Undecidability of Finitie FOL Satisfiability over Small Signatures

Bachelor Proposal Talk

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

Johannes Hostert

July 27, 2021, Saarland University

Recap

Discussed in last talk:

- Undecidability of FOL problems
  - Validity
  - intuitionistic Provability
  - Satisfiability, Kripe Validity, Kripke Satisfiability

Recap

Discussed in last talk:

- Undecidability of FOL problems
  - Validity
  - intuitionistic Provability
  - Satisfiability, Kripe Validity, Kripke Satisfiability
- Minimal version of these problems
  - Only a single binary predicate
  - $\forall \rightarrow$-fragment
  - Without $\perp$ (excluding Satisfiability)

Recap

Discussed in last talk:

- Undecidability of FOL problems
  - Validity
  - intuitionistic Provability
  - Satisfiability, Kripe Validity, Kripke Satisfiability
- Minimal version of these problems
  - Only a single binary predicate
  - $\forall \rightarrow$-fragment
  - Without $\bot$ (excluding Satisfiability)
- Source problem *UDPC*:
  - List of constraints of shape $(x, y)\#(a, b) \Leftrightarrow a = x + y + 1 \land 2 \cdot b = y^2 + y$
  - In Coq: $\mathscr{L}(\mathcal{V}^2 \times \mathcal{V}^2)$
  - Structurality allows elegantly simple axiomatizations

Recap

Discussed in last talk:

- Undecidability of FOL problems
  - Validity
  - intuitionistic Provability
  - Satisfiability, Kripe Validity, Kripke Satisfiability
- Minimal version of these problems
  - Only a single binary predicate
  - $\forall \rightarrow$-fragment
  - Without $\perp$ (excluding Satisfiability)
- Source problem *UDPC*:
  - List of constraints of shape $(x, y)\#(a, b) \Leftrightarrow a = x + y + 1 \wedge 2 \cdot b = y^2 + y$
  - In Coq: $\mathscr{L}(\mathcal{V}^2 \times \mathcal{V}^2)$
  - Structurality allows elegantly simple axiomatizations
- All mechanized in Coq

Finite Model Theory

New restriction: Models of a theory must be finite.

Finite Model Theory

New restriction: Models of a theory must be finite.

- Classical conception: Model is given as a table
- Problem *FSAT* $\varphi :\Leftrightarrow \exists M, M$ finite $\wedge\, M \models \varphi$

Finite Model Theory

New restriction: Models of a theory must be finite.

- Classical conception: Model is given as a table
- Problem *FSAT* $\varphi :\Leftrightarrow \exists M, M$ finite $\wedge\, M \vDash \varphi$

- *FSAT* is undecidable [Trakhtenbrot, 1950]:
  - Encode Turing Machine execution as a finite model (see [Libkin, 2004])
  - For minimal syntax, perform compression until a single binary predicate remains

Finite Model Theory

New restriction: Models of a theory must be finite.

- Classical conception: Model is given as a table
- Problem *FSAT* $\varphi :\Leftrightarrow \exists M, M$ finite $\wedge\, M \vDash \varphi$

- *FSAT* is undecidable [Trakhtenbrot, 1950]:
  - Encode Turing Machine execution as a finite model (see [Libkin, 2004])
  - For minimal syntax, perform compression until a single binary predicate remains

- *FSAT* is enumerable
  - finite Validity is co-enumerable and undecidable
  - finite FOL has no sound, complete and effective axiom system

Finite Model Theory

New restriction: Models of a theory must be finite.

- Classical conception: Model is given as a table
- Problem *FSAT* $\varphi :\Leftrightarrow \exists M, M$ finite $\wedge\, M \vDash \varphi$
- *FSAT* is undecidable [Trakhtenbrot, 1950]:
  - Encode Turing Machine execution as a finite model (see [Libkin, 2004])
  - For minimal syntax, perform compression until a single binary predicate remains
- *FSAT* is enumerable
  - finite Validity is co-enumerable and undecidable
  - finite FOL has no sound, complete and effective axiom system
- *FSAT* reduces to many problems in e.g. Program Verification [Calcagno et al., 2001]

Finite Model Theory in Coq

In Coq:

- Models must be listable
- Atomic predicates must be decidable

$\Rightarrow$ For fixed M, $M \vDash \varphi$ is decidable.

Finite Model Theory in Coq

In Coq:

- Models must be listable
- Atomic predicates must be decidable

$\Rightarrow$ For fixed M, $M \vDash \varphi$ is decidable.

