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
Introduction

First-Order Logic
Quantification only over individuals

∀\varphi. \varphi(0) \rightarrow (\forall n. \varphi(n) \rightarrow \varphi(n+1)) \rightarrow \forall n. \varphi(n)

Second-Order Logic
Quantification over individuals & their properties

∀P. P(0) \rightarrow (\forall n. P(n) \rightarrow P(n+1)) \rightarrow \forall n. P(n)
First-Order Logic
Quantification only over individuals
\[ \varphi(0) \rightarrow (\forall n. \varphi(n) \rightarrow \varphi(n+1)) \rightarrow \forall n. \varphi(n) \]
for all formulas \( \varphi \)

Second-Order Logic
Quantification over individuals & their properties
\[ \forall P. P(0) \rightarrow (\forall n. P(n) \rightarrow P(n+1)) \rightarrow \forall n. P(n) \]
First-Order Logic
Quantification only over individuals

\[ \varphi(0) \rightarrow (\forall n. \varphi(n) \rightarrow \varphi(n+1)) \rightarrow \forall n. \varphi(n) \]
for all formulas \( \varphi \)

Second-Order Logic
Quantification over individuals & their properties

\[ \forall P. P(0) \rightarrow (\forall n. P(n) \rightarrow P(n+1)) \rightarrow \forall n. P(n) \]

Behaviour of SOL depends on interpretation of second-order quantifiers:
First-Order Logic
Quantification only over individuals

\[ \varphi(0) \rightarrow (\forall n. \varphi(n) \rightarrow \varphi(n + 1)) \rightarrow \forall n. \varphi(n) \]
for all formulas \( \varphi \)

Second-Order Logic
Quantification over individuals & their properties

\[ \forall P. P(0) \rightarrow (\forall n. P(n) \rightarrow P(n + 1)) \rightarrow \forall n. P(n) \]

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

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

First-Order Logic
Quantification only over individuals

$$\varphi(0) \rightarrow (\forall n. \varphi(n) \rightarrow \varphi(n + 1)) \rightarrow \forall n. \varphi(n)$$
for all formulas $\varphi$

Second-Order Logic
Quantification over individuals & their properties

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

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
# Introduction

## First-Order Logic
Quantification only over individuals

\[ \varphi(0) \rightarrow (\forall n. \varphi(n) \rightarrow \varphi(n + 1)) \rightarrow \forall n. \varphi(n) \]

for all formulas \( \varphi \)

## Second-Order Logic
Quantification over individuals & their properties

\[ \forall P. P(0) \rightarrow (\forall n. P(n) \rightarrow P(n + 1)) \rightarrow \forall n. P(n) \]

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
Introduction

**First-Order Logic**
Quantification only over individuals

\[ \varphi(0) \rightarrow (\forall n. \varphi(n) \rightarrow \varphi(n+1)) \rightarrow \forall n. \varphi(n) \]
for all formulas \( \varphi \)

**Second-Order Logic**
Quantification over individuals & their properties

\[ \forall P. P(0) \rightarrow (\forall n. P(n) \rightarrow P(n+1)) \rightarrow \forall n. P(n) \]

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

\[ \varphi(0) \rightarrow (\forall n. \varphi(n) \rightarrow \varphi(n+1)) \rightarrow \forall n. \varphi(n) \]

for all formulas \( \varphi \)

**Second-Order Logic**
Quantification over individuals & their properties

\[ \forall P. P(0) \rightarrow (\forall n. P(n) \rightarrow P(n+1)) \rightarrow \forall n. P(n) \]

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.
Given a signature \( \Sigma = (\Sigma_F, \Sigma_P) \), we inductively define

\[ t ::= x_i | \vec{t}(F: \Sigma_F)(i:N) \]

\[ \phi, \psi ::= \bot | P \vec{t} | p_n i \vec{t} | \phi \Box \psi | \phi \nabla \psi | \phi \nabla n_2 \phi (P: \Sigma_P)(i, n:N) \]

Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021]) ⇒ 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.
Given a signature $\Sigma = (\Sigma_F, \Sigma_P)$, we inductively define

$$t ::= x_i \mid F \overrightarrow{t} \quad (F : \Sigma_F) \ (i : \mathbb{N})$$

$$\varphi, \psi ::= \bot \mid P \overrightarrow{t} \mid p^i_n \overrightarrow{t} \mid \varphi \downarrow \psi \mid \nabla \varphi \mid \nabla^n \varphi \quad (P : \Sigma_P) \ (i, n : \mathbb{N})$$
Mechanisation

