The Undecidability of First-Order Logic over Small Signatures

Final Bachelor Talk

Advisors: Andrej Dudenhefner, Dominik Kirst

Supervisor: Prof. Gert Smolka

Johannes Hostert

October 14, 2021, Saarland University

First-Order Logic

Quantifying over individuals: $\forall ab.a + b = b + a$

- Foundation of mathematics

First-Order Logic

Quantifying over individuals: $\forall ab. a + b = b + a$

- Foundation of mathematics
- "Truth" of FOL formulas undecidable in general [Church, 1936, Turing, 1936]

First-Order Logic

Quantifying over individuals: $\forall ab.a + b = b + a$

- Foundation of mathematics
- "Truth" of FOL formulas undecidable in general [Church, 1936, Turing, 1936]
- Can we simplify FOL formulas to recover decidability?
  - Restrict signature?
  - Restrict logical connectives?
  - Restrict semantics?

First-Order Logic

Quantifying over individuals: $\forall ab. a + b = b + a$

- Foundation of mathematics
- "Truth" of FOL formulas undecidable in general [Church, 1936, Turing, 1936]
- Can we simplify FOL formulas to recover decidability?
  - Restrict signature?
  - Restrict logical connectives?
  - Restrict semantics?
- Semantics: what does "truth" mean?
  - Satisfaction in all models
  - Satisfaction in finite models
  - Abstract deduction system

(Un)decidable fragments of FOL

Decidability results:

- FOL decidable for unary signature [Löwenheim, 1915]

(Un)decidable fragments of FOL

Decidability results:

- FOL decidable for unary signature [Löwenheim, 1915]
- FOL undecidable once there is an at least binary relation
  - Even when restricting to $\forall, \rightarrow, (\bot)$ logical fragment

(Un)decidable fragments of FOL

Decidability results:

- FOL decidable for unary signature [Löwenheim, 1915]
- FOL undecidable once there is an at least binary relation
  - Even when restricting to $\forall, \to, (\bot)$ logical fragment

Paper	Binary signature	Small fragment	Coq	Reduction
[Church, 1936]	$\times$	$\times$	$\times$	λ calculus
[Turing, 1936]	$\times$	$\times$	$\times$	TM

(Un)decidable fragments of FOL

Decidability results:

- FOL decidable for unary signature [Löwenheim, 1915]
- FOL undecidable once there is an at least binary relation
  - Even when restricting to $\forall, \rightarrow, (\bot)$ logical fragment

Paper	Binary signature	Small fragment	Coq	Reduction
[Church, 1936]	$\times$	$\times$	$\times$	λ calculus
[Turing, 1936]	$\times$	$\times$	$\times$	TM
[Kalmár, 1937]	$\checkmark$	$\times$	$\times$	signature compression
[Gentzen, 1936]		$\checkmark$[1]	$\times$	double negation

[1] $\forall, \rightarrow, \wedge, \bot$

(Un)decidable fragments of FOL

Decidability results:

- FOL decidable for unary signature [Löwenheim, 1915]
- FOL undecidable once there is an at least binary relation
  - Even when restricting to $\forall, \rightarrow, (\bot)$ logical fragment

Paper	Binary signature	Small fragment	Coq	Reduction
[Church, 1936]	$\times$	$\times$	$\times$	λ calculus
[Turing, 1936]	$\times$	$\times$	$\times$	TM
[Kalmár, 1937]	$\checkmark$	$\times$	$\times$	signature compression
[Gentzen, 1936]		$\checkmark$[1]	$\times$	double negation
[Forster et al., 2019]	$\times$	$\checkmark$	$\checkmark$	PCP
[Kirst and Hermes, 2021]	$\checkmark$	$\times$	$\checkmark$	ZF

[1] $\forall, \rightarrow, \wedge, \bot$

(Un)decidable fragments of FOL

Decidability results:

- FOL decidable for unary signature [Löwenheim, 1915]
- FOL undecidable once there is an at least binary relation
  - Even when restricting to $\forall, \rightarrow, (\bot)$ logical fragment

Paper	Binary signature	Small fragment	Coq	Reduction
[Church, 1936]	$\times$	$\times$	$\times$	λ calculus
[Turing, 1936]	$\times$	$\times$	$\times$	TM
[Kalmár, 1937]	$\checkmark$	$\times$	$\times$	signature compression
[Gentzen, 1936]		$\checkmark$[1]	$\times$	double negation
[Forster et al., 2019]	$\times$	$\checkmark$	$\checkmark$	PCP
[Kirst and Hermes, 2021]	$\checkmark$	$\times$	$\checkmark$	ZF
This thesis	$\checkmark$	$\checkmark$	$\checkmark$	H10 variant

[1] $\forall, \rightarrow, \wedge, \bot$

Problems in FOL

Show undecidability of the following problems:

- Validity VAL: Is formula satisfied by all Tarski models?
- Satisfiability SAT: Is there a Tarski model satisfying formula?

