Undecidability, Incompleteness, and Completeness of Second-Order Logic in Coq

Certified Programs and Proofs

Mark Koch and Dominik Kirst

January 18, 2022

Saarland University, Programming Systems Lab

First-Order Logic

Quantification only over individuals

Second-Order Logic

Quantification over individuals & their properties

First-Order Logic

Quantification only over individuals

Second-Order Logic

Quantification over individuals & their properties

$$\forall P.\ P(0) \to (\forall n.\ P(n) \to P(n+1)) \to \forall n.\ P(n)$$

First-Order Logic

Quantification only over individuals

$$\varphi(0) \to (\forall n.\, \varphi(n) \to \varphi(n+1)) \to \forall n.\, \varphi(n)$$

for all formulas φ

Second-Order Logic

Quantification over individuals & their properties

$$\forall P.\, P(0) \to (\forall n.\, P(n) \to P(n+1)) \to \forall n.\, P(n)$$

First-Order Logic

Quantification only over individuals

$$\varphi(0) \to (\forall n.\, \varphi(n) \to \varphi(n+1)) \to \forall n.\, \varphi(n)$$

for all formulas φ

Second-Order Logic

Quantification over individuals & their properties

$$\forall P.\, P(0) \to (\forall n.\, P(n) \to P(n+1)) \to \forall n.\; P(n)$$

Behaviour of SOL depends on interpretation of second-order quantifiers:

First-Order Logic

Quantification only over individuals

$$\varphi(0) \to (\forall n.\, \varphi(n) \to \varphi(n+1)) \to \forall n.\, \varphi(n)$$

for all formulas φ

Second-Order Logic

Quantification over individuals & their properties

$$\forall P.\, P(0) \to (\forall n.\, P(n) \to P(n+1)) \to \forall n.\, P(n)$$

Behaviour of SOL depends on interpretation of second-order quantifiers:

- **Full semantics:** Quantifiers span the full relation space

First-Order Logic

Quantification only over individuals

$$\varphi(0) \to (\forall n.\, \varphi(n) \to \varphi(n+1)) \to \forall n.\, \varphi(n)$$

for all formulas φ

Second-Order Logic

Quantification over individuals & their properties

$$\forall P.\, P(0) \to (\forall n.\, P(n) \to P(n+1)) \to \forall n.\; P(n)$$

Behaviour of SOL depends on interpretation of second-order quantifiers:

- **Full semantics:** Quantifiers span the full relation space
  $\Rightarrow$ Only one PA_2 model, rules out completeness

First-Order Logic

Quantification only over individuals

$$\varphi(0) \to (\forall n.\, \varphi(n) \to \varphi(n+1)) \to \forall n.\, \varphi(n)$$

for all formulas φ

Second-Order Logic

Quantification over individuals & their properties

$$\forall P.\, P(0) \to (\forall n.\, P(n) \to P(n+1)) \to \forall n.\, P(n)$$

Behaviour of SOL depends on interpretation of second-order quantifiers:

- **Full semantics:** Quantifiers span the full relation space
  $\Rightarrow$ Only one PA_2 model, rules out completeness
- **Henkin semantics:** Generalises the relation space

First-Order Logic

Quantification only over individuals

$$\varphi(0) \to (\forall n.\, \varphi(n) \to \varphi(n+1)) \to \forall n.\, \varphi(n)$$

for all formulas φ

Second-Order Logic

Quantification over individuals & their properties

$$\forall P.\, P(0) \to (\forall n.\, P(n) \to P(n+1)) \to \forall n.\, P(n)$$

Behaviour of SOL depends on interpretation of second-order quantifiers:

- **Full semantics:** Quantifiers span the full relation space
  $\Rightarrow$ Only one PA_2 model, rules out completeness
- **Henkin semantics:** Generalises the relation space
  $\Rightarrow$ Recovers completeness and other meta-properties of FOL

First-Order Logic

Quantification only over individuals

$$\varphi(0) \to (\forall n.\, \varphi(n) \to \varphi(n+1)) \to \forall n.\, \varphi(n)$$

for all formulas φ

Second-Order Logic

Quantification over individuals & their properties

$$\forall P.\, P(0) \to (\forall n.\, P(n) \to P(n+1)) \to \forall n.\, P(n)$$

Behaviour of SOL depends on interpretation of second-order quantifiers:

- **Full semantics:** Quantifiers span the full relation space
  $\Rightarrow$ Only one PA_2 model, rules out completeness
- **Henkin semantics:** Generalises the relation space
  $\Rightarrow$ Recovers completeness and other meta-properties of FOL

Results well known (e.g. [Shapiro, 1991]). We analyse them in constructive type theory and mechanise them using the Coq proof assistant.

Mechanisation

Mechanisation

Given a signature $\Sigma = (\Sigma_{\mathcal{F}}, \Sigma_{\mathcal{P}})$, we inductively define

$$t ::= x_i \mid \mathcal{F}\ \vec{t} \qquad\qquad (\mathcal{F} : \Sigma_{\mathcal{F}})\ (i : \mathbb{N})$$

$$\varphi, \psi ::= \dot{\bot} \mid \mathcal{P}\ \vec{t} \mid p_i^n\ \vec{t} \mid \varphi \mathbin{\dot{\Box}} \psi \mid \dot{\nabla} \varphi \mid \dot{\nabla}_2^n \varphi \qquad (\mathcal{P} : \Sigma_{\mathcal{P}})\ (i, n : \mathbb{N})$$

Mechanisation

Given a signature $\Sigma = (\Sigma_{\mathcal{F}}, \Sigma_{\mathcal{P}})$, we inductively define

$$t ::= x_i \mid \mathcal{F}\ \vec{t} \qquad (\mathcal{F} : \Sigma_{\mathcal{F}})\ (i : \mathbb{N})$$

$$\varphi, \psi ::= \dot{\bot} \mid \mathcal{P}\ \vec{t} \mid p_i^n\ \vec{t} \mid \varphi \mathbin{\dot{\square}} \psi \mid \dot{\nabla}\varphi \mid \dot{\nabla}_2^n \varphi \qquad (\mathcal{P} : \Sigma_{\mathcal{P}})\ (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])

Mechanisation

Given a signature $\Sigma = (\Sigma_{\mathcal{F}}, \Sigma_{\mathcal{P}})$, we inductively define

$$t ::= x_i \mid \mathcal{F}\ \vec{t} \qquad (\mathcal{F} : \Sigma_{\mathcal{F}})\ (i : \mathbb{N})$$

$$\varphi, \psi ::= \dot{\bot} \mid \mathcal{P}\ \vec{t} \mid p_i^n\ \vec{t} \mid \varphi \mathbin{\dot{\square}} \psi \mid \dot{\nabla} \varphi \mid \dot{\nabla}_2^n \varphi \qquad (\mathcal{P} : \Sigma_{\mathcal{P}})\ (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  $\Rightarrow$ De Bruijn binders

