The Undecidability of First-Order Logic over Small Signatures

First Bachelor Seminar Talk

Advisors: Andrej Dudenhefner, Dominik Kirst
Supervisor: Prof. Gert Smolka

Johannes Hostert

May 21, 2021, Saarland University

FOL

Statements like

- $\forall a^{\mathbb{N}}\, b^{\mathbb{N}}, a + b = b + a$
- $\forall a^{\mathbb{N}}, a \neq 0 \to \exists b^{\mathbb{N}}, a = S\, b$

Properties:

FOL

Statements like

- $\forall a^{\mathbb{N}}\, b^{\mathbb{N}}, a + b = b + a$
- $\forall a^{\mathbb{N}}, a \neq 0 \to \exists b^{\mathbb{N}}, a = S\, b$

Properties:

- Quantifiers range over $\mathbb{N}$

FOL

Statements like

- $\forall ab, a + b = b + a$
- $\forall a, a \neq 0 \rightarrow \exists b, a = S\, b$

Properties:

- Quantifiers range over individuals, not predicates

FOL

Statements like

- $\forall ab, a + b = b + a$
- $\forall a, a \neq 0 \to \exists b, a = S\, b$

Properties:

- Quantifiers range over individuals, not predicates
- Function symbols: $+, \cdot, 0, S$

FOL

Statements like

- $\forall ab, a + b = b + a$
- $\forall a, a \neq 0 \to \exists b, a = S\, b$

Properties:

- Quantifiers range over individuals, not predicates
- Function symbols: $+, \cdot, 0, S$
- Relation symbols: $=, <$

FOL

Statements like

- $\forall ab, a + b = b + a$
- $\forall a, a \neq 0 \to \exists b, a = S\, b$

Properties:

- Quantifiers range over individuals, not predicates
- Function symbols: $+, \cdot, 0, S$
- Relation symbols: $=, <$

FOL over $\{0, S, +, \cdot\}; \{=; <\}$
$\mathbb{N}$ is a (Tarski) model with usual interpretation for $0, S, +, \cdot, =, <$

Problems in FOL

φ a formula of FOL, is φ

- valid in all models?
- satisfied by a model?
- (intutitonistically) provable in the abstract deduction system?

Problems in FOL

φ a formula of FOL, is φ

- valid in all models?
- satisfied by a model?
- (intutitonistically) provable in the abstract deduction system?

All problems are undecidable [Church, 1936] [Turing, 1936]

Problems in FOL

φ a formula of FOL, is φ

- valid in all models?
- satisfied by a model?
- (intutitonistically) provable in the abstract deduction system?

All problems are undecidable [Church, 1936] [Turing, 1936]

In classical logic:

- All three problems coincide

Problems in FOL

φ a formula of FOL, is φ

- valid in all models?
- satisfied by a model?
- (intutitonistically) provable in the abstract deduction system?

All problems are undecidable [Church, 1936] [Turing, 1936]

In classical logic:

- All three problems coincide

In our inituitionistic formalization [Forster et al., 2019]:

Problems in FOL

φ a formula of FOL, is φ

- valid in all models?
- satisfied by a model?
- (intutitonistically) provable in the abstract deduction system?

All problems are undecidable [Church, 1936] [Turing, 1936]

In classical logic:

- All three problems coincide

In our inituitionistic formalization [Forster et al., 2019]:

- Mechanization in Coq Library of Undecidability Proofs [Forster et al., 2020]

Problems in FOL

φ a formula of FOL, is φ

- valid in all models?
- satisfied by a model?
- (intutitonistically) provable in the abstract deduction system?

All problems are undecidable [Church, 1936] [Turing, 1936]

In classical logic:

- All three problems coincide

In our inituitionistic formalization [Forster et al., 2019]:

- Mechanization in Coq Library of Undecidability Proofs [Forster et al., 2020]
- φ int. provable $\rightarrow \varphi$ valid

Problems in FOL

φ a formula of FOL, is φ

- valid in all models?
- satisfied by a model?
- (intutitonistically) provable in the abstract deduction system?

All problems are undecidable [Church, 1936] [Turing, 1936]

In classical logic:

- All three problems coincide

In our inituitionistic formalization [Forster et al., 2019]:

- Mechanization in Coq Library of Undecidability Proofs [Forster et al., 2020]
- φ int. provable $\rightarrow$ φ valid
- φ valid $\rightarrow$ $(\neg\neg\varphi)$ int. provable.