Problems in FOL

Show undecidability of the following problems:

- Validity VAL: Is formula satisfied by all Tarski models?
- Satisfiability SAT: Is there a Tarski model satisfying formula?
- Provability PRV: Is there a proof in intuitionistic ND system?

Problems in FOL

Show undecidability of the following problems:

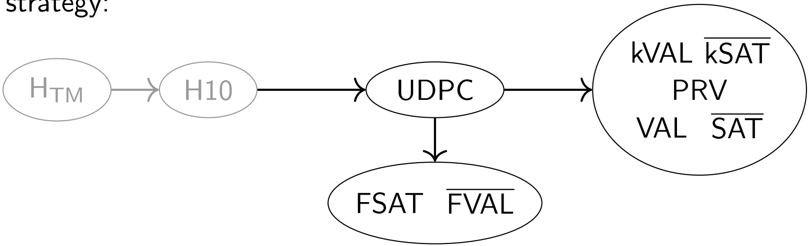
- Validity VAL: Is formula satisfied by all Tarski models?
- Satisfiability SAT: Is there a Tarski model satisfying formula?
- Provability PRV: Is there a proof in intuitionistic ND system?
- Finite satisfiability FSAT: Is formula satisfied by finite model?
- Finite validity FVAL, Kripke model semantics kSAT, kVAL

Problems in FOL

Show undecidability of the following problems:

- Validity VAL: Is formula satisfied by all Tarski models?
- Satisfiability SAT: Is there a Tarski model satisfying formula?
- Provability PRV: Is there a proof in intuitionistic ND system?
- Finite satisfiability FSAT: Is formula satisfied by finite model?
- Finite validity FVAL, Kripke model semantics kSAT, kVAL

Reduction strategy:



Uniform Diophantine Pair Constraints

$$\wr : \mathbb{N}^2 \times \mathbb{N}^2 \to \mathbb{P}$$
$$(a, b) \wr (c, d) := a + b + 1 = c \land d + d = b^2 + b$$

Uniform Diophantine Pair Constraints

$$\wr : \mathbb{N}^2 \times \mathbb{N}^2 \to \mathbb{P}$$
$$(a,b) \wr (c,d) := a + b + 1 = c \land d + d = b^2 + b$$

Characterizing properties

- $(a,0) \wr (a+1,0)$
- $(a,b'+1) \wr (c'+1,d)$ iff $(a,b') \wr (c',d') \land (d',b') \wr (d,d')$

Uniform Diophantine Pair Constraints

$$\wr : \mathbb{N}^2 \times \mathbb{N}^2 \to \mathbb{P}$$

$$\text{Base}\ \frac{}{(a,0) \wr (a+1,0)}$$

$$\text{Step}\ \frac{(a,b') \wr (c',d') \qquad (d',b') \wr (d,d') \qquad (b',0) \wr (b,0) \qquad (c',0) \wr (c,0)}{(a,b) \wr (c,d)}$$

Uniform Diophantine Pair Constraints

$$\wr : \mathbb{N}^2 \times \mathbb{N}^2 \to \mathbb{P}$$

$$\text{Base}\ \frac{}{(a, 0) \wr (a+1, 0)}$$

$$\text{Step}\ \frac{(a, b') \wr (c', d') \qquad (d', b') \wr (d, d') \qquad (b', 0) \wr (b, 0) \qquad (c', 0) \wr (c, 0)}{(a, b) \wr (c, d)}$$

UDPC : $\mathscr{L}(\mathcal{V}^2 \times \mathcal{V}^2) \to \mathbb{P}$ – Does list of constraint equations have solution?

Uniform Diophantine Pair Constraints

$$\wr : \mathbb{N}^2 \times \mathbb{N}^2 \to \mathbb{P}$$

$$\text{Base}\ \frac{}{(a,0) \wr (a+1,0)}$$

$$\text{Step}\ \frac{(a,b') \wr (c',d') \qquad (d',b') \wr (d,d') \qquad (b',0) \wr (b,0) \qquad (c',0) \wr (c,0)}{(a,b) \wr (c,d)}$$

UDPC : $\mathscr{L}(\mathcal{V}^2 \times \mathcal{V}^2) \to \mathbb{P}$ – Does list of constraint equations have solution?