Mechanisation

Given a signature $\Sigma = (\Sigma_{\mathcal{F}}, \Sigma_{\mathcal{P}})$, we inductively define

$$t ::= x_i \mid \mathcal{F}\ \vec{t} \qquad (\mathcal{F} : \Sigma_{\mathcal{F}})\ (i : \mathbb{N})$$

$$\varphi, \psi ::= \dot{\bot} \mid \mathcal{P}\ \vec{t} \mid p_i^n\ \vec{t} \mid \varphi\ \dot{\Box}\ \psi \mid \dot{\nabla}\varphi \mid \dot{\nabla}_2^n\ \varphi \qquad (\mathcal{P} : \Sigma_{\mathcal{P}})\ (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  $\Rightarrow$ De Bruijn binders, non-primitive equality

Mechanisation

Given a signature $\Sigma = (\Sigma_\mathcal{F}, \Sigma_\mathcal{P})$, we inductively define

$$t ::= x_i \mid \mathcal{F}\ \vec{t} \qquad (\mathcal{F} : \Sigma_\mathcal{F})\ (i : \mathbb{N})$$

$$\varphi, \psi ::= \dot{\bot} \mid \mathcal{P}\ \vec{t} \mid p_i^n\ \vec{t} \mid \varphi \mathbin{\dot{\square}} \psi \mid \dot{\nabla} \varphi \mid \dot{\nabla}_2^n \varphi \qquad (\mathcal{P} : \Sigma_\mathcal{P})\ (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  $\Rightarrow$ De Bruijn binders, non-primitive equality, type class for signatures

Mechanisation

Given a signature $\Sigma = (\Sigma_{\mathcal{F}}, \Sigma_{\mathcal{P}})$, we inductively define

$$t ::= x_i \mid \mathcal{F}\ \vec{t} \qquad (\mathcal{F} : \Sigma_{\mathcal{F}})\ (i : \mathbb{N})$$

$$\varphi, \psi ::= \dot{\bot} \mid \mathcal{P}\ \vec{t} \mid p_i^n\ \vec{t} \mid \varphi \dot{\square} \psi \mid \dot{\nabla} \varphi \mid \dot{\nabla}_2^n \varphi \qquad (\mathcal{P} : \Sigma_{\mathcal{P}})\ (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  $\Rightarrow$ De Bruijn binders, non-primitive equality, type class for signatures
- HOL mechanisations available (e.g. [Harrison, 2006, Kumar et al., 2016])

Mechanisation

Given a signature $\Sigma = (\Sigma_{\mathcal{F}}, \Sigma_{\mathcal{P}})$, we inductively define

$$t ::= x_i \mid \mathcal{F}\ \vec{t} \qquad (\mathcal{F} : \Sigma_{\mathcal{F}})\ (i : \mathbb{N})$$

$$\varphi, \psi ::= \dot{\bot} \mid \mathcal{P}\ \vec{t} \mid p_i^n\ \vec{t} \mid \varphi \dot{\Box} \psi \mid \dot{\nabla} \varphi \mid \dot{\nabla}_2^n \varphi \qquad (\mathcal{P} : \Sigma_{\mathcal{P}})\ (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  $\Rightarrow$ De Bruijn binders, non-primitive equality, type class for signatures
- HOL mechanisations available (e.g. [Harrison, 2006, Kumar et al., 2016]), but no previous work on SOL

Mechanisation

Given a signature $\Sigma = (\Sigma_{\mathcal{F}}, \Sigma_{\mathcal{P}})$, we inductively define

$$t ::= x_i \mid \mathcal{F}\ \vec{t} \qquad (\mathcal{F} : \Sigma_{\mathcal{F}})\ (i : \mathbb{N})$$

$$\varphi, \psi ::= \dot{\bot} \mid \mathcal{P}\ \vec{t} \mid p_i^n\ \vec{t} \mid \varphi \mathbin{\dot{\Box}} \psi \mid \dot{\nabla}\varphi \mid \dot{\nabla}_2^n\,\varphi \qquad (\mathcal{P} : \Sigma_{\mathcal{P}})\ (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  $\Rightarrow$ De Bruijn binders, non-primitive equality, type class for signatures
- HOL mechanisations available (e.g. [Harrison, 2006, Kumar et al., 2016]), but no previous work on SOL
- Unique challenges of SOL: arities

Mechanisation

Given a signature $\Sigma = (\Sigma_{\mathcal{F}}, \Sigma_{\mathcal{P}})$, we inductively define

$$t ::= x_i \mid \mathcal{F}\ \vec{t} \qquad\qquad (\mathcal{F} : \Sigma_{\mathcal{F}})\ (i : \mathbb{N})$$

$$\varphi, \psi ::= \dot{\bot} \mid \mathcal{P}\ \vec{t} \mid p_i^n\ \vec{t} \mid \varphi \mathbin{\dot{\square}} \psi \mid \dot{\nabla} \varphi \mid \dot{\nabla}_2^n \varphi \qquad (\mathcal{P} : \Sigma_{\mathcal{P}})\ (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  $\Rightarrow$ De Bruijn binders, non-primitive equality, type class for signatures
- HOL mechanisations available (e.g. [Harrison, 2006, Kumar et al., 2016]), but no previous work on SOL
- Unique challenges of SOL: arities, function quantifiers

Full Semantics: Undecidability and Incompleteness

Full Semantics

Definition (Full Semantics)

- A model $\mathcal{M}$ consists of a domain D and interpretations $\mathcal{F}^{\mathcal{M}} : D^{|\mathcal{F}|} \to D$ and $\mathcal{P}^{\mathcal{M}} : D^{|\mathcal{P}|} \to \text{Prop}$.

Full Semantics

Definition (Full Semantics)

- A model $\mathcal{M}$ consists of a domain D and interpretations $\mathcal{F}^{\mathcal{M}} : D^{|\mathcal{F}|} \to D$ and $\mathcal{P}^{\mathcal{M}} : D^{|\mathcal{P}|} \to \mathrm{Prop}$.
- Interpretation ($\vDash$) in $\mathcal{M}$ maps connectives $\Box$ and quantifiers ∇ to their counterparts in Prop.

Full Semantics

Definition (Full Semantics)

- A model $\mathcal{M}$ consists of a domain D and interpretations $\mathcal{F}^{\mathcal{M}} : D^{|\mathcal{F}|} \to D$ and $\mathcal{P}^{\mathcal{M}} : D^{|\mathcal{P}|} \to \text{Prop}$.
- Interpretation ($\vDash$) in $\mathcal{M}$ maps connectives $\Box$ and quantifiers ∇ to their counterparts in Prop.
- SOL quantifiers ∇_2^n range over the full relation space $D^n \to \text{Prop}$.

