Trakhtenbrot’s Theorem in Coq
A Constructive Approach to Finite Model Theory

Dominik Kirst, Dominique Larchey-Wendling

IJCAR’20, Paris, France
July 3, 2020
Fragments of the Entscheidungsproblem

Given a first-order formula $\varphi$, is there a model $M$ with $M \models \varphi$?

Church and Turing: general problem is undecidable\(^1\)

Restrictions on $\varphi$:
- Quantifier prefix (complexity of alternations)
- Signature (monadic vs. at least binary symbols)

Restrictions on $M$:
- Finite satisfiability (FSAT) also undecidable [Trakhtenbrot, 1950]
- Dual to general problem: finite validity not enumerable

\(^1\)Coq mechanisation presented at last year’s CPP [Forster et al., 2019]
Goal: Mechanise Trakhtenbrot’s Theorem in Coq

Textbook proofs by reduction from the halting problem:\(^2\)

- Encode Turing machine \( M \) as formula \( \varphi_M \) over custom signature
- Verify that the models of \( \varphi_M \) correspond to the runs of \( M \)
- Conclude that \( M \) halts if and only if \( \varphi_M \) has a finite model

Challenges for mechanisation:

- Reducing to binary signature usually left as exercise
- Requires a mechanised model of Turing machines
- Reduction function \( M \mapsto \varphi_M \) needs to be shown computable!

\(^2\)cf. [Libkin, 2010, Börger et al., 1997]
Simplification: Synthetic Computability

Every function definable in constructive type theory is computable

\[\Downarrow\]

Computability theory without a concrete model of computation!

A predicate \( p : X \to \mathbb{P} \)...

- is decidable: \( \exists f : X \to \mathbb{B}. \forall x. f x = \text{tt} \iff p x \)
- is enumerable: \( \exists g : \mathbb{N} \to X. \forall x. \exists n. g n = x \)
- reduces to \( q : Y \to \mathbb{P} \): \( \exists h : X \to Y. \forall x. p x \iff q (h x) \)

\[\Rightarrow\] No need to encode \( f, g, \) and \( h \) as Turing machines!
Where to start?

But don’t we need Turing machines to base the reduction to FSAT?
We can pick any problem from our Coq library of undecidability proofs:

Post correspondence problem (PCP):
- Domino-like matching problem
- Only involves boolean strings
- Easy to express in Coq’s type theory
- Easy to encode in first-order logic
- Reduction PCP \( \preceq \) FSAT easy to verify
Challenges: Constructive Finite Model Theory

Model Theory $\subseteq$ Set Theory $\cup$ Classical Logic

Working in constructive type theory poses a few questions:

- What is a good rendering of models?
- What is a good rendering of finiteness?
- Do the tools from model theory transport to our setting?
- Which other tools do we need to invent?
First-Order Satisfiability

Given a signature $\Sigma = (F_\Sigma; P_\Sigma)$, we represent terms and formulas by:

$$ t : \text{Term}_\Sigma ::= x \mid f \bar{t} \quad (x : \mathbb{N}, \ f : F_\Sigma, \ \bar{t} : \text{Term}_{\Sigma}^{\mid f \mid}) $$

$$ \varphi, \psi : \text{Form}_\Sigma ::= \bot \mid P \bar{t} \mid \varphi \Box \psi \mid \nabla \varphi \quad (P : P_\Sigma, \ \bar{t} : \text{Term}_{\Sigma}^{\mid P \mid}) $$

A (Tarski) model $\mathcal{M}$ over a domain $D$ is a pair of interpretation functions:

$$ -\mathcal{M} : \forall f : F_\Sigma. D^{\mid f \mid} \rightarrow D \quad -\mathcal{M} : \forall P : P_\Sigma. D^{\mid P \mid} \rightarrow \mathcal{P} $$

For assignments $\rho : \mathbb{N} \rightarrow D$ define evaluation $\hat{\rho} t$ and satisfaction $\mathcal{M} \vDash_\rho \varphi$:

$$ \hat{\rho} x := \rho x $$

$$ \hat{\rho} (f \bar{t}) := f^{\mathcal{M}}(\hat{\rho} \bar{t}) $$

$$ \mathcal{M} \vDash_\rho \bot := \bot $$

$$ \mathcal{M} \vDash_\rho \varphi \Box \psi := \mathcal{M} \vDash_\rho \varphi \Box \mathcal{M} \vDash_\rho \psi $$

$$ \mathcal{M} \vDash_\rho P \bar{t} := P^{\mathcal{M}}(\hat{\rho} \bar{t}) $$

$$ \mathcal{M} \vDash_\rho \nabla \varphi := \nabla a : D. \mathcal{M} \vDash_{a,\rho} \varphi $$

$$ \text{SAT}(\Sigma) \varphi := \text{there are } \mathcal{M} \text{ and } \rho \text{ such that } \mathcal{M} \vDash_\rho \varphi $$
Finiteness in Constructive Type Theory

Definition

A type $X$ is finite if there exists a list $l_X$ with $x \in l_X$ for all $x : X$.