Problems in FOL

φ a formula of FOL, is φ

- valid in all models?
- satisfied by a model?
- (intutitonistically) provable in the abstract deduction system?

All problems are undecidable [Church, 1936] [Turing, 1936]

In classical logic:

- All three problems coincide

In our inituitionistic formalization [Forster et al., 2019]:

- Mechanization in Coq Library of Undecidability Proofs [Forster et al., 2020]
- φ int. provable $\to$ φ valid
- φ valid $\to$ $(\neg\neg\varphi)$ int. provable.
- Only shows $\overline{Halts_{TM}} \preceq Satisfiability$

Special cases

Are there signatures where these problems are decidable?

Special cases

Are there signatures where these problems are decidable?

- Only unary functions/relations: ✓ [Löwenheim, 1915]

Special cases

Are there signatures where these problems are decidable?

- Only unary functions/relations: ✓ [Löwenheim, 1915]
- Binary relation: $\times$
- Binary function, unary relation: $\times$

Special cases

Are there signatures where these problems are decidable?

- Only unary functions/relations: ✓ [Löwenheim, 1915]
- Binary relation: $\times$
- Binary function, unary relation: $\times$

Proof:

- Textbook: Signature compression reduction chain [Kalmár, 1939]
  - Very hard to mechanize in Coq [Kirst and Larchey-Wendling, 2020]

Special cases

Are there signatures where these problems are decidable?

- Only unary functions/relations: ✓ [Löwenheim, 1915]
- Binary relation: $\times$
- Binary function, unary relation: $\times$

Proof:

- Textbook: Signature compression reduction chain [Kalmár, 1939]
  - Very hard to mechanize in Coq [Kirst and Larchey-Wendling, 2020]
- Our contribution: A single reduction
  - Straightforward mechanisation
  - No additional axioms

Diophantine constraints

Definition (UDC)

For $l : \mathscr{L}(\mathcal{V}^3)$, l has property UDC iff

$$\exists \rho^{\mathcal{V} \to \mathbb{N}}, \forall (x, y, z) \in l, 1 + \rho x + (\rho y)^2 = \rho z$$

l is a list of certain Diophantine equations over $\mathbb{N}$. Is there a satisfying assignment?

Diophantine constraints

Definition (UDC)

For $l : \mathscr{L}(\mathcal{V}^3)$, l has property UDC iff

$$\exists \rho^{\mathcal{V} \to \mathbb{N}}, \forall (x, y, z) \in l, 1 + \rho x + (\rho y)^2 = \rho z$$

l is a list of certain Diophantine equations over $\mathbb{N}$. Is there a satisfying assignment?

Diophantine constraints

Definition (UDC)

For $l : \mathscr{L}(\mathcal{V}^3)$, l has property UDC iff

$$\exists \rho^{\mathcal{V} \to \mathbb{N}}, \forall (x, y, z) \in l, 1 + \rho x + (\rho y)^2 = \rho z$$

l is a list of certain Diophantine equations over $\mathbb{N}$. Is there a satisfying assignment?

- Satisfiability of Diophantine equations in $\mathbb{N}$ is undecidable [Matiyasevich, 1970]

Diophantine constraints

Definition (UDC)

For $l : \mathscr{L}(\mathcal{V}^3)$, l has property UDC iff

$$\exists \rho^{\mathcal{V} \to \mathbb{N}}, \forall (x, y, z) \in l, 1 + \rho x + (\rho y)^2 = \rho z$$

l is a list of certain Diophantine equations over $\mathbb{N}$. Is there a satisfying assignment?

- Satisfiability of Diophantine equations in $\mathbb{N}$ is undecidable [Matiyasevich, 1970]
- UDC is also undecidable

Diophantine constraints

Definition (UDC)

For $l : \mathscr{L}(\mathcal{V}^3)$, l has property UDC iff

$$\exists \rho^{\mathcal{V}\to\mathbb{N}}, \forall (x, y, z) \in l, 1 + \rho x + (\rho y)^2 = \rho z$$

l is a list of certain Diophantine equations over $\mathbb{N}$. Is there a satisfying assignment?