Given a signature $\Sigma = (\Sigma_F, \Sigma_P)$, we inductively define:

$$t ::= x_i \mid F \overrightarrow{t} \quad (F : \Sigma_F) \ (i : N)$$

$$\varphi, \psi ::= \bot \mid P \overrightarrow{t} \mid p^n_i \overrightarrow{t} \mid \varphi \Box \psi \mid \nabla \varphi \mid \nabla^n \varphi \quad (P : \Sigma_P) \ (i, n : N)$$

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

Given a signature $\Sigma = (\Sigma_F, \Sigma_P)$, we inductively define

$t ::= x_i \mid \mathcal{F} \vec{t}$ \hspace{1cm} \((\mathcal{F} : \Sigma_F) \ (i : \mathbb{N})\)

$\varphi, \psi ::= \bot \mid \mathcal{P} \vec{t} \mid p^n_i \vec{t} \mid \varphi \Box \psi \mid \vec{\nabla} \varphi \mid \vec{\nabla}^n_2 \varphi$ \hspace{1cm} \((\mathcal{P} : \Sigma_P) \ (i, n : \mathbb{N})\)

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  - $\Rightarrow$ De Bruijn binders
Given a signature $\Sigma = (\Sigma_F, \Sigma_P)$, we inductively define

$$t ::= x_i \mid F \vec{t} \quad (F : \Sigma_F) \quad (i : \mathbb{N})$$

$$\varphi, \psi ::= \perp \mid \mathcal{P} \vec{t} \mid p_i^n \vec{t} \mid \varphi \square \psi \mid \nabla \varphi \mid \nabla_2^n \varphi \quad (P : \Sigma_P) \quad (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  - $\Rightarrow$ De Bruijn binders, non-primitive equality
Given a signature $\Sigma = (\Sigma_F, \Sigma_P)$, we inductively define

$$t ::= x_i \mid F \overrightarrow{t} \quad (F : \Sigma_F) \ (i : \mathbb{N})$$

$$\varphi, \psi ::= \bot \mid P \overrightarrow{t} \mid P^n \overrightarrow{t} \mid \varphi \boxdot \psi \mid \hat{\nabla} \varphi \mid \hat{\nabla}^n \varphi \quad (P : \Sigma_P) \ (i, n : \mathbb{N})$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  - ⇒ De Bruijn binders, non-primitive equality, type class for signatures
Given a signature $\Sigma = (\Sigma_F, \Sigma_P)$, we inductively define

\[
\begin{align*}
t & ::= x_i \mid \vec{F} \vec{t} \quad & (\vec{F} : \Sigma_F) \quad (i : \mathbb{N}) \\
\varphi, \psi & ::= \bot \mid \vec{P} \vec{t} \mid p_i^n \vec{t} \mid \varphi \bigcirc \psi \mid \vec{\nabla} \varphi \mid \vec{\nabla}_2^n \varphi \quad & (\vec{P} : \Sigma_P) \quad (i, n : \mathbb{N})
\end{align*}
\]

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  - 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 $\Sigma = (\Sigma_F, \Sigma_P)$, we inductively define

$$t ::= x_i \mid \mathcal{F} \vec{t} \quad (\mathcal{F} : \Sigma_F) \ (i : \mathbb{N})$$
$$\varphi, \psi ::= \bot \mid \mathcal{P} \vec{t} \mid p_i^n \vec{t} \mid \varphi \Box \psi \mid \nabla \varphi \mid \nabla^2 \varphi \quad (\mathcal{P} : \Sigma_P) \ (i, n : \mathbb{N})$$

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

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

$$
\phi, \psi ::= \bot \mid \mathcal{P} \overrightarrow{t} \mid p_i^n \overrightarrow{t} \mid \varphi \Box \psi \mid \dot{\nabla} \varphi \mid \dot{\nabla}^n_2 \varphi \quad (\mathcal{P} : \Sigma_\mathcal{P}) \ (i, n : \mathbb{N})
$$