This seems to be a good compromise:

- Easy to establish and work with
- Does not enforce discreteness (decidable equality)
- Enough to get expected properties:
  - Every strict order on a finite type is well-founded
  - Every finite decidable equivalence relation admits a quotient on $\mathbb{F}_n$

$\text{FSAT}(\Sigma) \varphi$ if additionally $D$ is finite and all $P^M$ are decidable

$\text{FSATEQ}(\Sigma; \equiv) \varphi$ if $x \equiv^M y \leftrightarrow x = y$ for all $x, y : D$ (hence discrete)
Encoding the Post Correspondence Problem

Inductive characterisation of PCP on sets $R$ of cards $(s, t)$ of strings:

\[
\begin{align*}
(s, t) & \in R \\
R & \triangleright (s, t) \\
(s, t) & \in R \\
R & \triangleright (u, v) \\
R & \triangleright (s \oplus u, t \oplus v) \\
R & \triangleright (s, s)
\end{align*}
\]

PCP $R$

We use the signature $\Sigma_{\text{BPMP}} := (\{\star^0, e^0, f^1_{tt}, f^1_{ff}\}; \{P^2, \prec^2, \equiv^2\}):$

- Chains like $f_{ff}(f_{tt}(e))$ represent strings while $\star$ signals overflow
- $P$ concerns only defined values and $\prec$ is a strict ordering:

\[
\begin{align*}
\varphi_P & := \forall xy. \, P \, x \, y \rightarrow x \not\equiv \star \land y \not\equiv \star \\
\varphi_{\prec} & := (\forall x. \, x \not< x) \land (\forall xyz. \, x < y \rightarrow y < z \rightarrow x < z)
\end{align*}
\]

- Sanity checks on $f$ regarding overflow, disjointness, and injectivity:

\[
\begin{align*}
\varphi_f & := \left(\begin{array}{c}
f_{tt} \star \equiv \star \land f_{ff} \star \equiv \star \\
\forall x. \, f_{tt} \not\equiv e \\
\forall x. \, f_{ff} \not\equiv e
\end{array}\right) \land \left(\begin{array}{c}
\forall xy. \, f_{tt} x \not\equiv \star \rightarrow f_{tt} x \equiv f_{tt} y \rightarrow x \equiv y \\
\forall xy. \, f_{ff} x \not\equiv \star \rightarrow f_{ff} x \equiv f_{ff} y \rightarrow x \equiv y \\
\forall xy. \, f_{tt} x \equiv f_{ff} y \rightarrow f_{tt} x \equiv \star \land f_{ff} y \equiv \star
\end{array}\right)
\end{align*}
\]
Trakhtenbrot’s Theorem

Given an instance $R$ of PCP, we construct a formula $\varphi_R$ by setting

$$\varphi_R := \varphi_P \land \varphi_\prec \land \varphi_f \land \varphi_\triangleright \land \exists x. P x x.$$  

Crucially, we enforce that $P$ satisfies the inversion principle of $R \triangleright (s, t)$:

$$\varphi_\triangleright := \forall xy. P x y \rightarrow \bigvee_{(s, t) \in R} \left\{ \begin{array}{c} x \equiv s \land y \equiv t \\ \exists uv. P u v \land x \equiv s \leftarrow u \land y \equiv t \leftarrow v \land u/v \prec x/y \end{array} \right.$$  

Theorem

PCP $R$ iff $\text{FSATEQ}(\Sigma_{BPCP}; \equiv)\varphi_R$, hence PCP $\preceq$ FSATEQ($\Sigma_{BPCP}; \equiv$).

Proof.

If $R$ has a solution of length $n$, then $\varphi_R$ is satisfied by the model of strings of length bounded by $n$. Conversely, if $M \models \varphi_R$ we can extract a solution of $R$ from $\varphi_\triangleright$ by well-founded induction on $\prec_M$ (which is applicable since $M$ is finite). □
Signature Transformations

Given a finite and discrete signature $\Sigma$ with arities bounded by $n$, we have:

$$\text{FSATEQ}(\Sigma; \equiv) \preceq \text{FSAT}(\Sigma) \preceq \text{FSAT}(\emptyset; P^{n+2}) \preceq \text{FSAT}(\emptyset; \in^2)$$

First reduction: axiomatise that $\equiv$ is a congruence for the symbols in $\Sigma$

Second reduction:
- Encode $k$-ary functions as $(k + 1)$-ary relations
- Align the relation arities to be constantly $n + 1$
- Merge relations into a single $(n + 2)$-ary relation indexed by constants
- Interpret constants with fresh variables

Caveat: intermediate reductions may rely on discrete models...
Discrete Models

\[
\text{FSAT}'(\Sigma) \varphi \text{ if } \text{FSAT}(\Sigma) \varphi \text{ on a discrete model}
\]

Can every finite model \( M \) be transformed to a discrete finite model \( M' \)?