- Satisfiability of Diophantine equations in $\mathbb{N}$ is undecidable [Matiyasevich, 1970]
- UDC is also undecidable
- UDC mechanized in Coq [Larchey-Wendling and Forster, 2019]

Our new constraint $UDPC$

Definition ($UDPC$)

For $(x, y)^{\mathbb{N}^2}, (a, b)^{\mathbb{N}^2}$, we define $(x, y)\#(a, b)$ iff

$$a = x + y + 1 \text{ and } b + b = y^2 + y$$

A list $l : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right)$ has property $UDPC$ iff

$$\exists \rho^{\mathcal{V} \to \mathbb{N}}, \forall ((x, y), (a, b)) \in l, (\rho x, \rho y)\#(\rho a, \rho b)$$

Our new constraint $UDPC$

Definition ($UDPC$)

For $(x, y)^{\mathbb{N}^2}, (a, b)^{\mathbb{N}^2}$, we define $(x, y)\#(a, b)$ iff

$$a = x + y + 1 \text{ and } b + b = y^2 + y$$

A list $l : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right)$ has property $UDPC$ iff

$$\exists \rho^{\mathcal{V} \to \mathbb{N}}, \forall ((x, y), (a, b)) \in l, (\rho x, \rho y)\#(\rho a, \rho b)$$

Properties:

- Undecidability mechanized in Coq ✓

Our new constraint $UDPC$

Definition ($UDPC$)

For $(x, y)^{\mathbb{N}^2}, (a, b)^{\mathbb{N}^2}$, we define $(x, y)\#(a, b)$ iff

$$a = x + y + 1 \text{ and } b + b = y^2 + y$$

A list $l : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right)$ has property $UDPC$ iff

$$\exists \rho^{\mathcal{V} \to \mathbb{N}}, \forall ((x, y), (a, b)) \in l, (\rho x, \rho y)\#(\rho a, \rho b)$$

Properties:

- Undecidability mechanized in Coq ✓
- Structurality:
  - $(x, y)\#(a, b) \Rightarrow (x + 1, y)\#(a + 1, b)$
  - $(x, y)\#(a, b) \Rightarrow (x, y + 1)\#(a + 1, y + b + 1)$

FOL standard model

Idea: Synthesize a FOL description of $\#$

FOL standard model

Idea: Synthesize a FOL description of $\#$

- Goal: Signature only has $\#$

FOL standard model

Idea: Synthesize a FOL description of $\#$

- Goal: Signature only has $\#$
- Domain of discourse: $\mathbb{N} \cup \mathbb{N}^2$

FOL standard model

Idea: Synthesize a FOL description of #

- Goal: Signature only has #
- Domain of discourse: $\mathbb{N} \cup \mathbb{N}^2$
- Extend # to numbers:
  - $n \# m :\Leftrightarrow n = m$
  - $n \#(l, r) :\Leftrightarrow n = l$
  - $(l, r) \# n :\Leftrightarrow r = n$

FOL standard model

Idea: Synthesize a FOL description of $\#$

- Goal: Signature only has $\#$
- Domain of discourse: $\mathbb{N} \cup \mathbb{N}^2$
- Extend $\#$ to numbers:
  - $n\#m :\Leftrightarrow n = m$
  - $n\#(l, r) :\Leftrightarrow n = l$
  - $(l, r)\#n :\Leftrightarrow r = n$
- $\mathbb{N} \cup \mathbb{N}^2; \#$ is standard model

FOL standard model

Idea: Synthesize a FOL description of $\#$

- Goal: Signature only has $\#$
- Domain of discourse: $\mathbb{N} \cup \mathbb{N}^2$
- Extend $\#$ to numbers:
  - $n\#m :\Leftrightarrow n = m$
  - $n\#(l, r) :\Leftrightarrow n = l$
  - $(l, r)\#n :\Leftrightarrow r = n$
- $\mathbb{N} \cup \mathbb{N}^2; \#$ is standard model
- Task: Find first-order axioms for $\#$

FOL axiomatization

1. FOL Syntactic sugar:

FOL axiomatization

1. FOL Syntactic sugar:
   - $N\ k := k \# k$
     - $N\ k$ in standard model: k is a number.

FOL axiomatization

1. FOL Syntactic sugar:
   - $N\,k := k\#k$
   - $P'\,k := k\#k \to \bot$
     - $P'\,k$ in standard model: k is a pair.

FOL axiomatization

1. FOL Syntactic sugar:
   - $N\,k := k\#k$
   - $P'\,k := k\#k \to \bot$
   - $P\,p\,l\,r := P'\,p \wedge N\,l \wedge N\,r \wedge l\#p \wedge p\#r$
     - $P\,p\,l\,r$ in standard model: p is the pair (l, r)