- Previous results are mechanized in [Kirst and Larchey-Wendling, 2020]:
  - Show *FSAT* undecidable by reducing from *PCP*.
  - Perform signature compression to minimal form

Finite Model Theory in Coq

In Coq:

- Models must be listable
- Atomic predicates must be decidable

$\Rightarrow$ For fixed M, $M \vDash \varphi$ is decidable.

- Previous results are mechanized in [Kirst and Larchey-Wendling, 2020]:
  - Show *FSAT* undecidable by reducing from *PCP*.
  - Perform signature compression to minimal form

- We propose instead:
  - Show *FSAT* undecidable by reducing from *UDPC*.
  - Signature is already minimal

Reductions into *FSAT*

Reduce *UDPC* to *FSAT*:

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 \times \mathcal{V}^2\right) \to \mathit{FOL}$
- Show that $\mathit{UDPC}\, h \to \mathit{FSAT}\,(F\, h)$
- Show that $\mathit{FSAT}\,(F\, h) \to \mathit{UDPC}\, h$

Reductions into *FSAT*

Reduce *UDPC* to *FSAT*:

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 \times \mathcal{V}^2\right) \to \mathit{FOL}$
- Show that $\mathit{UDPC}\, h \to \mathit{FSAT}\,(F\, h)$
- Show that $\mathit{FSAT}\,(F\, h) \to \mathit{UDPC}\, h$

Reduction idea:

- Encode solution to *UDPC* into finite model
- Reduction formula asserts a solution exists

Reductions into *FSAT*

Reduce *UDPC* to *FSAT*:

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 \times \mathcal{V}^2\right) \to \textit{FOL}$
- Show that $\textit{UDPC}\, h \to \textit{FSAT}\,(F\, h)$
- Show that $\textit{FSAT}\,(F\, h) \to \textit{UDPC}\, h$

Reduction idea:

- Encode solution to *UDPC* into finite model
- Reduction formula asserts a solution exists
- Second direction: Deconstruct solution using elimination axioms
- First direction: Construct concrete finite model

Reductions into *FSAT*

Reduce *UDPC* to *FSAT*:

- Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 \times \mathcal{V}^2\right) \to \mathit{FOL}$
- Show that $\mathit{UDPC}\, h \to \mathit{FSAT}\,(F\, h)$
- Show that $\mathit{FSAT}\,(F\, h) \to \mathit{UDPC}\, h$

Reduction function $F : \mathscr{L}\left(\mathcal{V}^2 \times \mathcal{V}^2\right) \to \mathit{FOL}$:

$$F\ h := \exists\, 0\, m, \mathit{Axioms} \wedge \coprod_{v \in \mathcal{V}(h)}, \mathit{code}\, h$$

$$\mathit{code}\, \emptyset := \top$$

$$\mathit{code}\,((a,b)\#(c,d) :: hs) := \mathit{rel}\, a\, b\, c\, d\, m \wedge \mathit{code}\, hs$$

where $\mathit{rel}\, a\, b\, c\, d\, m$ encodes that both $(a,b)\#(c,d)$ and m bounds a, b, c, d.

Coming up with axioms

Last talk:

- Axioms should “build up” solution in arbitrary model

Coming up with axioms

Last talk:

- Axioms should "build up" solution in arbitrary model

Now:

- Axioms should "deconstruct" solution in arbitrary model
- Approach: Turn old axioms into eliminators

Coming up with axioms

Last talk:

- Axioms should "build up" solution in arbitrary model

Now:

- Axioms should "deconstruct" solution in arbitrary model
- Approach: Turn old axioms into eliminators

Example:

$$\forall a, N\ a \to \exists a', N\ a' \wedge (a, 0) \# (a', 0)$$

Coming up with axioms

Last talk:

- Axioms should “build up” solution in arbitrary model

Now:

- Axioms should “deconstruct” solution in arbitrary model
- Approach: Turn old axioms into eliminators

Example:

$$\forall a, N\ a \rightarrow \exists a', N\ a' \wedge (a, 0) \# (a', 0)$$
$$\forall a', N\ a' \rightarrow \exists a, N\ a \wedge (a, 0) \# (a', 0)$$

Coming up with axioms

Last talk:

- Axioms should "build up" solution in arbitrary model

Now:

- Axioms should "deconstruct" solution in arbitrary model
- Approach: Turn old axioms into eliminators

Example:

$$\forall a, N\ a \rightarrow \exists a', N\ a' \wedge (a,0)\#(a',0)$$
$$\forall a, N\ a' \rightarrow a \not\equiv 0 \rightarrow \exists a, N\ a \wedge (a,0)\#(a',0)$$