Idea: first-order indistinguishability \( x \models y := \forall \varphi \rho. M \models x \cdot \rho \varphi \leftrightarrow M \models y \cdot \rho \varphi \)

Lemma

The relation \( x \models y \) is a decidable congruence for the symbols in \( \Sigma \).

Fact

\( \text{FSAT}'(\Sigma) \varphi \text{ iff } \text{FSAT}(\Sigma) \varphi \), hence in particular \( \text{FSAT}'(\Sigma) \varphi \preceq \text{FSAT}(\Sigma) \varphi \).

Proof.

If \( M \models \varphi \rho \varphi \) pick \( M' \) to be the quotient of \( M \) under \( x \models y \).  \( \square \)
Compressing Relations: $\text{FSAT}(\emptyset; P^n) \preceq \text{FSAT}(\emptyset; \in^2)$

Intuition: encode $P \, x_1 \ldots x_n$ as $(x_1, \ldots, x_n) \in p$ for a set $p$ representing $P$

So let’s play set theory! For a set $d$ representing the domain we define $\varphi'_\in$:

\[
(P \, x_1 \ldots x_n)'_\in := "(x_1, \ldots, x_n) \in p" \quad (\forall z. \varphi)'_\in := \forall z. z \in d \rightarrow (\varphi)'_\in \\
(\varphi \land \psi)'_\in := (\varphi)'_\in \land (\psi)'_\in \\
(\exists z. \varphi)'_\in := \exists z. z \in d \land (\varphi)'_\in
\]

Then $\varphi'_\in$ is $\varphi'_\in$ plus asserting $\in$ to be extensional and $d$ to be non-empty.

**Fact**

$\text{FSAT}(\emptyset; P^n) \varphi$ iff $\text{FSAT}(\emptyset; \in^2) \varphi'_\in$, hence $\text{FSAT}(\emptyset; P^n) \preceq \text{FSAT}(\emptyset; \in^2)$.

**Proof.**

The hard direction is to construct a model of $\varphi'_\in$ given a model $\mathcal{M}$ of $\varphi$.
We employ a segment of the model of hereditarily finite sets by [Smolka and Stark, 2016] large enough to accommodate $\mathcal{M}$. □
Full Signature Classification

Composing all signature transformations verified in the paper we obtain:

**Theorem**

If $\Sigma$ contains either an at least binary relation or a unary relation together with an at least binary function, then PCP reduces to FSAT($\Sigma$).

On the other hand, FSAT for monadic signatures remains decidable:

**Theorem**

If $\Sigma$ is discrete and has all arities bounded by 1 or if all relation symbols have arity 0, then FSAT($\Sigma$) is decidable.

In any case, since one can enumerate all finite models up to extensionality:

**Fact**

If $\Sigma$ is discrete and enumerable, then FSAT($\Sigma$) is enumerable.
Coq Mechanisation

- Includes all results presented in the paper (PDF is hyperlinked!)

- Roughly 10k loc with additional 3k loc of utility libraries
  - More than 4k loc for $\text{FSAT}(\emptyset; P^n) \preceq \text{FSAT}(\emptyset; \in^2)$
  - Less than 500 loc for $\text{PCP} \preceq \text{FSATEQ}(\Sigma_{\text{BPCP}}; \equiv)$

- FOL engineering similar to previous devs (cf. [Forster et al., 2020])
  - De Bruijn encoding of bound variables
  - Dependent syntax enforcing well-defined terms and formulas

- Axiom-free to ensure computability and interoperability

- Contributed to the Coq library of undecidability proofs

---

3 https://github.com/uds-psl/coq-library-undecidability
4 https://www.ps.uni-saarland.de/extras/fol-trakh/
Future Work

- Generalise some intermediate transformations from FSAT to SAT
- Explore and mechanise some direct consequences:
  - Undecidability of query containment and equivalence on data bases
  - Undecidability of separation logic
- Study further undecidability results in first-order logic
  - Deduction in axiom systems like Peano arithmetic or ZF set theory
  - Strengthen the result given in [Forster et al., 2019] to binary signatures
- Mechanise the classification of satisfiability wrt. quantifier prefixes
What to take home?

- Synthetic computability simplifies undecidability proofs
- PCP is a good starting point for reductions into FOL
- Initial reduction was easy, signature transformations were tough
- Constructive (finite) model theory has interesting structure to explore
- Undecidability library open for contributions!

Thank You!
Bibliography

*The Classical Decision Problem.*
Perspectives in Mathematical Logic. Springer-Verlag Berlin Heidelberg.