Second-Order Peano Arithmetic

Zero Addition : $\dot\forall x.\, O + x \equiv x$

Addition Recursion : $\dot\forall xy.\, (Sx) + y \equiv S(x + y)$

Disjointness : $\dot\forall x.\, O \equiv Sx \dot\to \dot\bot$

Equality Reflexivity : $\dot\forall x.\, x \equiv x$

Zero Multiplication : $\dot\forall x.\, O \cdot x \equiv O$

Multiplication Recursion : $\dot\forall xy.\, (Sx) \cdot y \equiv y + x \cdot y$

Successor Injectivity : $\dot\forall xy.\, Sx \equiv Sy \dot\to x \equiv y$

Equality Symmetry : $\dot\forall xy.\, x \equiv y \dot\to y \equiv x$

Induction : $\dot\forall P.\, P(O) \dot\to (\dot\forall x.\, P(x) \dot\to P(Sx)) \dot\to \dot\forall x.\, P(x)$

Second-Order Peano Arithmetic

Zero Addition : $\dot\forall x.\, O + x \equiv x$

Addition Recursion : $\dot\forall xy.\, (Sx) + y \equiv S(x + y)$

Disjointness : $\dot\forall x.\, O \equiv Sx \mathrel{\dot\rightarrow} \dot\bot$

Equality Reflexivity : $\dot\forall x.\, x \equiv x$

Zero Multiplication : $\dot\forall x.\, O \cdot x \equiv O$

Multiplication Recursion : $\dot\forall xy.\, (Sx) \cdot y \equiv y + x \cdot y$

Successor Injectivity : $\dot\forall xy.\, Sx \equiv Sy \mathrel{\dot\rightarrow} x \equiv y$

Equality Symmetry : $\dot\forall xy.\, x \equiv y \mathrel{\dot\rightarrow} y \equiv x$

Induction : $\dot\forall P.\, P(O) \mathrel{\dot\rightarrow} (\dot\forall x.\, P(x) \mathrel{\dot\rightarrow} P(Sx)) \mathrel{\dot\rightarrow} \dot\forall x.\, P(x)$

Theorem (Categoricity [Dedekind, 1888, Shapiro, 1991])

PA_2 is categorical for full semantics, i.e. all models of PA_2 are isomorphic.

Second-Order Peano Arithmetic

Zero Addition : $\dot\forall x.\, O + x \equiv x$

Addition Recursion : $\dot\forall xy.\, (Sx) + y \equiv S(x + y)$

Disjointness : $\dot\forall x.\, O \equiv Sx \dot\to \dot\bot$

Equality Reflexivity : $\dot\forall x.\, x \equiv x$

Zero Multiplication : $\dot\forall x.\, O \cdot x \equiv O$

Multiplication Recursion : $\dot\forall xy.\, (Sx) \cdot y \equiv y + x \cdot y$

Successor Injectivity : $\dot\forall xy.\, Sx \equiv Sy \dot\to x \equiv y$

Equality Symmetry : $\dot\forall xy.\, x \equiv y \dot\to y \equiv x$

Induction : $\dot\forall P.\, P(O) \dot\to (\dot\forall x.\, P(x) \dot\to P(Sx)) \dot\to \dot\forall x.\, P(x)$

Theorem (Categoricity [Dedekind, 1888, Shapiro, 1991])

PA_2 is categorical for full semantics, i.e. all models of PA_2 are isomorphic.

Proof.

Given models $\mathcal{M}_1, \mathcal{M}_2 \vDash PA_2$, inductively define $\cong : D_1 \to D_2 \to \text{Prop}$

$$O^{\mathcal{M}_1} \cong O^{\mathcal{M}_2} \qquad\qquad S^{\mathcal{M}_1}\, x \cong S^{\mathcal{M}_2}\, y \quad \text{if } x \cong y.$$

Second-Order Peano Arithmetic

Zero Addition : $\dot\forall x.\, O + x \equiv x$

Addition Recursion : $\dot\forall xy.\, (Sx) + y \equiv S(x + y)$

Disjointness : $\dot\forall x.\, O \equiv Sx \dot\to \dot\bot$

Equality Reflexivity : $\dot\forall x.\, x \equiv x$

Zero Multiplication : $\dot\forall x.\, O \cdot x \equiv O$

Multiplication Recursion : $\dot\forall xy.\, (Sx) \cdot y \equiv y + x \cdot y$

Successor Injectivity : $\dot\forall xy.\, Sx \equiv Sy \dot\to x \equiv y$

Equality Symmetry : $\dot\forall xy.\, x \equiv y \dot\to y \equiv x$

Induction : $\dot\forall P.\, P(O) \dot\to (\dot\forall x.\, P(x) \dot\to P(Sx)) \dot\to \dot\forall x.\, P(x)$

Theorem (Categoricity [Dedekind, 1888, Shapiro, 1991])

PA_2 is categorical for full semantics, i.e. all models of PA_2 are isomorphic.

Proof.

Given models $\mathcal{M}_1, \mathcal{M}_2 \vDash PA_2$, inductively define $\cong\, : D_1 \to D_2 \to \text{Prop}$

$$O^{\mathcal{M}_1} \cong O^{\mathcal{M}_2} \qquad\qquad S^{\mathcal{M}_1}\, x \cong S^{\mathcal{M}_2}\, y \quad \text{if } x \cong y.$$

Verify that $\cong$ is an isomorphism using the induction axiom. □

Corollary (Failure of Löwenheim-Skolem)

SOL does not have the Löwenheim-Skolem property for full semantics.

Consequences of Categoricity

Corollary (Failure of Löwenheim-Skolem)

SOL does not have the Löwenheim-Skolem property for full semantics.

Theorem (Failure of Compactness)

SOL is not compact for full semantics.

Consequences of Categoricity

Corollary (Failure of Löwenheim-Skolem)

SOL does not have the Löwenheim-Skolem property for full semantics.

Theorem (Failure of Compactness)

SOL is not compact for full semantics.

Proof.

Consider the theory $\mathcal{T}_{\neq} := \mathsf{PA}_2, x \neq O, x \neq S\,O, x \neq S\,(S\,O), \ldots$

Consequences of Categoricity

Corollary (Failure of Löwenheim-Skolem)

SOL does not have the Löwenheim-Skolem property for full semantics.

Theorem (Failure of Compactness)

SOL is not compact for full semantics.

Proof.

Consider the theory $\mathcal{T}_{\neq} := \mathrm{PA}_2, x \neq O, x \neq S\,O, x \neq S\,(S\,O), \ldots$

- Every finite subset of $\mathcal{T}_{\neq}$ has a model, for example $\mathbb{N}$.

Consequences of Categoricity

Corollary (Failure of Löwenheim-Skolem)

SOL does not have the Löwenheim-Skolem property for full semantics.

Theorem (Failure of Compactness)

SOL is not compact for full semantics.

Proof.

Consider the theory $\mathcal{T}_{\neq} := \mathrm{PA}_2, x \neq O, x \neq S\,O, x \neq S\,(S\,O), \ldots$

- Every finite subset of $\mathcal{T}_{\neq}$ has a model, for example $\mathbb{N}$.
- But $\mathbb{N}$ is not model of the whole theory $\mathcal{T}_{\neq}$.