- UDPC is undecidable
  - Reduce from Diophantine constraints satisfiability [Matiyasevich, 1970]
  - Source problem already mechanized [Larchey-Wendling and Forster, 2019]

Uniform Diophantine Pair Constraints

$$\wr : \mathbb{N}^2 \times \mathbb{N}^2 \to \mathbb{P}$$

$$\text{Base}\ \frac{}{(a,0) \wr (a+1,0)}$$

$$\text{Step}\ \frac{(a,b') \wr (c',d') \qquad (d',b') \wr (d,d') \qquad (b',0) \wr (b,0) \qquad (c',0) \wr (c,0)}{(a,b) \wr (c,d)}$$

UDPC : $\mathscr{L}(\mathcal{V}^2 \times \mathcal{V}^2) \to \mathbb{P}$ – Does list of constraint equations have solution?

- UDPC is undecidable
  - Reduce from Diophantine constraints satisfiability [Matiyasevich, 1970]
  - Source problem already mechanized [Larchey-Wendling and Forster, 2019]
- UDPC and $\wr$ originally due to Andrej Dudenhefner

Mechanizing VAL undecidability

Idea: Reduce from UDPC

Mechanizing VAL undecidability

Idea: Reduce from UDPC

- Construct formula $F(h)$ with binary relation $\wr$ valid iff h has solution

Mechanizing VAL undecidability

Idea: Reduce from UDPC

- Construct formula $F(h)$ with binary relation $\wr\!\wr$ valid iff h has solution
- Standard model: $M = \mathbb{N} + \mathbb{N}^2$, interpret $l \wr\!\wr r$ as

l \ r	$y : \mathbb{N}$	$(c, d) : \mathbb{N}^2$
$x : \mathbb{N}$	$x = y$	$x = c$
$(a, b) : \mathbb{N}^2$	$y = b$	$(a, b) \wr (c, d)$

Mechanizing VAL undecidability

Idea: Reduce from UDPC

- Construct formula $F(h)$ with binary relation $\wr\wr$ valid iff h has solution
- Standard model: $M = \mathbb{N} + \mathbb{N}^2$, interpret $l \wr\wr r$ as

l \ r	$y : \mathbb{N}$	$(c, d) : \mathbb{N}^2$
$x : \mathbb{N}$	$x = y$	$x = c$
$(a, b) : \mathbb{N}^2$	$y = b$	$(a, b) \wr (c, d)$

- F formalizes characterizing properties/constructors of $\wr$ in binary FOL
- $F(h)$ translates constraints h into FOL

Mechanizing VAL undecidability

Idea: Reduce from UDPC

- Construct formula $F(h)$ with binary relation $\wr\wr$ valid iff h has solution
- Standard model: $M = \mathbb{N} + \mathbb{N}^2$, interpret $l \wr\wr r$ as

l \ r	$y : \mathbb{N}$	$(c, d) : \mathbb{N}^2$
$x : \mathbb{N}$	$x = y$	$x = c$
$(a, b) : \mathbb{N}^2$	$y = b$	$(a, b) \wr (c, d)$

- F formalizes characterizing properties/constructors of $\wr$ in binary FOL
- $F(h)$ translates constraints h into FOL
- Correctness:
  - UDPC $h \to$ VAL$(F\,h)$: Construct proof in abstract model
  - VAL$(F\,h) \to$ UDPC h: Specialize standard model

Mechanizing VAL undecidability

Idea: Reduce from UDPC

- Construct formula $F(h)$ with binary relation $\wr\wr$ valid iff h has solution
- Standard model: $M = \mathbb{N} + \mathbb{N}^2$, interpret $l \wr\wr r$ as

l \ r	$y : \mathbb{N}$	$(c, d) : \mathbb{N}^2$
$x : \mathbb{N}$	$x = y$	$x = c$
$(a, b) : \mathbb{N}^2$	$y = b$	$(a, b) \wr (c, d)$

- F formalizes characterizing properties/constructors of $\wr$ in binary FOL
- $F(h)$ translates constraints h into FOL
- Correctness:
  - UDPC $h \rightarrow$ VAL$(F\, h)$: Construct proof in abstract model
  - VAL$(F\, h) \rightarrow$ UDPC h: Specialize standard model

$\Rightarrow$ VAL is undecidable for binary signature

Sharper results for VAL

Restrict the admissible logical operators to $\forall, \rightarrow$

Sharper results for VAL

Restrict the admissible logical operators to $\forall, \rightarrow$

1. Double negation translation [Gödel, 1933, Gentzen, 1936]
   - Replace $\exists$ by $\neg\forall\neg$ etc.
   - Not equivalent in intuitionistic meta-theory

Sharper results for VAL

Restrict the admissible logical operators to $\forall, \to$

1. Double negation translation [Gödel, 1933, Gentzen, 1936]
   - Replace $\exists$ by $\neg\forall\neg$ etc.
   - Not equivalent in intuitionistic meta-theory
2. Friedman A-translation [Friedman, 1978]
   - Replace $\perp$ by some formula Q...