- Follow previous FOL mechanisations (e.g. [Kirst and Hermes, 2021])
  - 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
Given a signature $\Sigma = (\Sigma_F, \Sigma_P)$, we inductively define
\[
t ::= x_i \quad | \quad F \overrightarrow{t} \quad (F : \Sigma_F) \quad (i : \mathbb{N})
\]
\[
\varphi, \psi ::= \bot \quad | \quad P \overrightarrow{t} \quad | \quad p_i^n t \quad | \quad \varphi \Box \psi \quad | \quad \nabla \varphi \quad | \quad \nabla_2^n \varphi \quad (P : \Sigma_P) \quad (i, n : \mathbb{N})
\]

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

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

Interpretation ($\models$) in $M$ maps connectives $\Box$ and quantifiers $\forall$ to their counterparts in $\text{Prop}$. 
Definition (Full Semantics)

- A model $\mathcal{M}$ consists of a domain $D$ and interpretations $\mathcal{F}^\mathcal{M} : D^{|\mathcal{F}|} \rightarrow D$ and $\mathcal{P}^\mathcal{M} : D^{|\mathcal{P}|} \rightarrow \text{Prop.}$
- Interpretation ($\models$) in $\mathcal{M}$ maps connectives $\Box$ and quantifiers $\nabla$ to their counterparts in Prop.
- SOL quantifiers $\nabla_2^n$ range over the full relation space $D^n \rightarrow \text{Prop.}$
Second-Order Peano Arithmetic

Zero Addition: $\forall x. O + x \equiv x$
Addition Recursion: $\forall xy. (Sx) + y \equiv S(x + y)$
Disjointness: $\forall x. O \equiv Sx \rightarrow \bot$
Equality Reflexivity: $\forall x. x \equiv x$

**Induction:** $\forall P. P(O) \rightarrow (\forall x. P(x) \rightarrow P(Sx)) \rightarrow \forall x. P(x)$

Zero Multiplication: $\forall x. O \cdot x \equiv O$
Multiplication Recursion: $\forall xy. (Sx) \cdot y \equiv y + x \cdot y$
Successor Injectivity: $\forall xy. Sx \equiv Sy \rightarrow x \equiv y$
Equality Symmetry: $\forall xy. x \equiv y \rightarrow y \equiv x$
Second-Order Peano Arithmetic

Zero Addition: $\forall x. O + x \equiv x$
Addition Recursion: $\forall xy. (Sx) + y \equiv S(x + y)$
Disjointness: $\forall x. O \equiv Sx \Rightarrow \bot$
Equality Reflexivity: $\forall x. x \equiv x$

Induction: $\forall P. P(O) \Rightarrow (\forall x. P(x) \Rightarrow P(Sx)) \Rightarrow \forall x. P(x)$

Zero Multiplication: $\forall x. O \cdot x \equiv O$
Multiplication Recursion: $\forall xy. (Sx) \cdot y \equiv y + x \cdot y$
Successor Injectivity: $\forall xy. Sx \equiv Sy \Rightarrow x \equiv y$
Equality Symmetry: $\forall xy. x \equiv y \Rightarrow y \equiv x$

**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: \( \forall x. O + x \equiv x \)

Addition Recursion: \( \forall x y. (Sx) + y \equiv S(x + y) \)

Disjointness: \( \forall x. O \equiv Sx \rightarrow \bot \)

Equality Reflexivity: \( \forall x. x \equiv x \)

Zero Multiplication: \( \forall x. O \cdot x \equiv O \)

Multiplication Recursion: \( \forall x y. (Sx) \cdot y \equiv y + x \cdot y \)

Successor Injectivity: \( \forall x y. Sx \equiv Sy \rightarrow x \equiv y \)

Equality Symmetry: \( \forall x y. x \equiv y \rightarrow y \equiv x \)

Induction: \( \forall P. P(O) \rightarrow (\forall x. P(x) \rightarrow P(Sx)) \rightarrow \forall x. P(x) \)

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

\( \text{PA}_2 \) is categorical for full semantics, i.e. all models of \( \text{PA}_2 \) are isomorphic.

Proof.

