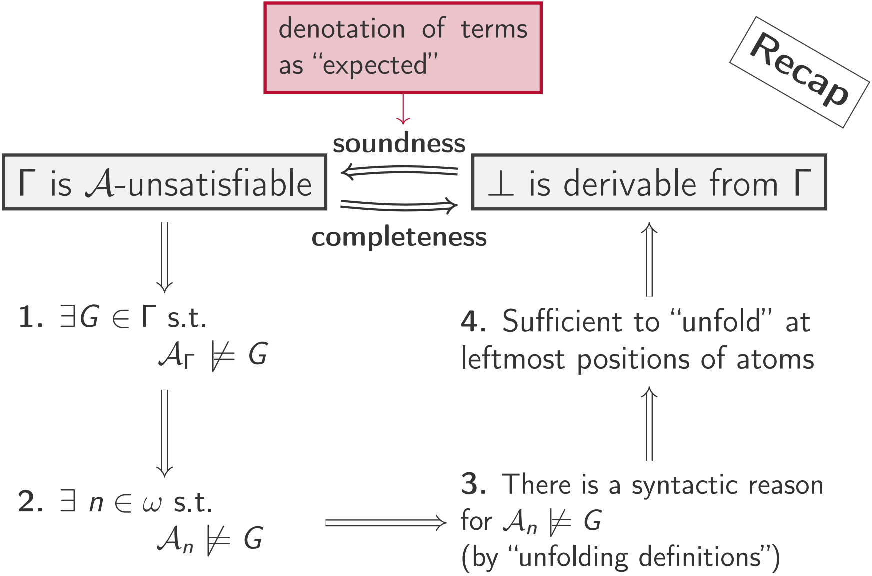
denotation of terms as “expected”

soundness

Γ is $\mathcal{A}$-unsatisfiable

$\bot$ is derivable from Γ

Recap



denotation of terms as "expected"

soundness

Γ is $\mathcal{A}$-unsatisfiable

$\perp$ is derivable from Γ

completeness

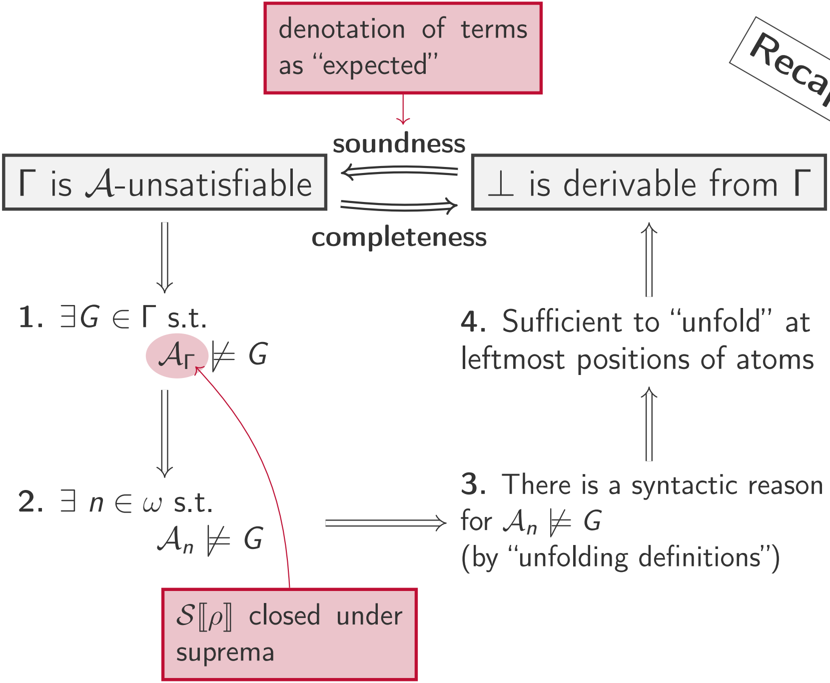
1. $\exists G \in \Gamma$ s.t. $\mathcal{A}_\Gamma \not\models G$