Corollary (Failure of Löwenheim-Skolem)

SOL does not have the Löwenheim-Skolem property for full semantics.

Theorem (Failure of Compactness)

SOL is not compact for full semantics.

Proof.

Consider the theory $\mathcal{T}_{\neq} := \mathrm{PA}_2, x \neq O, x \neq S\,O, x \neq S\,(S\,O), \ldots$

- Every finite subset of $\mathcal{T}_{\neq}$ has a model, for example $\mathbb{N}$.
- But $\mathbb{N}$ is not model of the whole theory $\mathcal{T}_{\neq}$. Since $\mathbb{N}$ is the only model of PA_2, we can conclude that $\mathcal{T}_{\neq}$ does not have a model. □

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Deduction system $\vdash : \mathcal{L}(\text{form}) \to \text{form} \to \text{Prop}$

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Deduction system $\vdash : \mathcal{L}(\text{form}) \to \text{form} \to \text{Prop}$

- Completeness: $\Gamma \vDash \varphi \to \Gamma \vdash \varphi$ for all lists Γ

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Deduction system $\vdash \, : \mathcal{L}(\mathsf{form}) \to \mathsf{form} \to \mathsf{Prop}$

- Completeness: $\Gamma \vDash \varphi \to \Gamma \vdash \varphi$ for all lists Γ
- Lift $\vdash$ to theories: $\mathcal{T} \vdash \varphi := \exists \Gamma \subseteq_{\mathsf{fin}} \mathcal{T}.\, \Gamma \vdash \varphi$

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Deduction system $\vdash \,:\, \mathcal{L}(\mathsf{form}) \to \mathsf{form} \to \mathsf{Prop}$

- Completeness: $\Gamma \vDash \varphi \to \Gamma \vdash \varphi$ for all lists Γ
- Lift $\vdash$ to theories: $\mathcal{T} \vdash \varphi := \exists \Gamma \subseteq_{\mathsf{fin}} \mathcal{T}.\, \Gamma \vdash \varphi$
- Strong completeness: $\mathcal{T} \vDash \varphi \to \mathcal{T} \vdash \varphi$

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Deduction system $\vdash : \mathcal{L}(\mathsf{form}) \to \mathsf{form} \to \mathsf{Prop}$

- Completeness: $\Gamma \vDash \varphi \to \Gamma \vdash \varphi$ for all lists Γ
- Lift $\vdash$ to theories: $\mathcal{T} \vdash \varphi := \exists \Gamma \subseteq_{\mathsf{fin}} \mathcal{T}.\, \Gamma \vdash \varphi$
- Strong completeness: $\mathcal{T} \vDash \varphi \to \mathcal{T} \vdash \varphi$

No computability assumptions on $\vdash$

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Does not rule out the weaker notion of completeness: $\Gamma \vDash \varphi \rightarrow \Gamma \vdash \varphi$

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Does not rule out the weaker notion of completeness: $\Gamma \vDash \varphi \to \Gamma \vdash \varphi$

- Requires more involved proof + assumption that $\vdash$ is enumerable

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Does not rule out the weaker notion of completeness: $\Gamma \vDash \varphi \rightarrow \Gamma \vdash \varphi$

- Requires more involved proof + assumption that $\vdash$ is enumerable
- Usually given as a consequence of Gödel's first incompleteness theorem

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Does not rule out the weaker notion of completeness: $\Gamma \vDash \varphi \rightarrow \Gamma \vdash \varphi$

- Requires more involved proof + assumption that $\vdash$ is enumerable
- Usually given as a consequence of Gödel's first incompleteness theorem

We argue via computability theory [Kleene, 1952, Kirst and Hermes, 2021],

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Does not rule out the weaker notion of completeness: $\Gamma \vDash \varphi \rightarrow \Gamma \vdash \varphi$

- Requires more involved proof + assumption that $\vdash$ is enumerable
- Usually given as a consequence of Gödel's first incompleteness theorem

We argue via computability theory [Kleene, 1952, Kirst and Hermes, 2021], using the synthetic approach [Richman, 1983, Bauer, 2006, Forster et al., 2019]:

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Does not rule out the weaker notion of completeness: $\Gamma \vDash \varphi \rightarrow \Gamma \vdash \varphi$

- Requires more involved proof + assumption that $\vdash$ is enumerable
- Usually given as a consequence of Gödel's first incompleteness theorem

We argue via computability theory [Kleene, 1952, Kirst and Hermes, 2021], using the synthetic approach [Richman, 1983, Bauer, 2006, Forster et al., 2019]:

- P is undecidable if an undecidable problem reduces to it (e.g. Halt).

Consequences of Categoricity

Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

Does not rule out the weaker notion of completeness: $\Gamma \vDash \varphi \rightarrow \Gamma \vdash \varphi$

- Requires more involved proof + assumption that $\vdash$ is enumerable
- Usually given as a consequence of Gödel's first incompleteness theorem

We argue via computability theory [Kleene, 1952, Kirst and Hermes, 2021], using the synthetic approach [Richman, 1983, Bauer, 2006, Forster et al., 2019]:

- P is undecidable if an undecidable problem reduces to it (e.g. Halt).
- P not enumerable if a not enumerable problem reduces to it (e.g. $\overline{\text{Halt}}$).

Incompleteness

Lemma

Validity in PA_2 is not enumerable.

Lemma

Validity in PA_2 is not enumerable.

Proof Sketch.

Via reduction from the complement of Hilbert's tenth problem $\overline{H_{10}}$:[1]

[1]Whose undecidability [Davis and Putnam, 1959, Robinson, 1952, Matijasevič, 1971] has already been mechanised in Coq [Larchey-Wendling and Forster, 2019].

Lemma

Validity in PA_2 is not enumerable.

Proof Sketch.

Via reduction from the complement of Hilbert's tenth problem $\overline{H_{10}}$:[1]

$$\underbrace{x + 2}_{s} = \underbrace{y^2 + z}_{t}$$

[1] Whose undecidability [Davis and Putnam, 1959, Robinson, 1952, Matijasevič, 1971] has already been mechanised in Coq [Larchey-Wendling and Forster, 2019].

Incompleteness

Lemma

Validity in PA_2 is not enumerable.

Proof Sketch.

Via reduction from the complement of Hilbert's tenth problem $\overline{H_{10}}$:[1]

$$\underbrace{x + 2}_{s} = \underbrace{y^2 + z}_{t} \quad \rightsquigarrow \quad \varphi_{s,t} := \dot{\exists} xyz.\, x + S\,(S\,O) \equiv y \cdot y + z$$

[1] Whose undecidability [Davis and Putnam, 1959, Robinson, 1952, Matijasevič, 1971] has already been mechanised in Coq [Larchey-Wendling and Forster, 2019].

Lemma

Validity in PA_2 is not enumerable.

Proof Sketch.