Given models \( M_1, M_2 \models \text{PA}_2 \), inductively define \( \cong : D_1 \rightarrow D_2 \rightarrow \text{Prop} \)

\[ O^{M_1} \cong O^{M_2} \quad S^{M_1} x \cong S^{M_2} y \quad \text{if } x \cong y. \]
Second-Order Peano Arithmetic

Zero Addition: $\forall x. O + x \equiv x$
Addition Recursion: $\forall xy. (Sx) + y \equiv S(x + y)$
Disjointness: $\forall x. O \equiv Sx \rightarrow \bot$
Equality Reflexivity: $\forall x. x \equiv x$

Induction: $\forall P. P(O) \rightarrow (\forall x. P(x) \rightarrow P(Sx)) \rightarrow \forall x. P(x)$

Zero Multiplication: $\forall x. O \cdot x \equiv O$
Multiplication Recursion: $\forall xy. (Sx) \cdot y \equiv y + x \cdot y$
Successor Injectivity: $\forall xy. Sx \equiv Sy \rightarrow x \equiv y$
Equality Symmetry: $\forall xy. x \equiv y \rightarrow y \equiv x$

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

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

Verify that $\cong$ is an isomorphism using the induction axiom. $\square$
Corollary (Failure of Löwenheim-Skolem)

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

Proof. Consider the theory $T \neq \mathbb{PA}$, $x \neq 0$, $x \neq S0$, $x \neq S(S0)$, ...

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

But $\mathbb{N}$ is not a model of the whole theory $T \neq$.

Since $\mathbb{N}$ is the only model of $\mathbb{PA}$, we can conclude that $T \neq$ does not have a model. □
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 $T \neq \mathbb{PA}$, $x \neq 0$, $x \neq SO$, $x \neq S(SO)$, ...

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

But $\mathbb{N}$ is not model of the whole theory $T \neq \mathbb{PA}$.

Since $\mathbb{N}$ is the only model of $\mathbb{PA}$, we can conclude that $T \neq \mathbb{PA}$ does not have a model. □
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 $\mathcal{T} := PA_2, x \neq O, x \neq SO, x \neq S(SO), ...$
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 \( T \neq := \text{PA}_2, x \neq O, x \neq S O, x \neq S (S O), ... \)

- Every finite subset of \( T \neq \) 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 $\mathcal{T}_\neq := \text{PA}_2, x \neq O, x \neq S O, x \neq S (S O), ...$

- Every finite subset of $\mathcal{T}_\neq$ has a model, for example $\mathbb{N}$.
- But $\mathbb{N}$ is not model of the whole theory $\mathcal{T}_\neq$. 
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 $\mathcal{T} := \text{PA}_2, x \neq O, x \neq S O, x \neq S (S O), ...$

- Every finite subset of $\mathcal{T}$ has a model, for example $\mathbb{N}$.
- But $\mathbb{N}$ is not a model of the whole theory $\mathcal{T}$. Since $\mathbb{N}$ is the only model of $\text{PA}_2$, we can conclude that $\mathcal{T}$ does not have a model. $\square$
Consequences of Categoricity

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

SOL is not strongly complete for full semantics.

Deduction system $\vdash: \mathcal{L}(\text{form}) \rightarrow \text{form} \rightarrow \text{Prop}$
Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

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

- Completeness: \( \Gamma \models \varphi \rightarrow \Gamma \vdash \varphi \) for all lists \( \Gamma \)
Consequences of Categoricity

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

SOL is not strongly complete for full semantics.

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

- Completeness: $\Gamma \models \varphi \rightarrow \Gamma \vdash \varphi$ for all lists $\Gamma$
- Lift $\vdash$ to theories: $\mathcal{T} \vdash \varphi := \exists \Gamma \subseteq \text{fin} \mathcal{T}. \Gamma \vdash \varphi$
Consequences of Categoricity

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

SOL is not strongly complete for full semantics.

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

- Completeness: $\Gamma \models \varphi \rightarrow \Gamma \vdash \varphi$ for all lists $\Gamma$
- Lift $\vdash$ to theories: $\mathcal{T} \vdash \varphi := \exists \Gamma \subseteq_{\text{fin}} \mathcal{T}. \Gamma \vdash \varphi$
- Strong completeness: $\mathcal{T} \models \varphi \rightarrow \mathcal{T} \vdash \varphi$
Consequences of Categoricity

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

SOL is not strongly complete for full semantics.

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

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

No computability assumptions on $\vdash$
Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.
Theorem (Failure of Strong Completeness [Tennant, 1990])

SOL is not strongly complete for full semantics.