2. $\exists\, n \in \omega$ s.t. $\mathcal{A}_n \not\models G$

3. There is a syntactic reason for $\mathcal{A}_n \not\models G$ (by "unfolding definitions")

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

Recap



denotation of terms as “expected”

soundness

Γ is $\mathcal{A}$-unsatisfiable

⊥ is derivable from Γ

completeness

1. $\exists G \in \Gamma$ s.t.
$\mathcal{A}_\Gamma \not\models G$

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

3. There is a syntactic reason for $\mathcal{A}_n \not\models G$ (by “unfolding definitions”)

4. Sufficient to “unfold” at leftmost positions of atoms

$\mathcal{S}[\![\rho]\!]$ closed under suprema

Generalised HoCHC Satisfiability Problem

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

Γ: set of HoCHCs

$\mathcal{F}$: fixed interpretation of types s.t.

Generalised HoCHC Satisfiability Problem

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

Γ: set of HoCHCs

$\mathcal{F}$: fixed interpretation of types s.t.

sufficiently rich to give "expected" denotations

$$\mathcal{F}[\![\iota]\!] = \mathsf{dom}(\mathcal{A}) \quad \mathcal{F}[\![o]\!] = \{0,1\} \quad \mathcal{F}[\![\tau \to \sigma]\!] \subseteq [\mathcal{F}[\![\tau]\!] \to \mathcal{F}[\![\sigma]\!]]$$

Generalised HoCHC Satisfiability Problem

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

Γ: set of HoCHCs

$\mathcal{F}$: fixed interpretation of types s.t.

sufficiently rich to give "expected" denotations

$$\mathcal{F}[\![\iota]\!] = \mathsf{dom}(\mathcal{A}) \quad \mathcal{F}[\![o]\!] = \{0,1\} \quad \mathcal{F}[\![\tau \to \sigma]\!] \subseteq [\mathcal{F}[\![\tau]\!] \to \mathcal{F}[\![\sigma]\!]]$$

Satisfiability

Γ is *$(\mathcal{A},\mathcal{F})$-satisfiable* if there exists a *$(\Sigma',\mathcal{F})$-structure* $\mathcal{B}$ s.t.

Generalised HoCHC Satisfiability Problem

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

Γ: set of HoCHCs

$\mathcal{F}$: fixed interpretation of types s.t.

sufficiently rich to give "expected" denotations

$$\mathcal{F}[\![\iota]\!] = \mathrm{dom}(\mathcal{A}) \quad \mathcal{F}[\![o]\!] = \{0, 1\} \quad \mathcal{F}[\![\tau \to \sigma]\!] \subseteq [\mathcal{F}[\![\tau]\!] \to \mathcal{F}[\![\sigma]\!]]$$

Satisfiability

Γ is *$(\mathcal{A}, \mathcal{F})$-satisfiable* if there exists a *$(\Sigma', \mathcal{F})$-structure* $\mathcal{B}$ s.t.

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

Semantic Invariance

$\mathcal{A}$-monotone-satisfiable

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

$\Updownarrow$ [POPL'18]

$\mathcal{A}$-satisfiable

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

$(\mathcal{A}, \mathcal{F})$-satisfiable

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

Semantic Invariance

$\mathcal{A}$-monotone-satisfiable

$\Updownarrow$ [POPL'18]

$\mathcal{A}$-satisfiable

$\Uparrow$ soundness + completeness

$(\mathcal{A}, \mathcal{F})$-satisfiable

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

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

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

Semantic Invariance

$\mathcal{A}$-monotone-satisfiable

$\Updownarrow$ [POPL'18]

$\mathcal{A}$-satisfiable

$\Updownarrow$ soundness + completeness

$(\mathcal{A}, \mathcal{F})$-satisfiable

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

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

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

closed under suprema

Semantic Invariance

$\mathcal{A}$-monotone-satisfiable

$$\mathcal{M}\llbracket \tau \to \sigma \rrbracket \coloneqq [\mathcal{M}\llbracket \tau \rrbracket \xrightarrow{m} \mathcal{M}\llbracket \sigma \rrbracket]$$