Via reduction from the complement of Hilbert's tenth problem $\overline{\mathrm{H}_{10}}$:[1]

$$\underbrace{x+2}_{s} = \underbrace{y^2+z}_{t} \quad \rightsquigarrow \quad \varphi_{s,t} := \dot{\exists} xyz.\, x + S\,(S\,O) \equiv y \cdot y + z$$

$s = t$ is unsolvable iff $\mathbb{N} \vDash \dot{\neg}\varphi_{s,t}$

[1] Whose undecidability [Davis and Putnam, 1959, Robinson, 1952, Matijasevič, 1971] has already been mechanised in Coq [Larchey-Wendling and Forster, 2019].

Lemma

Validity in PA_2 is not enumerable.

Proof Sketch.

Via reduction from the complement of Hilbert's tenth problem $\overline{H_{10}}$:[1]

$$\underbrace{x + 2}_{s} = \underbrace{y^2 + z}_{t} \quad \rightsquigarrow \quad \varphi_{s,t} := \dot{\exists} xyz.\, x + S\,(S\,O) \equiv y \cdot y + z$$

$s = t$ is unsolvable iff $\mathbb{N} \vDash \dot{\neg}\varphi_{s,t}$ and thus iff $PA_2 \vDash \dot{\neg}\varphi_{s,t}$ by categoricity.

[1] Whose undecidability [Davis and Putnam, 1959, Robinson, 1952, Matijasevič, 1971] has already been mechanised in Coq [Larchey-Wendling and Forster, 2019].

Incompleteness

Theorem (Incompleteness)

SOL is not complete for full semantics

Theorem (Incompleteness)

SOL is not complete for full semantics, i.e. the existence of a sound, enumerable and complete deduction system implies enumerability of $\overline{\mathsf{H}_{10}}$.

Incompleteness

Theorem (Incompleteness)

SOL is not complete for full semantics, i.e. the existence of a sound, enumerable and complete deduction system implies enumerability of $\overline{\mathsf{H}_{10}}$.

Theorem (Undecidability)

Second-order validity and satisfiability in the empty signature are undecidable.

Incompleteness

Theorem (Incompleteness)

SOL is not complete for full semantics, i.e. the existence of a sound, enumerable and complete deduction system implies enumerability of $\overline{\mathsf{H}_{10}}$.

Theorem (Undecidability)

Second-order validity and satisfiability in the empty signature are undecidable.

Proof Sketch.

- $s = t$ has a solution iff $\dot\forall\, O\, S + \times \equiv .\ \mathsf{PA}_2 \dot\to \varphi_{s,t}$ is valid.

Incompleteness

Theorem (Incompleteness)

SOL is not complete for full semantics, i.e. the existence of a sound, enumerable and complete deduction system implies enumerability of $\overline{\mathsf{H}_{10}}$.

Theorem (Undecidability)

Second-order validity and satisfiability in the empty signature are undecidable.

Proof Sketch.

- $s = t$ has a solution iff $\dot\forall\, O\, S + \times \equiv .\ \mathsf{PA}_2 \dot\rightarrow \varphi_{s,t}$ is valid.
- $s = t$ has a solution iff $\dot\exists\, O\, S + \times \equiv .\ \mathsf{PA}_2 \dot\wedge \varphi_{s,t}$ is satisfiable. □

Henkin Semantics: Completeness

Henkin Semantics

Definition (Henkin Semantics).

- Second-order quantifiers ∇_2^n only range over the relations contained in a universe $\mathbb{U}_n : (D^n \to \mathsf{Prop}) \to \mathsf{Prop}$.

Henkin Semantics

Definition (Henkin Semantics).

- Second-order quantifiers ∇_2^n only range over the relations contained in a universe $\mathbb{U}_n : (D^n \to \mathsf{Prop}) \to \mathsf{Prop}$.
- $\mathbb{U}_n$ is specified by a Henkin model $\mathcal{H}$.

Definition (Henkin Semantics).

- Second-order quantifiers ∇_2^n only range over the relations contained in a universe $\mathbb{U}_n : (D^n \to \mathsf{Prop}) \to \mathsf{Prop}$.
- $\mathbb{U}_n$ is specified by a Henkin model $\mathcal{H}$.
- $\mathbb{U}_n$ should satisfy comprehension, i.e. it must at least contain all second-order definable properties.

Definition (Henkin Semantics).

- Second-order quantifiers ∇_2^n only range over the relations contained in a universe $\mathbb{U}_n : (D^n \to \text{Prop}) \to \text{Prop}$.
- $\mathbb{U}_n$ is specified by a Henkin model $\mathcal{H}$.
- $\mathbb{U}_n$ should satisfy comprehension, i.e. it must at least contain all second-order definable properties.

The second-order ND system $\vdash_2$ is obtained by extending the first-order system $\vdash_1$ with rules for second-order quantifiers and comprehension:

$$\frac{}{A \vdash_2 \dot{\exists} P. \dot{\forall} x_1 \ldots x_n.\, P(x_1, \ldots, x_2) \dot{\leftrightarrow} \varphi}\ \text{Compr}_\varphi$$

Definition (Henkin Semantics).

- Second-order quantifiers ∇_2^n only range over the relations contained in a universe $\mathbb{U}_n : (D^n \rightarrow \text{Prop}) \rightarrow \text{Prop}$.
- $\mathbb{U}_n$ is specified by a Henkin model $\mathcal{H}$.
- $\mathbb{U}_n$ should satisfy comprehension, i.e. it must at least contain all second-order definable properties.

The second-order ND system $\vdash_2$ is obtained by extending the first-order system $\vdash_1$ with rules for second-order quantifiers and comprehension:

$$\frac{}{A \vdash_2 \dot{\exists} P.\, \dot{\forall} x_1 \ldots x_n.\, P(x_1, \ldots, x_2) \dot{\leftrightarrow} \varphi}\ \text{Compr}_\varphi$$

$\vdash_2$ is complete for Henkin semantics [Henkin, 1949].

Connection to FOL

SOL with Henkin semantics is essentially just many-sorted FOL:

Connection to FOL

SOL with Henkin semantics is essentially just many-sorted FOL:

$$\varphi := \dot{\forall} x.\, \dot{\exists} P.\, P(x, x) \quad \rightsquigarrow \quad \varphi^\star := \dot{\forall} x^{\mathcal{I}}.\, \dot{\exists} p^{\mathcal{P}_2}.\, \mathsf{App}_2(p, x, x)$$

Connection to FOL

SOL with Henkin semantics is essentially just many-sorted FOL:

$$\varphi := \dot{\forall} x.\, \dot{\exists} P.\, P(x, x) \quad \rightsquigarrow \quad \varphi^\star := \dot{\forall} x^{\mathcal{I}}.\, \dot{\exists} p^{\mathcal{P}_2}.\, \mathrm{App}_2(p, x, x)$$

$\Rightarrow$ We can transport the completeness theorem from FOL to SOL

Connection to FOL

SOL with Henkin semantics is essentially just many-sorted FOL:

$$\varphi := \dot{\forall} x.\, \dot{\exists} P.\, P(x,x) \quad \rightsquigarrow \quad \varphi^\star := \dot{\forall} x^{\mathcal{I}}.\, \dot{\exists} p^{\mathcal{P}_2}.\, \mathsf{App}_2(p,x,x)$$

$$\varphi^\star := \dot{\forall} x.\, \mathsf{isIndi}(x) \mathbin{\dot{\to}} \dot{\exists} p.\, \mathsf{isPred}_2(p) \mathbin{\dot{\wedge}} \mathsf{App}_2(p,x,x)$$

Guard the quantifiers with predicates to distinguish the sorts [Van Dalen, 1994].