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

- 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: $\Gamma \models \varphi \rightarrow \Gamma \vdash \varphi$

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

SOL is not strongly complete for full semantics.

Does not rule out the weaker notion of completeness: \( \Gamma \models \varphi \rightarrow \Gamma \vdash \varphi \)

- 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: $\Gamma \models \varphi \rightarrow \Gamma \vdash \varphi$

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


- $P$ 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: $\Gamma \models \varphi \rightarrow \Gamma \vdash \varphi$

- 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. Halt).
Incompleteness

**Lemma**

Validity in PA₂ is not enumerable.
Incompleteness

**Lemma**

Validity in PA$_2$ is not enumerable.

**Proof Sketch.**

Via reduction from the complement of Hilbert’s tenth problem $\overline{H}_{10}$:\textsuperscript{1}

\textsuperscript{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 $\text{PA}_2$ is not enumerable.

**Proof Sketch.**

Via reduction from the complement of Hilbert’s tenth problem $\overline{H_{10}}$:\(^1\)

\[ x + 2 = y^2 + z \]

\[ s = \underbrace{y^2 + z}_t \]

\(^1\)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 PA₂ is not enumerable.

Proof Sketch.

Via reduction from the complement of Hilbert’s tenth problem \( \overline{H_{10}} \):¹

\[
\begin{align*}
    x + 2 &= y^2 + z \\
    \forall s, t :&= \exists xyz \cdot x + S(S O) \equiv y \cdot y + z
\end{align*}
\]

**Lemma**

Validity in \( \text{PA}_2 \) is not enumerable.

**Proof Sketch.**

Via reduction from the complement of Hilbert’s tenth problem \( \overline{H}_{10} \):\(^1\)

\[
x + 2 = y^2 + z \quad \sim \quad \varphi_{s,t} := \exists x y z . x + S(SO) \equiv y \cdot y + z
\]

\( s = t \) is unsolvable iff \( \mathbb{N} \models \neg \varphi_{s,t} \)

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

**Lemma**

Validity in $\text{PA}_2$ is not enumerable.

**Proof Sketch.**

Via reduction from the complement of Hilbert’s tenth problem $\overline{H_{10}}$:

\[
\begin{align*}
\underbrace{x + 2 = y^2 + z}_{s} \quad &\quad \text{and} \quad \underbrace{\varphi_{s,t} := \exists \, xyz. \, x + S (S \circ)}_{t} \equiv y \cdot y + z \\
\end{align*}
\]

$s = t$ is unsolvable iff $\mathbb{N} \nvdash \neg \varphi_{s,t}$ and thus iff $\text{PA}_2 \models \neg \varphi_{s,t}$ by categoricity.

\(^1\)Whose undecidability [Davis and Putnam, 1959, Robinson, 1952, Matijasevič, 1971] has already been mechanised in Coq [Larchey-Wendling and Forster, 2019].
Theorem (Incompleteness)
SOL is not complete for full semantics
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 $H_{10}$.
**Theorem (Incompleteness)**

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

**Theorem (Undecidability)**

Second-order validity and satisfiability in the empty signature are undecidable.
**Theorem (Incompleteness)**

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

**Theorem (Undecidability)**

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

**Proof Sketch.**

- $s = t$ has a solution iff $\forall O S + \times \equiv. \ PA_2 \rightarrow \varphi_{s,t}$ is valid.
Theorem (Incompleteness)

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

Theorem (Undecidability)

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

Proof Sketch.

- $s = t$ has a solution iff $\forall O S + \times \equiv. PA_2 \rightarrow \varphi_{s,t}$ is valid.
- $s = t$ has a solution iff $\exists O S + \times \equiv. PA_2 \land \varphi_{s,t}$ is satisfiable.
Henkin Semantics: Completeness
Definition (Henkin Semantics).

- Second-order quantifiers $\nabla_2^n$ only range over the relations contained in a universe $\mathbb{U}_n : (D^n \rightarrow \text{Prop}) \rightarrow \text{Prop}$. 
Definition (Henkin Semantics).

- Second-order quantifiers $\nabla^n_2$ 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 $\mathcal{H}$.
Definition (Henkin Semantics).