FOL axiomatization

1. FOL Syntactic sugar:
   - $N\,k := k\#k$
   - $P'\,k := k\#k \to \bot$
   - $P\,p\,l\,r := P'\,p \land N\,l \land N\,r \land l\#p \land p\#r$
   - $(a, b)\#(c, d) := \exists p\,q, P\,p\,a\,b \land P\,q\,c\,d \land p\#q$
     - Necessary since we cannot simply construct pairs

FOL axiomatization

1. FOL Syntactic sugar:
   - $N\,k := k\#k$
   - $P'\,k := k\#k \to \bot$
   - $P\,p\,l\,r := P'\,p \wedge N\,l \wedge N\,r \wedge l\#p \wedge p\#r$
   - $(a,b)\#(c,d) := \exists p\,q, P\,p\,a\,b \wedge P\,q\,c\,d \wedge p\#q$
2. FOL Axioms:

FOL axiomatization

1. FOL Syntactic sugar:
   - $N\,k := k\#k$
   - $P'\,k := k\#k \to \bot$
   - $P\,p\,l\,r := P'\,p \wedge N\,l \wedge N\,r \wedge l\#p \wedge p\#r$
   - $(a,b)\#(c,d) := \exists p\,q, P\,p\,a\,b \wedge P\,q\,c\,d \wedge p\#q$
2. FOL Axioms:
   - $N\,0$
     - 0 is a number

FOL axiomatization

1. FOL Syntactic sugar:
   - $N\, k := k\#k$
   - $P'\, k := k\#k \to \bot$
   - $P\, p\, l\, r := P'\, p \wedge N\, l \wedge N\, r \wedge l\#p \wedge p\#r$
   - $(a, b)\#(c, d) := \exists p\, q, P\, p\, a\, b \wedge P\, q\, c\, d \wedge p\#q$
2. FOL Axioms:
   - $N\, 0$
   - $\varphi_1 := \forall x, \exists a, (x, 0)\#(a, 0)$
     - All numbers have successors

FOL axiomatization

1. FOL Syntactic sugar:
   - $N\,k := k\#k$
   - $P'\,k := k\#k \to \bot$
   - $P\,p\,l\,r := P'\,p \wedge N\,l \wedge N\,r \wedge l\#p \wedge p\#r$
   - $(a, b)\#(c, d) := \exists p\,q, P\,p\,a\,b \wedge P\,q\,c\,d \wedge p\#q$
2. FOL Axioms:
   - $N\,0$
   - $\varphi_1 := \forall x, \exists a, (x, 0)\#(a, 0)$
   - $\varphi_2 := \forall abcxya'y', (x, y)\#(a, b) \to (b, y)\#(c, b) \to (a, 0)\#(a', 0) \to (y, 0)\#(y', 0) \to (x, y')\#(a', c)$
     - Characerizes $\#$
     - Reformulation of $(x, y)\#(a, b) \Rightarrow (x, y + 1)\#(a + 1, y + b + 1)$

The reduction

Let $h = [((x_1, y_1), (a_1, b_1)), \ldots, ((x_n, y_n), (a_n, b_n))]$ an instance of $UDPC$.

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right) \to FOL_{\{\},\{\#\}}$

The reduction

Let $h = [((x_1, y_1), (a_1, b_1)), \ldots, ((x_n, y_n), (a_n, b_n))]$ an instance of $UDPC$.

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right) \to FOL_{\{\},\{\#\}}$
  - $F(h) = \forall 0, N\,0 \to \varphi_1 \to \varphi_2 \to F'(h)$

The reduction

Let $h = [((x_1, y_1), (a_1, b_1)), \ldots, ((x_n, y_n), (a_n, b_n))]$ an instance of $UDPC$.

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right) \to FOL_{\{\},\{\#\}}$
  - $F(h) = \forall 0, N\,0 \to \varphi_1 \to \varphi_2 \to F'(h)$
  - $F'(h) = \displaystyle\exists_{v \in \mathcal{V}(h)} \bigwedge_{i=1}^{n} (x_i, y_i)\#(a_i, b_i)$

The reduction

Let $h = [((x_1, y_1), (a_1, b_1)), \ldots, ((x_n, y_n), (a_n, b_n))]$ an instance of $UDPC$.