⟺ [POPL'18]

$\mathcal{A}$-satisfiable

$$\mathcal{S}\llbracket \tau \to \sigma \rrbracket \coloneqq [\mathcal{S}\llbracket \tau \rrbracket \to \mathcal{S}\llbracket \sigma \rrbracket]$$

⟺ soundness + completeness

$(\mathcal{A}, \mathcal{F})$-satisfiable

$$\mathcal{F}\llbracket \tau \to \sigma \rrbracket \subseteq [\mathcal{F}\llbracket \tau \rrbracket \to \mathcal{F}\llbracket \sigma \rrbracket]$$

closed under suprema

$\mathcal{A}$-Henkin-satisfiable

$(\mathcal{A}, \mathcal{F}')$-satisfiable for *some* $\mathcal{F}'$

Semantic Invariance

$\mathcal{A}$-monotone-satisfiable

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

$\Updownarrow$ [POPL'18]

$\mathcal{A}$-satisfiable

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

$\Updownarrow$ soundness + completeness

$(\mathcal{A}, \mathcal{F})$-satisfiable

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

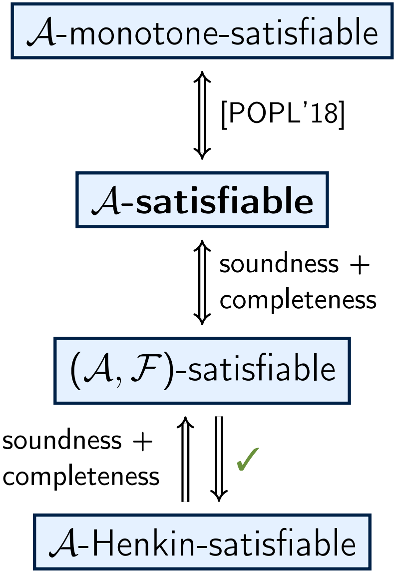
closed under suprema

$\Downarrow$ ✓

$\mathcal{A}$-Henkin-satisfiable

$(\mathcal{A}, \mathcal{F}')$-satisfiable for *some* $\mathcal{F}'$

Semantic Invariance



$\mathcal{A}$-monotone-satisfiable

⇕ [POPL'18]

$\mathcal{A}$-satisfiable

⇕ soundness + completeness

$(\mathcal{A}, \mathcal{F})$-satisfiable

soundness + completeness ⇑ ⇓ ✓

$\mathcal{A}$-Henkin-satisfiable

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

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

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

closed under suprema

$(\mathcal{A}, \mathcal{F}')$-satisfiable for *some* $\mathcal{F}'$

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

Conclusion

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

Luke Ong **Dominik Wagner**

`dominik.wagner@cs.ox.ac.uk`

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

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

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

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

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

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

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

$$\text{Res.}\ \frac{\neg(n \geq 1) \vee \neg\,\mathsf{Iter}\ \mathsf{Add}\ n\,n\,x \vee \neg(x \leq n + n) \qquad D_3}{\begin{array}{r}\neg(n \geq 1) \vee \neg(n > 0) \vee \neg\,\mathsf{Iter}\ \mathsf{Add}\ n\,(n-1)\,y \vee \\ \neg\,\mathsf{Add}\ n\,y\,x \vee \neg(x \leq n + n)\end{array}}$$

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

$$\text{Res.}\ \frac{\neg(n \geq 1) \vee \neg\,\mathsf{Iter}\ \mathsf{Add}\ n\, n\, x \ \vee \neg(x \leq n + n) \qquad D_3}{\begin{array}{r}\neg(n \geq 1) \vee \neg(n > 0) \vee \neg\,\mathsf{Iter}\ \mathsf{Add}\ n\,(n-1)\, y \ \vee \\ \neg\,\mathsf{Add}\ n\, y\, x \vee \neg(x \leq n + n)\end{array}}$$