- Second-order quantifiers $\nabla^n_2$ 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.
Definition (Henkin Semantics).

- Second-order quantifiers $\nabla^n_2$ only range over the relations contained in a universe $U_n : (D^n \rightarrow \text{Prop}) \rightarrow \text{Prop}$.
- $U_n$ is specified by a Henkin model $\mathcal{H}$.
- $U_n$ 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:

$$A \vdash_2 \exists P. \forall x_1 \ldots x_n. P(x_1, \ldots, x_2) \leftrightarrow \varphi$$

$\text{Compr}_\varphi$
Definition (Henkin Semantics).

- Second-order quantifiers $\nabla^n_2$ 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:

$$A \vdash_2 \exists P. \forall x_1 \ldots x_n. P(x_1, \ldots, x_2) \leftrightarrow \varphi$$

$\vdash_2$ is complete for Henkin semantics [Henkin, 1949].
SOL with Henkin semantics is essentially just many-sorted FOL:
Connection to FOL

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

\[ \varphi := \forall x. \exists P. P(x, x) \quad \leadsto \quad \varphi^* := \forall x^I. \exists p^{P_2}. \text{App}_2(p, x, x) \]

Guard the quantifiers with predicates to distinguish the sorts \[ \text{[Van Dalen, 1994]} \].

However, difficult to prove \( \vdash_1 \varphi \rightarrow \vdash_2 \varphi \).

\[ \text{[Nour and Raffalli, 2003]} \] propose:

\[ \varphi^* := \forall x. \exists p. \text{App}_2(p, x, x) \]

\( x \) and \( p \) represent individuals and predicates at the same time.

\( \Rightarrow \) We can transport the completeness theorem from FOL to SOL.
Connection to FOL

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

\[ \varphi := \forall x. \exists P. P(x, x) \leadsto \varphi^* := \forall x^I. \exists p^P_2. \text{App}_2(p, x, x) \]

⇒ We can transport the completeness theorem from FOL to SOL
Connection to FOL

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

\[ \varphi := \forall x. \exists P. P(x, x) \quad \leadsto \quad \varphi^* := \forall x^I. \exists p^P_2. \text{App}_2(p, x, x) \]

\[ \varphi^* := \forall x. \text{isIndi}(x) \quad \Rightarrow \quad \exists p. \text{isPred}_2(p) \land \text{App}_2(p, x, x) \]

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

⇒ We can transport the completeness theorem from FOL to SOL
SOL with Henkin semantics is essentially just many-sorted FOL:

\[ \varphi := \forall x. \exists P. P(x, x) \quad \leadsto \quad \varphi^* := \forall x^I. \exists p^P. \text{App}_2(p, x, x) \]

\[ \varphi^* := \forall x. \text{isIndi}(x) \rightarrow \exists p. \text{isPred}_2(p) \land \text{App}_2(p, x, x) \]

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

However, difficult to prove \( \vdash_1 \varphi^* \rightarrow \vdash_2 \varphi \).

\[ \Rightarrow \text{We can transport the completeness theorem from FOL to SOL} \]
SOL with Henkin semantics is essentially just many-sorted FOL:

\[ \varphi := \forall x. \exists P. P(x, x) \quad \sim \quad \varphi^* := \forall x^I. \exists p^{P_2}. \text{App}_2(p, x, x) \]

\[ \varphi^* := \forall x. \text{isIndi}(x) \Rightarrow \exists p. \text{isPred}_2(p) \land \text{App}_2(p, x, x) \]

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

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

\[ \Rightarrow \text{We can transport the completeness theorem from FOL to SOL} \]
Connection to FOL

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

\[ \varphi := \forall x \exists P \cdot P(x, x) \iff \varphi^* := \forall x^I \exists p^{P_2} \cdot \text{App}_2(p, x, x) \]

\[ \varphi^* := \forall x \; \text{isIndi}(x) \to \exists p \; \text{isPred}_2(p) \land \text{App}_2(p, x, x) \]

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

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

\[ \varphi^* := \forall x \exists p \cdot \text{App}_2(p, x, x) \]

$\Rightarrow$ We can transport the completeness theorem from FOL to SOL
SOL with Henkin semantics is essentially just many-sorted FOL:

\[ \varphi := \forall x. \exists P. P(x, x) \quad \iff \quad \varphi^* := \forall x^I. \exists p^{P_2}. \text{App}_2(p, x, x) \]

\[ \varphi^* := \forall x. \text{isIndi}(x) \Rightarrow \exists p. \text{isPred}_2(p) \land \text{App}_2(p, x, x) \]

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

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

\[ \varphi^* := \forall x. \exists p. \text{App}_2(p, x, x) \]

\( x \) and \( p \) represent individuals and predicates at the same time.