Sharper results for VAL

Restrict the admissible logical operators to $\forall, \rightarrow$

1. Double negation translation [Gödel, 1933, Gentzen, 1936]
   - Replace $\exists$ by $\neg\forall\neg$ etc.
   - Not equivalent in intuitionistic meta-theory
2. Friedman A-translation [Friedman, 1978]
   - Replace $\bot$ by some formula Q...
   - such that interpretation of Q is UDPC h...

Sharper results for VAL

Restrict the admissible logical operators to $\forall, \rightarrow$

1. Double negation translation [Gödel, 1933, Gentzen, 1936]
   - Replace $\exists$ by $\neg\forall\neg$ etc.
   - Not equivalent in intuitionistic meta-theory
2. Friedman A-translation [Friedman, 1978]
   - Replace $\bot$ by some formula Q...
   - such that interpretation of Q is UDPC h...
   - without introducing additional relation symbols

Sharper results for VAL

Restrict the admissible logical operators to $\forall, \rightarrow$

1. Double negation translation [Gödel, 1933, Gentzen, 1936]
   - Replace $\exists$ by $\neg\forall\neg$ etc.
   - Not equivalent in intuitionistic meta-theory
2. Friedman A-translation [Friedman, 1978]
   - Replace $\perp$ by some formula Q...
   - such that interpretation of Q is UDPC h...
   - without introducing additional relation symbols

- Approach known from [Forster et al., 2019]

Sharper results for VAL

Restrict the admissible logical operators to $\forall, \rightarrow$

1. Double negation translation [Gödel, 1933, Gentzen, 1936]
   - Replace $\exists$ by $\neg\forall\neg$ etc.
   - Not equivalent in intuitionistic meta-theory
2. Friedman A-translation [Friedman, 1978]
   - Replace $\bot$ by some formula Q...
   - such that interpretation of Q is UDPC h...
   - without introducing additional relation symbols

- Approach known from [Forster et al., 2019]

In summary:

- VAL undecidable for binary signature over $\forall, \rightarrow$-fragment
- SAT also undecidable for binary signature $\forall, \rightarrow, \bot$-fragment

Sharper results for VAL

Restrict the admissible logical operators to $\forall, \rightarrow$

1. Double negation translation [Gödel, 1933, Gentzen, 1936]
   - Replace $\exists$ by $\neg\forall\neg$ etc.
   - Not equivalent in intuitionistic meta-theory
2. Friedman A-translation [Friedman, 1978]
   - Replace $\bot$ by some formula Q...
   - such that interpretation of Q is UDPC h...
   - without introducing additional relation symbols

- Approach known from [Forster et al., 2019]

In summary:

- VAL undecidable for binary signature over $\forall, \rightarrow$-fragment
- SAT also undecidable for binary signature $\forall, \rightarrow, \bot$-fragment

These are the minimal results

Provability

Provability

- In a classical meta-theory: $\mathsf{VAL}\,\varphi \Rightarrow \mathsf{PRV}_c\,\varphi$ [Gödel, 1930]

Provability

- In a classical meta-theory: $\mathsf{VAL}\,\varphi \Rightarrow \mathsf{PRV}_c\,\varphi$ [Gödel, 1930]
- In Coq: $\mathsf{VAL}\,\varphi \Rightarrow \neg\neg\mathsf{PRV}_c\,\varphi$ for $\forall, \rightarrow, \bot$-fragment [Forster et al., 2020]

Provability

- In a classical meta-theory: $\mathsf{VAL}\,\varphi \Rightarrow \mathsf{PRV}_c\,\varphi$ [Gödel, 1930]
- In Coq: $\mathsf{VAL}\,\varphi \Rightarrow \neg\neg\mathsf{PRV}_c\,\varphi$ for $\forall, \rightarrow, \bot$-fragment [Forster et al., 2020]
- Soundness $\mathsf{PRV}\,\varphi \Rightarrow \mathsf{VAL}\,\varphi$ still holds

Provability

- In a classical meta-theory: $\mathsf{VAL}\,\varphi \Rightarrow \mathsf{PRV}_c\,\varphi$ [Gödel, 1930]
- In Coq: $\mathsf{VAL}\,\varphi \Rightarrow \neg\neg\mathsf{PRV}_c\,\varphi$ for $\forall, \rightarrow, \bot$-fragment [Forster et al., 2020]
- Soundness $\mathsf{PRV}\,\varphi \Rightarrow \mathsf{VAL}\,\varphi$ still holds
- Dedicated reduction to (intuitionistic) PRV using same reduction function
- Correctness:
  - $\mathsf{UDPC}\,h \rightarrow \mathsf{PRV}(F\,h)$: Construct abstract proof in natural deduction system
  - $\mathsf{PRV}(F\,h) \rightarrow \mathsf{UDPC}\,h$: Soundness and standard model