Coming up with axioms

Last talk:

- Axioms should "build up" solution in arbitrary model

Now:

- Axioms should "deconstruct" solution in arbitrary model
- Approach: Turn old axioms into eliminators

The axioms (so far):

- Predecessor axiom
- Eliminator laws for $\#$

$\mathit{FSAT}\,(F\;h) \to \mathit{UDPC}\,h$

Idea: extract solution from finite model

$\mathit{FSAT}\,(F\,h) \to \mathit{UDPC}\,h$

Idea: extract solution from finite model

- Find numbers representing points from model
- Induction on points of model along $<$

$\mathit{FSAT}\,(F\,h) \to \mathit{UDPC}\,h$

Idea: extract solution from finite model

- Find numbers representing points from model
- Induction on points of model along $<$
- $<$ is well-founded in model
  - Axiom asserting it is transitive
  - Define $< := \leq \wedge \neq$, so $<$ is irreflexive by definition
  - Fact: Transitive, irreflexive relations on finite types are well-founded.

$\mathit{FSAT}\,(F\,h) \to \mathit{UDPC}\,h$

Idea: extract solution from finite model

- Find numbers representing points from model
- Induction on points of model along $<$
- $<$ is well-founded in model
  - Axiom asserting it is transitive
  - Define $< := \leq \wedge \neq$, so $<$ is irreflexive by definition
  - Fact: Transitive, irreflexive relations on finite types are well-founded.
- Deconstruct $\mathit{rel}\,a\,b\,c\,d\,m$ using induction on b.

$\mathit{UDPC}\, h \to \mathit{FSAT}\,(F\, h)$

Idea: construct concrete finite model

- Prefix of standard model from last talk.

$UDPC\,h \rightarrow FSAT\,(F\,h)$

Idea: construct concrete finite model

- Prefix of standard model from last talk.
  - Model $M = \mathbb{N}_{\leq m} \cup \mathbb{N}^2_{\leq m}$: numbers up to m, and their pairs

$\mathit{UDPC}\,h \rightarrow \mathit{FSAT}\,(F\,h)$

Idea: construct concrete finite model

- Prefix of standard model from last talk.
  - Model $M = \mathbb{N}_{\leq m} \cup \mathbb{N}^2_{\leq m}$: numbers up to m, and their pairs
  - Interpretation of $\#$:
    - $(x, y)\#(a, b)$ as defined above

$UDPC\,h \to FSAT\,(F\,h)$

Idea: construct concrete finite model

- Prefix of standard model from last talk.
  - Model $M = \mathbb{N}_{\leq m} \cup \mathbb{N}^2_{\leq m}$: numbers up to m, and their pairs
  - Interpretation of $\#$:
    - $(x, y)\#(a, b)$ as defined above
    - $n_1\#(a, b) := n_1 = a$
    - $(x, y)\#n_2 := y = n_2$

$\mathit{UDPC}\, h \to \mathit{FSAT}\,(F\, h)$

Idea: construct concrete finite model

- Prefix of standard model from last talk.
  - Model $M = \mathbb{N}_{\leq m} \cup \mathbb{N}^2_{\leq m}$: numbers up to m, and their pairs
  - Interpretation of $\#$:
    - $(x, y)\#(a, b)$ as defined above
    - $n_1\#(a, b) := n_1 = a$
    - $(x, y)\#n_2 := y = n_2$
    - $n_1\#n_2 := n_1 \leq n_2$

$UDPC\,h \to FSAT\,(F\,h)$

Idea: construct concrete finite model

- Prefix of standard model from last talk.
  - Model $M = \mathbb{N}_{\leq m} \cup \mathbb{N}^2_{\leq m}$: numbers up to m, and their pairs
  - Interpretation of $\#$:
    - $(x, y)\#(a, b)$ as defined above
    - $n_1\#(a, b) := n_1 = a$
    - $(x, y)\#n_2 := y = n_2$
    - $n_1\#n_2 := n_1 \leq n_2$
- m is the highest number in the solution of h

$\mathit{UDPC}\, h \to \mathit{FSAT}\,(F\, h)$

Idea: construct concrete finite model

- Prefix of standard model from last talk.
  - Model $M = \mathbb{N}_{\leq m} \cup \mathbb{N}^2_{\leq m}$: numbers up to m, and their pairs
  - Interpretation of $\#$:
    - $(x, y)\#(a, b)$ as defined above
    - $n_1\#(a, b) := n_1 = a$
    - $(x, y)\#n_2 := y = n_2$
    - $n_1\#n_2 := n_1 \leq n_2$
- m is the highest number in the solution of h
- Show that all axioms hold