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right) \to FOL_{\{\},\{\#\}}$
  - $F(h) = \forall 0, N\,0 \to \varphi_1 \to \varphi_2 \to F'(h)$
  - $F'(h) = \displaystyle\exists_{v \in \mathcal{V}(h)} \bigwedge_{i=1}^{n} (x_i, y_i)\#(a_i, b_i)$
  - Note: 0 is a variable

The reduction

Let $h = [((x_1, y_1), (a_1, b_1)), \ldots, ((x_n, y_n), (a_n, b_n))]$ an instance of $UDPC$.

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right) \to FOL_{\{\},\{\#\}}$
  - $F(h) = \forall 0, N\,0 \to \varphi_1 \to \varphi_2 \to F'(h)$
  - $F'(h) = \displaystyle\coprod_{v \in \mathcal{V}(h)} \bigwedge_{i=1}^{n} (x_i, y_i)\#(a_i, b_i)$
  - Note: 0 is a variable
- Reduction soundness: $h \in UDPC$ if $F(h)$ valid
  - $F(h)$ holds in the standard model

The reduction

Let $h = [((x_1, y_1), (a_1, b_1)), \ldots, ((x_n, y_n), (a_n, b_n))]$ an instance of $UDPC$.

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right) \to FOL_{\{\},\{\#\}}$
  - $F(h) = \forall 0, N\,0 \to \varphi_1 \to \varphi_2 \to F'(h)$
  - $F'(h) = \displaystyle\coprod_{v \in \mathcal{V}(h)} \bigwedge_{i=1}^{n} (x_i, y_i)\#(a_i, b_i)$
  - Note: 0 is a variable
- Reduction soundness: $h \in UDPC$ if $F(h)$ valid
  - $F(h)$ holds in the standard model
- Reduction completeness: $F(h)$ valid if $h \in UDPC$
  - Abstract proof in arbitrary model
  - Uses axioms φ_1, φ_2, etc

The reduction

Let $h = [((x_1, y_1), (a_1, b_1)), \ldots, ((x_n, y_n), (a_n, b_n))]$ an instance of $UDPC$.

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 * \mathcal{V}^2\right) \rightarrow FOL_{\{\},\{\#\}}$
  - $F(h) = \forall 0, N\,0 \rightarrow \varphi_1 \rightarrow \varphi_2 \rightarrow F'(h)$
  - $F'(h) = \underset{v \in \mathcal{V}(h)}{\boxed{\exists}} \bigwedge_{i=1}^{n} (x_i, y_i)\#(a_i, b_i)$
  - Note: 0 is a variable
- Reduction soundness: $h \in UDPC$ if $F(h)$ valid
  - $F(h)$ holds in the standard model
- Reduction completeness: $F(h)$ valid if $h \in UDPC$
  - Abstract proof in arbitrary model
  - Uses axioms φ_1, φ_2, etc
- Similar argument for int. provability.

Refinement

Can we make the signature even more minimal?

- Minimal signature ✓

Refinement

Can we make the signature even more minimal?

- Minimal signature ✓
- Minimal logical connectives:
  - Double negation translation:
    Replace $\exists\varphi$ with $\neg\forall\neg\varphi$ etc.

Refinement

Can we make the signature even more minimal?

- Minimal signature ✓
- Minimal logical connectives:
  - Double negation translation:
    Replace $\exists\varphi$ with $\neg\forall\neg\varphi$ etc.
- Negation:
  - Friedman translation:
    Replace $\perp$ with $c_1 \# c_2$ for globally fixed c_1, c_2.

Refinement

Can we make the signature even more minimal?

- Minimal signature ✓
- Minimal logical connectives:
  - Double negation translation:
    Replace $\exists\varphi$ with $\neg\forall\neg\varphi$ etc.
- Negation:
  - Friedman translation:
    Replace $\bot$ with $c_1\#c_2$ for globally fixed c_1, c_2.
- $F(h)$ only uses $\#, \forall, \rightarrow$.
- Stronger undecidability result

Conclusion

Reductions formalized for:

- Validity
- int. Provability

$\Rightarrow$ int. Satisfiability

$\Rightarrow$ Kripke validity/ int. satisfiability

About 1300 LoC

Conclusion

Reductions formalized for:

- Validity
- int. Provability
- $\Rightarrow$ int. Satisfiability
- $\Rightarrow$ Kripke validity/ int. satisfiability

About 1300 LoC

Future plans:

- Finite satisfiability
- look at classical provability
- Finish and refine Coq formalization

References

[Church, 1936] Church, A. (1936). A note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1(1):40–41.