Provability

- In a classical meta-theory: $\mathsf{VAL}\,\varphi \Rightarrow \mathsf{PRV}_c\,\varphi$ [Gödel, 1930]
- In Coq: $\mathsf{VAL}\,\varphi \Rightarrow \neg\neg\mathsf{PRV}_c\,\varphi$ for $\forall, \rightarrow, \bot$-fragment [Forster et al., 2020]
- Soundness $\mathsf{PRV}\,\varphi \Rightarrow \mathsf{VAL}\,\varphi$ still holds
- Dedicated reduction to (intuitionistic) PRV using same reduction function
- Correctness:
  - $\mathsf{UDPC}\,h \rightarrow \mathsf{PRV}(F\,h)$: Construct abstract proof in natural deduction system
  - $\mathsf{PRV}(F\,h) \rightarrow \mathsf{UDPC}\,h$: Soundness and standard model
- Results:
  - PRV undecidable for binary signature over $\forall, \rightarrow$-fragment
  - Kripke semantics undecidable for binary signature over $\forall, \rightarrow, (\bot)$-fragment

Provability

- In a classical meta-theory: $\mathsf{VAL}\,\varphi \Rightarrow \mathsf{PRV}_c\,\varphi$ [Gödel, 1930]
- In Coq: $\mathsf{VAL}\,\varphi \Rightarrow \neg\neg\mathsf{PRV}_c\,\varphi$ for $\forall, \rightarrow, \bot$-fragment [Forster et al., 2020]
- Soundness $\mathsf{PRV}\,\varphi \Rightarrow \mathsf{VAL}\,\varphi$ still holds
- Dedicated reduction to (intuitionistic) PRV using same reduction function
- Correctness:
  - $\mathsf{UDPC}\,h \rightarrow \mathsf{PRV}(F\,h)$: Construct abstract proof in natural deduction system
  - $\mathsf{PRV}(F\,h) \rightarrow \mathsf{UDPC}\,h$: Soundness and standard model
- Results:
  - PRV undecidable for binary signature over $\forall, \rightarrow$-fragment
  - Kripke semantics undecidable for binary signature over $\forall, \rightarrow, (\bot)$-fragment
  - classical provability PRV_c similarly undecidable, assuming LEM

Finite Satisfiability

- Restrict Tarski models to finite types
  - Follow existing mechanization [Kirst and Larchey-Wendling, 2020]

Finite Satisfiability

- Restrict Tarski models to finite types
  - Follow existing mechanization [Kirst and Larchey-Wendling, 2020]
- Existing results:

Paper	Binary signature	Logical fragment	Coq	Method
[Kalmár, 1937]	?	$\times$	$\times$	signature compression

Finite Satisfiability

- Restrict Tarski models to finite types
  - Follow existing mechanization [Kirst and Larchey-Wendling, 2020]
- Existing results:

Paper	Binary signature	Logical fragment	Coq	Method
[Kalmár, 1937]	?	$\times$	$\times$	signature compression
[Trakhtenbrot, 1950]	$\times$	$\times$	$\times$	μ-recursive functions

Finite Satisfiability

- Restrict Tarski models to finite types
  - Follow existing mechanization [Kirst and Larchey-Wendling, 2020]
- Existing results:

Paper	Binary signature	Logical fragment	Coq	Method
[Kalmár, 1937]	?	$\times$	$\times$	signature compression
[Trakhtenbrot, 1950]	$\times$	$\times$	$\times$	μ-recursive functions
[Libkin, 2004]	($\checkmark$)	$\times$	$\times$	TM

Finite Satisfiability

- Restrict Tarski models to finite types
  - Follow existing mechanization [Kirst and Larchey-Wendling, 2020]
- Existing results:

Paper	Binary signature	Logical fragment	Coq	Method
[Kalmár, 1937]	?	×	×	signature compression
[Trakhtenbrot, 1950]	×	×	×	μ-recursive functions
[Libkin, 2004]	(✓)	×	×	TM
[Kirst and Larchey-Wendling, 2020]	✓	×	✓	PCP

Finite Satisfiability

- Restrict Tarski models to finite types
  - Follow existing mechanization [Kirst and Larchey-Wendling, 2020]
- Existing results:

Paper	Binary signature	Logical fragment	Coq	Method
[Kalmár, 1937]	?	×	×	signature compression
[Trakhtenbrot, 1950]	×	×	×	μ-recursive functions
[Libkin, 2004]	(✓)	×	×	TM
[Kirst and Larchey-Wendling, 2020]	✓	×	✓	PCP
This thesis	✓	✓	✓	H10 variant

Finite Satisfiability

- Restrict Tarski models to finite types
  - Follow existing mechanization [Kirst and Larchey-Wendling, 2020]
- Existing results:

Paper	Binary signature	Logical fragment	Coq	Method
[Kalmár, 1937]	?	×	×	signature compression
[Trakhtenbrot, 1950]	×	×	×	μ-recursive functions
[Libkin, 2004]	(✓)	×	×	TM
[Kirst and Larchey-Wendling, 2020]	✓	×	✓	PCP
This thesis	✓	✓	✓	H10 variant

- FSAT mechanization build "inversely" to previous reduction
  - Then: Encode given solution into model

Finite Satisfiability

- Restrict Tarski models to finite types
  - Follow existing mechanization [Kirst and Larchey-Wendling, 2020]
- Existing results:

Paper	Binary signature	Logical fragment	Coq	Method
[Kalmár, 1937]	?	×	×	signature compression
[Trakhtenbrot, 1950]	×	×	×	μ-recursive functions
[Libkin, 2004]	(✓)	×	×	TM
[Kirst and Larchey-Wendling, 2020]	✓	×	✓	PCP
This thesis	✓	✓	✓	H10 variant

- FSAT mechanization build "inversely" to previous reduction
  - Then: Encode given solution into model
  - Now: Given model encoding solution, extract it

Reduction for FSAT

Idea: Still reduce from UDPC

- We have to extract solution from arbitrary model

Reduction for FSAT

Idea: Still reduce from UDPC

- We have to extract solution from arbitrary model
  - Previous: axioms were resembling constructors

Reduction for FSAT

Idea: Still reduce from UDPC

- We have to extract solution from arbitrary model
  - Previous: axioms were resembling constructors
  - Now: axioms resemble eliminators

Reduction for FSAT

Idea: Still reduce from UDPC

- We have to extract solution from arbitrary model
  - Previous: axioms were resembling constructors
  - Now: axioms resemble eliminators
- Finite standard model: $M = \mathbb{N}_{\leq m} + \mathbb{N}^2_{\leq m}$
  - Reduction pecularities require encoding $\leq$ into interpretation

Reduction for FSAT

Idea: Still reduce from UDPC

- We have to extract solution from arbitrary model
  - Previous: axioms were resembling constructors
  - Now: axioms resemble eliminators
- Finite standard model: $M = \mathbb{N}_{\leq m} + \mathbb{N}^2_{\leq m}$
  - Reduction pecularities require encoding $\leq$ into interpretation
- Correctness:
  - UDPC $h \to$ FSAT$(F'\,h)$: Standard model
  - FSAT$(F'\,h) \to$ UDPC h: Extract solution from given model

Reduction for FSAT

Idea: Still reduce from UDPC

- We have to extract solution from arbitrary model
  - Previous: axioms were resembling constructors
  - Now: axioms resemble eliminators
- Finite standard model: $M = \mathbb{N}_{\leq m} + \mathbb{N}^2_{\leq m}$
  - Reduction pecularities require encoding $\leq$ into interpretation
- Correctness:
  - UDPC $h \to$ FSAT$(F'\,h)$: Standard model
  - FSAT$(F'\,h) \to$ UDPC h: Extract solution from given model

$\Rightarrow$ FSAT is undecidable for binary signature

FVAL and the small fragment

What about the $\forall, \rightarrow, \bot$-fragment?

FVAL and the small fragment

What about the $\forall, \to, \bot$-fragment?

- Finite models "behave classically": $M \vDash \varphi$ decidable

FVAL and the small fragment

What about the $\forall, \rightarrow, \bot$-fragment?

- Finite models "behave classically": $M \vDash \varphi$ decidable
- Double negation translation holds in general

FVAL and the small fragment

What about the $\forall, \rightarrow, \bot$-fragment?

- Finite models "behave classically": $M \vDash \varphi$ decidable
- Double negation translation holds in general
- FSAT is undecidable for binary signature over $\forall, \rightarrow, \bot$-fragment

What about FVAL?

FVAL and the small fragment

What about the $\forall, \rightarrow, \bot$-fragment?

- Finite models "behave classically": $M \vDash \varphi$ decidable
- Double negation translation holds in general
- FSAT is undecidable for binary signature over $\forall, \rightarrow, \bot$-fragment

What about FVAL?