$\mathit{UDPC}\,h \to \mathit{FSAT}\,(F\,h)$

Idea: construct concrete finite model

- Prefix of standard model from last talk.
  - Model $M = \mathbb{N}_{\leq m} \cup \mathbb{N}^2_{\leq m}$: numbers up to m, and their pairs
  - Interpretation of $\#$:
    - $(x, y)\#(a, b)$ as defined above
    - $n_1\#(a, b) := n_1 = a$
    - $(x, y)\#n_2 := y = n_2$
    - $n_1\#n_2 := n_1 \leq n_2$
- m is the highest number in the solution of h
- Show that all axioms hold
- In Coq:
  - M needs to be listable
    - $\leq$ on $\mathbb{N}$ has derivation uniqueness
  - $\#$ is decidable: $\mathbb{N}$ is discrete, $\leq$ is decidable

Axioms, summarized

In total, we have 5 axioms:

Axioms, summarized

In total, we have 5 axioms:

- $\forall xyz, x < y \to y < z \to x < z$

Axioms, summarized

In total, we have 5 axioms:

- $\forall xyz, x < y \to y < z \to x < z$

- $\forall a, N\, a \to a \not\equiv 0 \to \exists a', (a', 0) \# (a, 0)$

Axioms, summarized

In total, we have 5 axioms:

- $\forall xyz, x < y \to y < z \to x < z$
- $\forall a, N\, a \to a \not\equiv 0 \to \exists a', (a', 0)\#(a, 0)$
- $\forall ab, (a, 0)\#(b, 0) \to a < b \land \forall k, k < b \to k \leq a$

Axioms, summarized

In total, we have 5 axioms:

- $\forall xyz, x < y \to y < z \to x < z$
- $\forall a, N\, a \to a \not\equiv 0 \to \exists a', (a', 0)\#(a, 0)$
- $\forall ab, (a, 0)\#(b, 0) \to a < b \wedge \forall k, k < b \to k \leq a$
- $\forall abcd, (a, b)\#(c, d) \to b \not\equiv 0 \to$ $\exists b'c'd', (b', 0)\#(b, 0) \wedge (c', 0)\#(c, 0) \wedge (a, b')\#(c', d') \wedge (d', b')\#(d, d') \wedge d' < d$
- $\forall acd, (a, 0)\#(c, d) \to d \equiv 0$
    - Elimination principles for $\#$
    - Derived from old axioms for $\#$
    - Surprisingly elegant, given that they characterize $\#$ rather completely.

More compression

- *FSAT* shown undecidable for minimal signature
- What about $\forall \rightarrow$-fragment?

More compression

- *FSAT* shown undecidable for minimal signature
- What about $\forall \rightarrow$-fragment?
  - Satisfiability for fixed model is decidable
  - Trivial reduction into small fragment by double-negation translation

More compression

- *FSAT* shown undecidable for minimal signature
- What about $\forall \rightarrow$-fragment?
  - Satisfiability for fixed model is decidable
  - Trivial reduction into small fragment by double-negation translation
- What about Friedman translation / $\bot$ eliminiation?
  - Impossible for *FSAT*

More compression

- *FSAT* shown undecidable for minimal signature
- What about $\forall \rightarrow$-fragment?
  - Satisfiability for fixed model is decidable
  - Trivial reduction into small fragment by double-negation translation
- What about Friedman translation / $\perp$ eliminiation?
  - Impossible for *FSAT*
  - If formula is positive, it is satisfied by trivial model

Summary

We have done:

- Mechanized above reductions in Coq

Summary

We have done:

- Mechanized above reductions in Coq
- in total: < 1000 LoC
- [Kirst and Larchey-Wendling, 2020]: 10k LoC

Summary

We have done:

- Mechanized above reductions in Coq
- in total: < 1000 LoC
- [Kirst and Larchey-Wendling, 2020]: 10k LoC

My contributions:

- Adapt $\#$ for *FSAT*
- Adapt old and find new axioms
- Formalization in Coq

Summary

We have done:

- Mechanized above reductions in Coq
- in total: < 1000 LoC
- [Kirst and Larchey-Wendling, 2020]: 10k LoC

My contributions:

- Adapt $\#$ for *FSAT*
- Adapt old and find new axioms
- Formalization in Coq

Not my contributions:

- Original axioms for, and definitions of $\#$

Bachelor project

Accomplished goals:

- Mechanize minimal reductions for validity, provability: ✓
- Mechanize minimal reductions for *FSAT*: ✓

Bachelor project

Accomplished goals:

- Mechanize minimal reductions for validity, provability: ✓
- Mechanize minimal reductions for *FSAT*: ✓

Remaining goals:

- Clean up Coq formalization
- Distill into dependency-less formalization
- Upstream into Coq Library of Undecidability Proofs [Forster et al., 2020]
  - We may want to change the definition of undecidability

Bachelor project

Accomplished goals:

- Mechanize minimal reductions for validity, provability: ✓
- Mechanize minimal reductions for *FSAT*: ✓

Remaining goals:

- Clean up Coq formalization
- Distill into dependency-less formalization
- Upstream into Coq Library of Undecidability Proofs [Forster et al., 2020]
  - We may want to change the definition of undecidability

Optional goals:

- Finite Validity reduction with Friedman translation
- Analyze reducing quantifier prefix
- What about classical proof systems

References

[Calcagno et al., 2001] Calcagno, C., Yang, H., and O'Hearn, P. W. (2001). Computability and complexity results for a spatial assertion language for data structures. In Hariharan, R., Vinay, V., and Mukund, M., editors, *FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science*, pages 108–119, Berlin, Heidelberg. Springer Berlin Heidelberg.

[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.

[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 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.

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

[Trakhtenbrot, 1950] Trakhtenbrot, B. (1950). The impossibility of an algorithm for the decidability problem on finite classes. Proceedings of the USSR Academy of Sciences.

[Trakhtenbrot, 1950]

- Very ancient notation
- Given a general-recursive function f, construct formula $\mathfrak{A}$ that is finitely satisfied only if f has a root
- Construction by induction on syntax of f
- Paper leaves actual construction to the reader
- Reduction is an interesting approach which might be elegantly mechanizable
- Paper is not concerned with minimal representation
  - [Kalmár, 1937] already published a reduction from FOL to FOL with minimal signature
  - [Kalmár, 1937] claims the reduction should work for finite models without presenting proof
  - The fact that one can reduce to a binary signature was folklore knowledge in 1950

[Kirst and Larchey-Wendling, 2020]

Part on Trakhenbrot:

- Show *FSAT* undecidable by reducing from *PCP*
- Signature compression chain:
  - Arbitrary FOL with equality to arbitrary FOL without equality
    - Take quotient over first-order indistinguishability
  - Arbitrary FOL to single predicate FOL
    - Actually three different reductions
    - Compress functions to predicates
    - Compress predicates to one predicate + unary functions
    - Compress functions to free variables
  - single predicate to binary predicate
    - Construction using $\in$ and HF-sets

Other results:

- Monadic signature is shown decidable
  - Function and relation symbols have arity ≤ 1, or
  - Relation symbols have arity 0

[Libkin, 2004]

- Textbook on Finite Model Theory
- Interesting section for us is 9.1
- Reduction from Turing Machine Halting Problem to *FSAT*
- Making this use minimal signature is (explicitly) left to the reader

The full reduction

1. Syntactic sugar:
   - $N\,k := k\#k$
   - $P'\,k := k\#k \to \bot$
   - $P\,p\,l\,r := P'\,p \wedge N\,l \wedge N\,r \wedge l\#p \wedge p\#r$
   - $(a,b)\#(c,d) := \exists p\,q, P\,p\,a\,b \wedge P\,q\,c\,d \wedge p\#q$
   - $x \equiv y := \forall k, k\#x \leftrightarrow k\#y \wedge x\#k \leftrightarrow y\#k$
   - $x \leq y := N\,x \wedge N\,y \wedge x\#y$
   - $x < y := x \leq y \wedge x \not\equiv y$
   - $rel\,a\,b\,c\,d\,m := (a,b)\#(c,d) \wedge a \leq m \wedge b \leq m \wedge c \leq m \wedge d \leq m$
2. Axioms:
   - $\forall xyz, x < y \to y < z \to x < z$
   - $\forall a, N\,a \to a \not\equiv 0 \to \exists a', (a',0)\#(a,0)$
   - $\forall ab, (a,0)\#(b,0) \to a < b \wedge \forall k, k < b \to k \leq a$
   - $\forall abcd, (a,b)\#(c,d) \to b \not\equiv 0 \to$
     $\exists b'c'd', (b',0)\#(b,0) \wedge (c',0)\#(c,0) \wedge (a,b')\#(c',d') \wedge (d',b')\#(d,d') \wedge d' < d$
   - $\forall acd, (a,0)\#(c,d) \to d \equiv 0$