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

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

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

$$\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\neg(n \geq 1) \vee \neg\,\text{Iter Add}\ n\,n\,x \ \vee \neg(x \leq n+n) \qquad D_3}{\begin{array}{r} \neg(n \geq 1) \vee \neg(n > 0) \vee \neg\,\text{Iter Add}\ n\,(n-1)\,y \ \vee \\ \neg\,\text{Add}\ n\,y\,x \ \vee \neg(x \leq n+n) \end{array}} \qquad D_1}{\begin{array}{r} \neg(n \geq 1) \vee \neg(n > 0) \vee \neg\,\text{Iter Add}\ n\,(n-1)\,y \ \vee \\ \neg(x = n+y) \ \vee \neg(x \leq n+n) \end{array}}$$

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

$$\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\neg(n \geq 1) \vee \neg\,\text{Iter Add}\ n\,n\,x \ \vee \neg(x \leq n+n) \qquad D_3}{\begin{gathered}\neg(n \geq 1) \vee \neg(n > 0) \vee \neg\,\text{Iter Add}\ n\,(n-1)\,y \ \vee\\ \neg\,\text{Add}\ n\,y\,x \ \vee \neg(x \leq n+n)\end{gathered}} \qquad D_1}{\begin{gathered}\neg(n \geq 1) \vee \neg(n > 0) \vee \neg\,\text{Iter Add}\ n\,(n-1)\,y \ \vee\\ \neg(x = n+y) \ \vee \neg(x \leq n+n)\end{gathered}} \qquad D_2}{\begin{gathered}\neg(n \geq 1) \ \vee\ \neg(n > 0) \ \vee\ \neg(n-1 \leq 0) \vee\\ \neg(n = y) \vee \neg(x = n+y) \vee \neg(x \leq n+n)\end{gathered}}$$

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

$$\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\text{Res.}\ \dfrac{\neg(n \ge 1) \lor \neg\,\text{Iter Add}\ n\,n\,x \ \lor \neg(x \le n+n) \qquad D_3}{\begin{array}{r}\neg(n \ge 1) \lor \neg(n > 0) \lor \neg\,\text{Iter Add}\ n\,(n-1)\,y\ \lor \\ \neg\,\text{Add}\ n\,y\,x \ \lor \neg(x \le n+n)\end{array}} \qquad D_1}{\begin{array}{r}\neg(n \ge 1) \lor \neg(n > 0) \lor \neg\,\text{Iter Add}\ n\,(n-1)\,y\ \lor \\ \neg(x = n+y) \ \lor \neg(x \le n+n)\end{array}} \qquad D_2}{\begin{array}{r}\neg(n \ge 1) \ \lor\ \neg(n > 0) \ \lor\ \neg(n-1 \le 0) \lor \\ \neg(n = y) \lor \neg(x = n+y) \lor \neg(x \le n+n)\end{array}}$$

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

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

$$
\text{Const. Ref.}\ \dfrac{
\text{Res.}\ \dfrac{
\text{Res.}\ \dfrac{
\text{Res.}\ \dfrac{\neg(n \geq 1) \lor \neg\,\text{Iter Add}\ n\,n\,x \ \lor \neg(x \leq n+n) \qquad D_3}{\begin{array}{r}\neg(n \geq 1) \lor \neg(n > 0) \lor \neg\,\text{Iter Add}\ n\,(n-1)\,y \ \lor \\ \neg\,\text{Add}\ n\,y\,x \ \lor \neg(x \leq n+n)\end{array}} \qquad D_1
}{\begin{array}{r}\neg(n \geq 1) \lor \neg(n > 0) \lor \neg\,\text{Iter Add}\ n\,(n-1)\,y \ \lor \\ \neg(x = n+y) \ \lor \neg(x \leq n+n)\end{array}} \qquad D_2
}{\begin{array}{r}\neg(n \geq 1) \lor \neg(n > 0) \lor \neg(n-1 \leq 0) \lor \\ \neg(n = y) \lor \neg(x = n+y) \lor \neg(x \leq n+n)\end{array}}
}{\bot}
$$

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