- FVAL is undecidable for binary signature over $\forall, \rightarrow, \bot$-fragment
  - using $\neg F'(h)$ as reduction function

FVAL and the small fragment

What about the $\forall, \rightarrow, \bot$-fragment?

- Finite models "behave classically": $M \vDash \varphi$ decidable
- Double negation translation holds in general
- FSAT is undecidable for binary signature over $\forall, \rightarrow, \bot$-fragment

What about FVAL?

- FVAL is undecidable for binary signature over $\forall, \rightarrow, \bot$-fragment
  - using $\neg F'(h)$ as reduction function
- Conjecture: FVAL undecidable for binary signature over $\forall, \rightarrow$-fragment

FVAL and the small fragment

What about the $\forall, \rightarrow, \bot$-fragment?

- Finite models "behave classically": $M \vDash \varphi$ decidable
- Double negation translation holds in general
- FSAT is undecidable for binary signature over $\forall, \rightarrow, \bot$-fragment

What about FVAL?

- FVAL is undecidable for binary signature over $\forall, \rightarrow, \bot$-fragment
  - using $\neg F'(h)$ as reduction function
- Conjecture: FVAL undecidable for binary signature over $\forall, \rightarrow$-fragment
  - Friedman translation should be possible
  - Likely to require expanded standard model

FVAL and the small fragment

What about the $\forall, \rightarrow, \bot$-fragment?

- Finite models “behave classically”: $M \vDash \varphi$ decidable
- Double negation translation holds in general
- FSAT is undecidable for binary signature over $\forall, \rightarrow, \bot$-fragment

What about FVAL?

- FVAL is undecidable for binary signature over $\forall, \rightarrow, \bot$-fragment
  - using $\neg F'(h)$ as reduction function
- Conjecture: FVAL undecidable for binary signature over $\forall, \rightarrow$-fragment
  - Friedman translation should be possible
  - Likely to require expanded standard model
- There is no sound, complete, enumerable deduction system for finite semantics
  - FVAL is co-enumerable and undecidable

Summary

We have shown undecidability of

- PRV, VAL, kVAL for binary signature over $\forall, \rightarrow$-fragment
- SAT, kSAT, FSAT, FVAL for binary signature over $\forall, \rightarrow, \bot$-fragment

Summary

We have shown undecidability of

- PRV, VAL, kVAL for binary signature over $\forall, \rightarrow$-fragment
- SAT, kSAT, FSAT, FVAL for binary signature over $\forall, \rightarrow, \bot$-fragment
- Note: We defined undecidable $P :=$ decidable $P \rightarrow$ semidecidable $\overline{\mathsf{H}_{\mathsf{TM}}}$

Summary

We have shown undecidability[2] of

- PRV, VAL, kVAL for binary signature over $\forall, \rightarrow$-fragment
- SAT, kSAT, FSAT, FVAL for binary signature over $\forall, \rightarrow, \perp$-fragment

Coq mechanization:

- $\sim$900 LoC for PRV and corollaries
  - [Kirst and Hermes, 2021]: 4.5k LoC

[2] defined by semidecidability of $\overline{\mathsf{H}_{\mathsf{TM}}}$

Summary

We have shown undecidability[2] of

- PRV, VAL, kVAL for binary signature over $\forall, \rightarrow$-fragment
- SAT, kSAT, FSAT, FVAL for binary signature over $\forall, \rightarrow, \bot$-fragment

Coq mechanization:

- $\sim$900 LoC for PRV and corollaries
- $\sim$1200 LoC for FSAT, FVAL
  - [Kirst and Larchey-Wendling, 2020]: >5k LoC

[2] defined by semidecidability of $\overline{\mathsf{H}_{\mathsf{TM}}}$

Summary

We have shown undecidability[2] of

- PRV, VAL, kVAL for binary signature over $\forall, \rightarrow$-fragment
- SAT, kSAT, FSAT, FVAL for binary signature over $\forall, \rightarrow, \bot$-fragment

Coq mechanization:

- $\sim$900 LoC for PRV and corollaries
- $\sim$1200 LoC for FSAT, FVAL
- $\sim$200 LoC for UDPC
  - Requires undecidability of H10 [Larchey-Wendling and Forster, 2019]: 8k LoC

[2] defined by semidecidability of $\overline{\mathsf{H}_{\mathsf{TM}}}$

Summary

We have shown undecidability[2] of

- PRV, VAL, kVAL for binary signature over $\forall, \rightarrow$-fragment
- SAT, kSAT, FSAT, FVAL for binary signature over $\forall, \rightarrow, \bot$-fragment

Coq mechanization:

- $\sim$900 LoC for PRV and corollaries
- $\sim$1200 LoC for FSAT, FVAL
- $\sim$200 LoC for UDPC

Pain points:

- Working with de Brujin indices and double negation is unintuitive

[2] defined by semidecidability of $\overline{\mathrm{H}_{\mathrm{TM}}}$

Summary

We have shown undecidability[2] of

- PRV, VAL, kVAL for binary signature over $\forall, \rightarrow$-fragment
- SAT, kSAT, FSAT, FVAL for binary signature over $\forall, \rightarrow, \bot$-fragment

Coq mechanization:

- $\sim$900 LoC for PRV and corollaries
- $\sim$1200 LoC for FSAT, FVAL
- $\sim$200 LoC for UDPC

Pain points:

- Working with de Brujin indices and double negation is unintuitive
- Constructing abstract deduction system proofs is painful
  - Existing FOL toolbox [Hostert et al., 2021] could be expanded

[2] defined by semidecidability of $\overline{\mathsf{H}_{\mathsf{TM}}}$

Summary

We have shown undecidability[2] of

- PRV, VAL, kVAL for binary signature over $\forall, \rightarrow$-fragment
- SAT, kSAT, FSAT, FVAL for binary signature over $\forall, \rightarrow, \perp$-fragment

Coq mechanization:

- ~900 LoC for PRV and corollaries
- ~1200 LoC for FSAT, FVAL
- ~200 LoC for UDPC

Pain points:

- Working with de Brujin indices and double negation is unintuitive
- Constructing abstract deduction system proofs is painful

Future work:

- FVAL for $\forall, \rightarrow$-fragment

[2] defined by semidecidability of $\overline{H_{TM}}$

Summary

We have shown undecidability[2] of

- PRV, VAL, kVAL for binary signature over $\forall, \rightarrow$-fragment
- SAT, kSAT, FSAT, FVAL for binary signature over $\forall, \rightarrow, \bot$-fragment

Coq mechanization:

- $\sim$900 LoC for PRV and corollaries
- $\sim$1200 LoC for FSAT, FVAL
- $\sim$200 LoC for UDPC

Pain points:

- Working with de Brujin indices and double negation is unintuitive
- Constructing abstract deduction system proofs is painful

Future work:

- FVAL for $\forall, \rightarrow$-fragment
- PRV_c with MP

[2] defined by semidecidability of $\overline{\mathsf{H}_{\mathsf{TM}}}$

References

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

[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.

[Forster et al., 2020] Forster, Y., Kirst, D., and Wehr, D. (2020). Completeness Theorems for First-Order Logic Analysed in Constructive Type Theory. In *Symposium on Logical Foundations Of Computer Science (LFCS 2020), January 4-7, 2020, Deerfield Beach, Florida, U.S.A.*

[Friedman, 1978] Friedman, H. (1978). Classically and intuitionistically provably recursive functions. In Müller, G. H. and Scott, D. S., editors, *Higher Set Theory*, pages 21–27, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Gentzen, 1936] Gentzen, G. (1936). Die Widerspruchsfreiheit der reinen Zahlentheorie. *Mathematische Annalen*, 112:493–565.

[Gödel, 1933] Gödel, K. (1928-1933). Zur intuitionistischen Arithmetik und Zahlentheorie – Ergebnisse eines Mathematischen Kolloquiums. pages 493–565.

[Gödel, 1930] Gödel, K. (1930). Die Vollständigkeit der Axiome des logischen Funktionenkalküls. In Berka, K. and Kreiser, L., editors, *Logik-Texte: Kommentierte Auswahl zur Geschichte der Modernen Logik (vierte Auflage)*, pages 305–315. Akademie-Verlag, Berlin.

[Hostert et al., 2021] Hostert, J., Koch, M., and Kirst, D. (2021). A Toolbox for Mechanised First-Order Logic. In *The Coq Workshop 2021*.

[Kalmár, 1937] Kalmár, L. (1937). Zurückführung des Entscheidungsproblems auf den Fall von Formeln mit einer einzigen, binären, Funktionsvariablen. *Compositio Mathematica*, 4:137–144.

[Kirst and Hermes, 2021] Kirst, D. and Hermes, M. (2021). Synthetic Undecidability and Incompleteness of First-Order Axiom Systems in Coq. In *Interactive Theorem Proving - 12th International Conference, ITP 2021, Rome, Italy*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[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.

[Libkin, 2004] Libkin, L. (2004). *Elements of Finite Model Theory*. Springer.

[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.

[Trakhtenbrot, 1950] Trakhtenbrot, B. (1950). The Impossibility of an Algorithm for the Decidability Problem on Finite Classes. In *Proceedings of the USSR Academy of Sciences*.

[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.