\( \Rightarrow \) We can transport the completeness theorem from FOL to SOL.
\[ \mathcal{T} \models_{2} \varphi \]

\[ (\mathcal{T} \cup \text{Compr})^* \models_{1} \varphi^* \]
Reduction to FOL [Nour and Raffalli, 2003]

\[
\mathcal{T} \models_2 \varphi
\]

\[
\Downarrow
\]

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

\[
(\mathcal{T} \cup \text{Compr})^* \models_1 \varphi^* \rightarrow \text{---------------------------} \rightarrow \mathcal{T} \cup \text{Compr})^* \models_1 \varphi^*
\]
Reduction to FOL [Nour and Raffalli, 2003]

\[ T \vDash_2 \varphi \]

\[ (T \cup \text{Compr})^* \vDash_1 \varphi^* \rightarrow (T \cup \text{Compr})^* \vDash_1 \varphi^* \]

FOL completeness (MP / LEM) [Forster et al., 2021]
Reduction to FOL [Nour and Raffalli, 2003]

\[ T \vdash_2 \varphi \]

FOL completeness (MP / LEM)

\[ (T \cup \text{Compr})^* \vdash_1 \varphi^* \rightarrow (T \cup \text{Compr})^* \vdash_1 \varphi^* \]

[Forster et al., 2021]
Reduction to FOL [Nour and Raffalli, 2003]

\[ \mathcal{T} \vdash_2 \varphi \]

\[ (\mathcal{T} \cup \text{Compr})^* \models_1 \varphi^* \quad \xrightarrow{\text{FOL completeness (MP / LEM)}} \quad (\mathcal{T} \cup \text{Compr})^* \vdash_1 \varphi^* \]

Define FOL to SOL translation \( \circ \) that satisfies
Reduction to FOL [Nour and Raffalli, 2003]

\[ \mathcal{T} \models_2 \varphi \]

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

\[(\mathcal{T} \cup \text{Compr})^* \models_1 \varphi^* \rightarrow \mathcal{T} \cup \text{Compr}^* \models_1 \varphi^*\]

Define FOL to SOL translation \(\Diamond\) that satisfies

\[ \models_2 \varphi^* \Diamond \iff \varphi \]
Reduction to FOL [Nour and Raffalli, 2003]

\[ \mathcal{T} \vdash_{2} \varphi \]

\[ \vdash_{1} \varphi^{\ast} \leftrightarrow \varphi \]

\[ \vdash_{1} A \rightarrow A^\diamond \vdash_{2} \varphi^\diamond \]

FOL completeness (MP / LEM)

\[ (\mathcal{T} \cup \text{Compr})^* \vdash_{1} \varphi^* \rightarrow \vdash_{2} \varphi \]

[Forster et al., 2021]
Reduction to FOL [Nour and Raffalli, 2003]

\[ T \vdash_2 \varphi \]

\[ \vdash_2 \varphi \]

\[ (T \cup \text{Compr})^* \vdash_1 \varphi^* \]

\[ (T \cup \text{Compr})^* \vdash_1 \varphi^* \]

Define FOL to SOL translation \( _\diamond \) that satisfies

\[ \vdash_2 \varphi^* \leftrightarrow \varphi \]

\[ A \vdash_1 \varphi \rightarrow A^\diamond \vdash_2 \varphi^\diamond \]

**Theorem (Relative Completeness)**

If FOL is complete, then so is SOL with Henkin semantics.
Reduction to FOL [Nour and Raffalli, 2003]

\[ \mathcal{T} \vdash_2 \varphi \]

FOL completeness (MP / LEM) \[ \text{[Forster et al., 2021]} \]

\[ (\mathcal{T} \cup \text{Compr})^* \vdash_1 \varphi^* \rightarrow (\mathcal{T} \cup \text{Compr})^* \vdash_1 \varphi^* \]