[Forster et al., 2020] Forster, Y., Dominique, Larchey-Wendling, Andrej, Dudenhefner, Heiter, E., Kirst, D., Kunze, F., and Smolka, G. (2020). A coq library of undecidable problems. CoqPL 20.

[Forster et al., 2019] 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.

[Kalmár, 1939] Kalmár, L. (1939). On the reduction of the decision problem. First paper. Ackermann prefix, a single binary predicate. *Journal of Symbolic Logic*, 4(1):1–9.

[Kirst and Larchey-Wendling, 2020] Kirst, D. and Larchey-Wendling, D. (2020). Trakhtenbrot's theorem in coq. In Peltier, N. and Sofronie-Stokkermans, V., editors, *Automated Reasoning*, pages 79–96, Cham. Springer International Publishing.

[Larchey-Wendling and Forster, 2019] Larchey-Wendling, D. and Forster, Y. (2019). Hilbert's Tenth Problem in Coq. 4th International Conference on Formal Structures for Computation and Deduction.

[Löwenheim, 1915] Löwenheim, L. (1915). Über Möglichkeiten im Relativkalkül. *Mathematische Annalen*, 76:447–470.

[Matiyasevich, 1970] Matiyasevich, Y. V. (1970). Enumerable sets are diophantine. *Doklady Akademii Nauk SSSR*, 191:279–282.

[Turing, 1936] Turing, A. M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265.

FOL deduction system

$$\text{II}\ \frac{\varphi, A \vdash \psi}{A \vdash \varphi \to \psi} \qquad \text{IE}\ \frac{A \vdash \varphi \to \psi \qquad A, \varphi \vdash \psi}{A \vdash \psi} \qquad \text{Ctx}\ \frac{\varphi \in A}{A \vdash \varphi}$$

$$\forall\text{I}\ \frac{A \vdash \varphi \qquad x \text{ free in } A}{A \vdash \forall x, \varphi} \qquad \forall\text{E}\ \frac{A \vdash \forall x, \varphi}{A \vdash \varphi} \qquad \bot\text{E}\ \frac{A \vdash \bot}{A \vdash \varphi}$$

For a classical deduction system, we also assume Pierce's law:

$$\text{Pc}\ \frac{}{A \vdash_c (((\varphi \to \psi) \to \varphi) \to \varphi}$$

In the mechanization, de Bruijn indices are used

Reduction completeness

Show two lemmata:

1. $\forall n \in \mathbb{N}, \exists f : \mathbb{N} \to D$, which contains the "representation chain" of the first n numbers in D.
   - Induction on n
   - Base case: $N\,0$
   - Induction step: Axiom φ_1
2. $\forall xyab$, if $(x, y)\#(a, b)$, and if we have a chain up to $\max\{\varphi x, \varphi y, \varphi a, \varphi b\}$ represented by f, then $(fx, fy)\#(fa, fb)$ holds.
   - Induction on y, with x, a, b free.
   - Base case: Lemma 1
   - Induction step: Axiom φ_2, IH for x, a, b; b, c, b and Lemma 1 for a, y.

Showing $UDPC$ undecidable

Known-undecidable constraint problem:

$$1 + x + y^2 = z$$

New constraints for each such constraint:

$$(a, a)\#(b, t_1)$$
$$(c, y)\#(b, a)$$
$$(c, x)\#(z, t_2)$$

a, b, c, t_1, t_2 are fresh

Details of the mechanisation

- FOL mechanisation uses de Bruijn indices
  - Formulas are hard to read
  - Keeping track of all the indices is hard
  - Deduction system manipulates indices in $\forall I$ and $\forall E$ rules
- Reasoning about the chain in the deduction system is hard
  - We can not have function $\mathbb{N} \to D$
  - Bad idea: have representation at fixed indices
  - Better idea: data structure encoding de Bruijn indices of representations