$\Rightarrow$ We can transport the completeness theorem from FOL to SOL

Connection to FOL

SOL with Henkin semantics is essentially just many-sorted FOL:

$$\varphi := \dot{\forall} x.\, \dot{\exists} P.\, P(x, x) \quad \rightsquigarrow \quad \varphi^\star := \dot{\forall} x^{\mathcal{I}}.\, \dot{\exists} p^{\mathcal{P}_2}.\, \mathsf{App}_2(p, x, x)$$

$$\varphi^\star := \dot{\forall} x.\, \mathsf{isIndi}(x) \mathbin{\dot{\to}} \dot{\exists} p.\, \mathsf{isPred}_2(p) \mathbin{\dot{\wedge}} \mathsf{App}_2(p, x, x)$$

Guard the quantifiers with predicates to distinguish the sorts [Van Dalen, 1994].

However, difficult to prove $\vdash_1 \varphi^\star \to\ \vdash_2 \varphi$.

$\Rightarrow$ We can transport the completeness theorem from FOL to SOL

Connection to FOL

SOL with Henkin semantics is essentially just many-sorted FOL:

$$\varphi := \dot{\forall} x.\, \dot{\exists} P.\, P(x,x) \quad \rightsquigarrow \quad \varphi^\star := \dot{\forall} x^{\mathcal{I}}.\, \dot{\exists} p^{\mathcal{P}_2}.\, \mathsf{App}_2(p,x,x)$$

$$\varphi^\star := \dot{\forall} x.\, \mathsf{isIndi}(x) \mathbin{\dot{\to}} \dot{\exists} p.\, \mathsf{isPred}_2(p) \mathbin{\dot{\wedge}} \mathsf{App}_2(p,x,x)$$

Guard the quantifiers with predicates to distinguish the sorts [Van Dalen, 1994].

However, difficult to prove $\vdash_1 \varphi^\star \to\ \vdash_2 \varphi$. [Nour and Raffalli, 2003] propose:

$\Rightarrow$ We can transport the completeness theorem from FOL to SOL

Connection to FOL

SOL with Henkin semantics is essentially just many-sorted FOL:

$$\varphi := \dot\forall x.\, \dot\exists P.\, P(x,x) \quad \rightsquigarrow \quad \varphi^\star := \dot\forall x^{\mathcal{I}}.\, \dot\exists p^{\mathcal{P}_2}.\, \mathsf{App}_2(p,x,x)$$

$$\varphi^\star := \dot\forall x.\, \mathsf{isIndi}(x) \mathrel{\dot\rightarrow} \dot\exists p.\, \mathsf{isPred}_2(p) \mathrel{\dot\wedge} \mathsf{App}_2(p,x,x)$$

Guard the quantifiers with predicates to distinguish the sorts [Van Dalen, 1994].

However, difficult to prove $\vdash_1 \varphi^\star \rightarrow\ \vdash_2 \varphi$. [Nour and Raffalli, 2003] propose:

$$\varphi^\star := \dot\forall x.\, \dot\exists p.\, \mathsf{App}_2(p,x,x)$$

$\Rightarrow$ We can transport the completeness theorem from FOL to SOL

Connection to FOL

SOL with Henkin semantics is essentially just many-sorted FOL:

$$\varphi := \dot\forall x.\, \dot\exists P.\, P(x,x) \quad \rightsquigarrow \quad \varphi^\star := \dot\forall x^{\mathcal{I}}.\, \dot\exists p^{\mathcal{P}_2}.\, \mathrm{App}_2(p,x,x)$$

$$\varphi^\star := \dot\forall x.\, \mathrm{isIndi}(x) \mathrel{\dot\to} \dot\exists p.\, \mathrm{isPred}_2(p) \mathrel{\dot\wedge} \mathrm{App}_2(p,x,x)$$

Guard the quantifiers with predicates to distinguish the sorts [Van Dalen, 1994].

However, difficult to prove $\vdash_1 \varphi^\star \to \ \vdash_2 \varphi$. [Nour and Raffalli, 2003] propose:

$$\varphi^\star := \dot\forall x.\, \dot\exists p.\, \mathrm{App}_2(p,x,x)$$

x and p represent individuals and predicates at the same time.

$\Rightarrow$ We can transport the completeness theorem from FOL to SOL

Reduction to FOL [Nour and Raffalli, 2003]

$$\mathcal{T} \models_2 \varphi$$

$$\updownarrow$$

$$(\mathcal{T} \cup \mathsf{Compr})^\star \models_1 \varphi^\star$$

Reduction to FOL [Nour and Raffalli, 2003]

$$\mathcal{T} \vDash_2 \varphi$$

$$\updownarrow$$

$$(\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star \dashrightarrow (\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star$$

FOL completeness (MP / LEM)
[Forster et al., 2021]

Reduction to FOL [Nour and Raffalli, 2003]

$\mathcal{T} \vDash_2 \varphi$ $\longleftrightarrow$ $(\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star$

$(\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star \dashrightarrow (\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star$

FOL completeness (MP / LEM)
[Forster et al., 2021]

$(\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star \longrightarrow \mathcal{T} \vdash_2 \varphi$

Reduction to FOL [Nour and Raffalli, 2003]

$$\begin{array}{ccc} \mathcal{T} \vDash_2 \varphi & & \mathcal{T} \vdash_2 \varphi \\ \updownarrow & & \uparrow \\ (\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star & \xrightarrow[\text{[Forster et al., 2021]}]{\text{FOL completeness (MP / LEM)}} & (\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star \end{array}$$

Reduction to FOL [Nour and Raffalli, 2003]

$\mathcal{T} \vDash_2 \varphi$ $\qquad\qquad\qquad\qquad$ $\mathcal{T} \vdash_2 \varphi$

$\updownarrow$ $\qquad\qquad\qquad\qquad\qquad\qquad$ $\uparrow$

FOL completeness (MP / LEM)
[Forster et al., 2021]

$(\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star \dashrightarrow (\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star$

Define FOL to SOL translation $\_^\diamond$ that satisfies

Reduction to FOL [Nour and Raffalli, 2003]

$$\mathcal{T} \vDash_2 \varphi \qquad\qquad\qquad\qquad \mathcal{T} \vdash_2 \varphi$$

$$\updownarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

FOL completeness (MP / LEM)
[Forster et al., 2021]

$$(\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star \dashrightarrow (\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star$$