Define FOL to SOL translation \( \circ \) that satisfies

\[ \vdash_2 \varphi^* \circ \leftrightarrow \varphi \]

\[ A \vdash_1 \varphi \rightarrow A^\circ \vdash_2 \varphi^\circ \]

**Theorem (Completeness)**

SOL with Henkin semantics is complete under LEM.
Reduction to FOL [Nour and Raffalli, 2003]

\[ \mathcal{T} \vdash_2 \varphi \]

FOL completeness (MP / LEM)

\[ (\mathcal{T} \cup \text{Compr})^* \vdash_1 \varphi^* \Rightarrow \mathcal{T} \vdash_2 \varphi \]

Define FOL to SOL translation \( \_ \circ \) that satisfies

\[ \vdash_2 \varphi^* \circ \leftrightarrow \varphi \]

Theorem (Completeness)

SOL with Henkin semantics is complete under LEM.

Theorem (Compactness)

SOL with Henkin semantics is compact under LEM.
Theorem (Relative Löwenheim-Skolem)

If FOL has the Löwenheim-Skolem property, then so does SOL with Henkin semantics.
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
- Internal Categoricity [Väänänen and Wang, 2012]. Would require extensive tooling, maybe similar to the proof mode in [Hostert et al., 2021].
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

[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

Internal Categoricity [Väänänen and Wang, 2012]. Would require extensive tooling, maybe similar to the proof mode in [Hostert et al., 2021].
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]

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

Internal Categoricity [Väänänen and Wang, 2012]. Would require extensive tooling, maybe similar to the proof mode in [Hostert et al., 2021].
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. ZF₂
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/](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. ZF₂
- Internal Categoricity [Väänänen and Wang, 2012]. Would require extensive tooling, maybe similar to the proof mode in [Hostert et al., 2021].


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


A toolbox for mechanised first-order logic.
The Coq Workshop.

Synthetic undecidability and incompleteness of first-order axiom systems in Coq.
In ITP.

Introduction to metamathematics.

Kumar, R., Arthan, R., Myreen, M. O., and Owens, S. (2016).
Self-formalisation of higher-order logic.

Hilbert’s Tenth Problem in Coq.


Edinburgh University Press.

Springer.

*Notre Dame Journal of Formal Logic,* 56.
Theorem (Failure of Strong Completeness).

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

Proof.
Let ⊢ be sound and strongly complete.

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

\[
\mathcal{T}_\neq \models \bot \quad \text{Completeness} \quad \rightarrow \quad \mathcal{T}_\neq \not\models \bot \\
\text{for } \Gamma \subseteq_{\text{fin}} \mathcal{T}_\neq \quad \text{Soundness} \quad \rightarrow \quad \Gamma \models \bot
\]

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

\(\square\)
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 $P : X \rightarrow \text{Prop}$ ...

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

- Turn Henkin model $\mathcal{H}$ into first-order model $\mathcal{H}^*$ with $D^* := D \cup U$ and $\text{App}_n(x :: \vec{v}) := \text{toPred}_n x (\text{tolInd} \vec{v})$

  $$\mathcal{H} \models_2 \varphi \iff \mathcal{H}^* \models_1 \varphi^*$$

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

  $$\mathcal{M} \models_1 \text{Compr}^* \rightarrow \mathcal{M}^\diamond \models_2 \varphi \iff \mathcal{M} \models_1 \varphi^*$$
Define a backwards translation $\_^{\diamond} : \text{form}_1 \rightarrow \text{form}_2$. For example

$$(\forall x. \text{App}_0(x) \land \text{App}_1(x, x))^{\diamond}$$

$$\parallel$$

$$\forall x \ X^0 X^1. X^0 \land X^1(x)$$

$$(\text{App}_1(f(x), y))^{\diamond} = \downarrow_1(y)$$

Special error symbols $\downarrow_n$ if first argument is not a variable
Consider a theory \( \mathcal{T} \) depending on a single predicate symbol \( \mathcal{P} \)

\[
\text{Categ}(\mathcal{T}) := \forall D_1 D_2 P_1 P_2. \mathcal{T}(P_1)^{D_1} \Rightarrow \mathcal{T}(P_2)^{D_2} \Rightarrow \exists \approx . \text{Iso}(\approx, D_1, D_2, P_1, P_2)
\]

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