Define FOL to SOL translation $\_^\diamond$ that satisfies

$$\vdash_2 \varphi^{\star\diamond} \mathrel{\dot{\leftrightarrow}} \varphi$$

Reduction to FOL [Nour and Raffalli, 2003]

$$\mathcal{T} \vDash_2 \varphi \qquad\qquad \mathcal{T} \vdash_2 \varphi$$

FOL completeness (MP / LEM)
[Forster et al., 2021]

$$(\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star \dashrightarrow (\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star$$

Define FOL to SOL translation $\_^\diamond$ that satisfies

$$\vdash_2 \varphi^{\star\diamond} \dot{\leftrightarrow} \varphi \qquad\qquad A \vdash_1 \varphi \to A^\diamond \vdash_2 \varphi^\diamond$$

Reduction to FOL [Nour and Raffalli, 2003]

$$\mathcal{T} \vDash_2 \varphi \qquad\qquad\qquad\qquad \mathcal{T} \vdash_2 \varphi$$

$$\updownarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

FOL completeness (MP / LEM)
[Forster et al., 2021]

$$(\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star \dashrightarrow (\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star$$

Define FOL to SOL translation $\_^\diamond$ that satisfies

$$\vdash_2 \varphi^{\star\diamond} \dot{\leftrightarrow} \varphi \qquad\qquad A \vdash_1 \varphi \rightarrow A^\diamond \vdash_2 \varphi^\diamond$$

Theorem (Relative Completeness)

If FOL is complete, then so is SOL with Henkin semantics.

Reduction to FOL [Nour and Raffalli, 2003]

$$\mathcal{T} \vDash_2 \varphi \qquad\qquad\qquad\qquad \mathcal{T} \vdash_2 \varphi$$

$$\updownarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

FOL completeness (MP / LEM)
[Forster et al., 2021]

$$(\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star \dashrightarrow (\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star$$

Define FOL to SOL translation $\_^\diamond$ that satisfies

$$\vdash_2 \varphi^{\star\diamond} \mathrel{\dot\leftrightarrow} \varphi \qquad\qquad A \vdash_1 \varphi \to A^\diamond \vdash_2 \varphi^\diamond$$

Theorem (Completeness)

SOL with Henkin semantics is complete under LEM.

Reduction to FOL [Nour and Raffalli, 2003]

$$\mathcal{T} \vDash_2 \varphi \qquad\qquad\qquad\qquad \mathcal{T} \vdash_2 \varphi$$

$$\updownarrow \qquad\qquad\qquad\qquad\qquad \uparrow$$

FOL completeness (MP / LEM)
[Forster et al., 2021]

$$(\mathcal{T} \cup \mathsf{Compr})^\star \vDash_1 \varphi^\star \dashrightarrow (\mathcal{T} \cup \mathsf{Compr})^\star \vdash_1 \varphi^\star$$

Define FOL to SOL translation $\_^\diamond$ that satisfies

$$\vdash_2 \varphi^{\star\diamond} \dot{\leftrightarrow} \varphi \qquad\qquad A \vdash_1 \varphi \to A^\diamond \vdash_2 \varphi^\diamond$$

Theorem (Completeness)

SOL with Henkin semantics is complete under LEM.

Theorem (Compactness)

SOL with Henkin semantics is compact under LEM.

Further Consequences of the Reduction

Theorem (Relative Löwenheim-Skolem)

If FOL has the Löwenheim-Skolem property, then so does SOL with Henkin semantics.

Conclusion

Mechanisation (Hyperlinked with PDF):

Conclusion

Mechanisation (Hyperlinked with PDF):

- Except for completeness, all results are fully constructive

Conclusion

Mechanisation (Hyperlinked with PDF):

- Except for completeness, all results are fully constructive
- Overall 8k new LOC and 1.5k reused

Conclusion

Mechanisation (Hyperlinked with PDF):

- Except for completeness, all results are fully constructive
- Overall 8k new LOC and 1.5k reused
- Undecidability results contributed to the Coq Library of Undecidability Proofs [Forster et al., 2020]

Conclusion

Mechanisation (Hyperlinked with PDF):

- Except for completeness, all results are fully constructive
- Overall 8k new LOC and 1.5k reused
- Undecidability results contributed to the Coq Library of Undecidability Proofs [Forster et al., 2020]

`https://www.ps.uni-saarland.de/extras/cpp22-sol/`

Conclusion

Mechanisation (Hyperlinked with PDF):

- Except for completeness, all results are fully constructive
- Overall 8k new LOC and 1.5k reused
- Undecidability results contributed to the Coq Library of Undecidability Proofs [Forster et al., 2020]

`https://www.ps.uni-saarland.de/extras/cpp22-sol/`

Future work:

- Löwenheim-Skolem theorem for FOL (work in progress)

Conclusion

Mechanisation (Hyperlinked with PDF):

- Except for completeness, all results are fully constructive
- Overall 8k new LOC and 1.5k reused
- Undecidability results contributed to the Coq Library of Undecidability Proofs [Forster et al., 2020]

```
https://www.ps.uni-saarland.de/extras/cpp22-sol/
```

Future work:

- Löwenheim-Skolem theorem for FOL (work in progress)
- Other second-order axiomatisations, e.g. ZF_2

Conclusion

Mechanisation (Hyperlinked with PDF):

- Except for completeness, all results are fully constructive
- Overall 8k new LOC and 1.5k reused
- Undecidability results contributed to the Coq Library of Undecidability Proofs [Forster et al., 2020]

`https://www.ps.uni-saarland.de/extras/cpp22-sol/`

Future work:

- Löwenheim-Skolem theorem for FOL (work in progress)
- Other second-order axiomatisations, e.g. ZF_2
- Internal Categoricity [Väänänen and Wang, 2012]

Conclusion

Mechanisation (Hyperlinked with PDF):

- Except for completeness, all results are fully constructive
- Overall 8k new LOC and 1.5k reused
- Undecidability results contributed to the Coq Library of Undecidability Proofs [Forster et al., 2020]

```
https://www.ps.uni-saarland.de/extras/cpp22-sol/
```

Future work:

- Löwenheim-Skolem theorem for FOL (work in progress)
- Other second-order axiomatisations, e.g. ZF_2
- Internal Categoricity [Väänänen and Wang, 2012]. Would require extensive tooling, maybe similar to the proof mode in [Hostert et al., 2021].

References i

Bauer, A. (2006).
First steps in synthetic computability theory.
Electronic Notes in Theoretical Computer Science, 155:5–31.
Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS XXI).

Davis, M. and Putnam, H. (1959).
A computational proof procedure; Axioms for number theory; Research on Hilbert's Tenth Problem.
Air Force Office of Scientific Research, Air Research and Development . . . .

Dedekind, R. (1888).
Was sind und was sollen die Zahlen?
Vieweg, Braunschweig.

Forster, Y., Kirst, D., and Smolka, G. (2019).
On synthetic undecidability in Coq, with an application to the Entscheidungsproblem.
In *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2019, page 38–51, New York, NY, USA. Association for Computing Machinery.

Forster, Y., Kirst, D., and Wehr, D. (2021).
Completeness theorems for first-order logic analysed in constructive type theory: Extended version.
Journal of Logic and Computation, 31(1):112–151.

Forster, Y., Larchey-Wendling, D., Dudenhefner, A., Heiter, E., Kirst, D., Kunze, F., Smolka, G., Spies, S., Wehr, D., and Wuttke, M. (2020).
A Coq library of undecidable problems.
In *CoqPL 2020 The Sixth International Workshop on Coq for Programming Languages*.

Harrison, J. (2006).
Towards self-verification of HOL Light.
In Furbach, U. and Shankar, N., editors, *Proceedings of the third International Joint Conference, IJCAR 2006*, volume 4130 of *Lecture Notes in Computer Science*, pages 177–191, Seattle, WA. Springer-Verlag.

Henkin, L. (1949).
The completeness of the first-order functional calculus.
The journal of symbolic logic, 14(3):159–166.

Hostert, J., Koch, M., and Kirst, D. (2021).
A toolbox for mechanised first-order logic.
The Coq Workshop.

Kirst, D. and Hermes, M. (2021).
Synthetic undecidability and incompleteness of first-order axiom systems in Coq.
In *ITP*.

Kleene, S. C. (1952).
Introduction to metamathematics.

Kumar, R., Arthan, R., Myreen, M. O., and Owens, S. (2016).
Self-formalisation of higher-order logic.
Journal of Automated Reasoning, 56(3):221–259.

Larchey-Wendling, D. and Forster, Y. (2019).
Hilbert's Tenth Problem in Coq.
In Geuvers, H., editor, *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*, volume 131 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:20, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Matijasevič, Y. V. (1971).
Diophantine representation of recursively enumerable predicates.
In *Studies in Logic and the Foundations of Mathematics*, volume 63, pages 171–177. Elsevier.

Nour, K. and Raffalli, C. (2003).
Simple proof of the completeness theorem for second-order classical and intuitionistic logic by reduction to first-order mono-sorted logic.
Theoretical computer science, 308(1-3):227–237.

Richman, F. (1983).
Church's thesis without tears.
The Journal of Symbolic Logic, 48(3):797–803.

Robinson, J. (1952).
Existential definability in arithmetic.
Transactions of the American Mathematical Society, 72(3):437–449.

Shapiro, S. (1991).
***Foundations without foundationalism: A case for second-order logic*, volume 17.**
Clarendon Press.

Tennant, N. (1990).
***Natural logic*.**
Edinburgh University Press.

Van Dalen, D. (1994).
***Logic and structure*, volume 3.**
Springer.

Väänänen, J. and Wang, T. (2012).
Internal categoricity in arithmetic and set theory.
Notre Dame Journal of Formal Logic, 56.

Failure of Strong Completeness

Theorem (Failure of Strong Completeness).

SOL is not strongly complete for full semantics and decidable theories.

Proof.

Let $\vdash$ be sound and strongly complete.

There is no model of $\mathcal{T}_{\neq}$. Thus

$$\mathcal{T}_{\neq} \vDash \dot{\bot} \xrightarrow{\text{Completeness}} \mathcal{T}_{\neq} \vdash \dot{\bot} \to \begin{array}{c} \Gamma \vdash \dot{\bot} \\ \text{for } \Gamma \subseteq_{\text{fin}} \mathcal{T}_{\neq} \end{array} \xrightarrow{\text{Soundness}} \Gamma \vDash \dot{\bot}$$

But $\Gamma \subseteq_{\text{fin}} \mathcal{T}_{\neq}$ has a model. $\square$

Synthetic Computability Theory [Forster et al., 2019]

Every function definable in constructive type theory is computable.

This allows a synthetic rendering of computability theory without relying on a concrete model of computation.

A problem $P : X \to \mathrm{Prop}$ ...

- is **decidable** if $\exists f : X \to \mathbb{B}. \forall x. P(X) \leftrightarrow f(x) = \mathrm{true}$.
- is **enumerable** if $\exists f : \mathbb{N} \to \mathcal{O}(X). \forall x. P(X) \leftrightarrow \exists n. f(n) = x$.
- **reduces to** $Q : Y \to \mathrm{Prop}$ if $\exists f : X \to Y. \forall x. P(x) \leftrightarrow Q(f(x))$.

Semantic Henkin Reduction

- Turn Henkin model $\mathcal{H}$ into first-order model $\mathcal{H}^\star$ with $D^\star := D \cup \mathbb{U}$ and $\mathsf{App}_n\,(x :: \vec{v}) := \mathsf{toPred}_n\, x\,(\mathsf{toIndi}\ \vec{v})$

$$\mathcal{H} \vDash_2 \varphi \;\leftrightarrow\; \mathcal{H}^\star \vDash_1 \varphi^\star$$

- Turn first-order model $\mathcal{M}$ into Henkin model $\mathcal{M}^\diamond$ with $D^\diamond := D$ and $\mathbb{U}$ induced by interpretation of App.

$$\mathcal{M} \vDash_1 \mathsf{Compr}^\star \;\rightarrow\; \mathcal{M}^\diamond \vDash_2 \varphi \;\leftrightarrow\; \mathcal{M} \vDash_1 \varphi^\star$$

Backwards Translation

Define a backwards translation $\_^{\diamond} : \mathsf{form}_1 \to \mathsf{form}_2$. For example

$$(\forall x.\, \mathsf{App}_0(x) \mathbin{\dot\wedge} \mathsf{App}_1(x, x))^{\diamond}$$

$$||$$

$$\forall x\, X^0\, X^1.\, X^0 \mathbin{\dot\wedge} X^1(x)$$

$$(\mathsf{App}_1(f(x), y))^{\diamond} = \dot\bot_1(y)$$

Special error symbols $\dot\bot_n$ if first argument is not a variable

Internal Categoricity [Väänänen and Wang, 2012]

Consider a theory $\mathcal{T}$ depending on a single predicate symbol $\mathcal{P}$

$$\mathrm{Categ}(\mathcal{T}) := \dot{\forall} D_1 D_2 P_1 P_2 .\, \mathcal{T}(P_1)^{D_1} \dot{\rightarrow} \mathcal{T}(P_2)^{D_2} \dot{\rightarrow} \dot{\exists} \cong .\, \mathrm{Iso}(\cong, D_1, D_2, P_1, P_2)$$

where $\mathcal{T}(P_1)^{D_1}$ replaces $\mathcal{P}$ with the variable P_1 and guards all quantifiers with the domain predicate D_1.

- $\mathcal{T}$ is categorical iff $\vDash \mathrm{Categ}(\mathcal{T})$
- Provable in many cases (despite incompleteness), e.g. $\vdash \mathrm{Categ}(\mathrm{PA}_2)$.
  - $\Rightarrow$ Categoricity can be written and proven at the object level, without depending on any external set theory (or type theory in our case)