On synthetic undecidability in Coq, with an application to the Entscheidungsproblem.
In *International Conference on Certified Programs and Proofs*, pages 38–51. ACM.

Completeness Theorems for First-Order Logic Analysed in Constructive Type Theory.
In *Symposium on Logical Foundations Of Computer Science, 2020, Deerfield Beach, Florida, U.S.A.*

A Certifying Extraction with Time Bounds from Coq to Call-By-Value Lambda Calculus.

*Elements of Finite Model Theory.*

Hereditarily Finite Sets in Constructive Type Theory.

Trakhtenbrot, B. A. (1950).
The impossibility of an algorithm for the decidability problem on finite classes.
Disclaimer: Synthetic Undecidability

Coq is consistent with axioms making all problems decidable!
- FSAT not shown undecidable in the sense that there is no Coq decider
- Only reductions like $\text{PCP} \preceq \text{FSAT}$ can be verified
- Axioms like undecidability of PCP then have expected consequences

Alternative: switch to a concrete model of computation
- Pro: problems can be shown to have no deciding e.g. Turing machine
- Contra: reductions need to be encoded as e.g. Turing machines
- Tool support in principle possible (cf. [Forster and Kunze, 2019])
First-Order Indistinguishability

We define operators $F_{\mathcal{F}}, F_{\mathcal{P}} : (D \to D \to \mathcal{P}) \to (D \to D \to \mathcal{P})$ by:

$$F_{\mathcal{F}}(\mathcal{R}) x y := \forall f. f \in l_{\mathcal{F}} \to \forall (\vec{v} : D^{\left|f\right|}) (i : \mathbb{F}_{|f|}). \mathcal{R} (f.\mathcal{M} \vec{v}[x/i]) (f.\mathcal{M} \vec{v}[y/i])$$

$$F_{\mathcal{P}}(\mathcal{R}) x y := \forall P. P \in l_{\mathcal{P}} \to \forall (\vec{v} : D^{\left|P\right|}) (i : \mathbb{F}_{|P|}). P^{\mathcal{M}} \vec{v}[x/i] \leftrightarrow P^{\mathcal{M}} \vec{v}[y/i]$$

We then consider $F(\mathcal{R}) := F_{\mathcal{F}}(\mathcal{R}) \cap F_{\mathcal{P}}(\mathcal{R})$ and show:

**Theorem**

*First-order indistinguishability $\dot{=} \uparrow$ up to $l_{\mathcal{F}}/l_{\mathcal{P}}$ is extensionally equivalent to $\equiv_F$ (Kleene’s greatest fixpoint of $F$), i.e. for any $x, y : D$ we have*

$$x \dot{=} y \iff x \equiv_F y \quad \text{where} \quad x \equiv_F y := \forall n : \mathbb{N}. F^n(\lambda uv. \top) x y.$$ 

*Moreover, the relation $x \equiv_F y$ is decidable and hence so is $x \dot{=} y$.*
Hereditarily Finite Sets

Theorem

Given a decidable $n$-ary relation $R : X^n \to \mathbb{P}$ over a finite, discrete and inhabited type $X$, one can compute a finite and discrete type $Y$ equipped with a decidable relation $\in : Y \to Y \to \mathbb{P}$, two distinguished elements $d, r : Y$ and a pair of maps $i : X \to Y$ and $s : Y \to X$ s.t.

1. $\in$ is extensional;
2. extensionally equal elements of $Y$ are equal;
3. all $n$-tuples of members of $d$ exist in $Y$;
4. $\forall x : X. i x \in d$;
5. $\forall y : Y. y \in d \to \exists x. y = i x$;
6. $\forall x : X. s(i x) = x$;
7. $R \vec{v}$ iff $i(\vec{v})$ is a $n$-tuple member of $r$, for any $\vec{v} : X^n$.

Proof.

The type $Y$ is built from the type of hereditarily finite sets. The idea is first to construct $d$ as a transitive set of which the elements are in bijection $i/s$ with the type $X$, hence $d$ is the cardinal of $X$ in the set-theoretic meaning. Then the iterated powersets $\mathcal{P}(d), \mathcal{P}^2(d), \ldots, \mathcal{P}^k(d)$ are all transitive as well and contain $d$ both as a member and as a subset. Considering $\mathcal{P}^{2n}(d)$ which contains all the $n$-tuples built from the members of $d$, we define $r$ as the set of $n$-tuples collecting the encoding $i(\vec{v})$ of vectors $\vec{v} : X^n$ such that $R \vec{v}$. We show $r \in p$ for $p$ defined as $p := \mathcal{P}^{2n+1}(d)$. Then we define $Y := \{z \mid z \in p\}$ and restrict membership $\in$ to $Y$. 

D. Kirst, D. Larchey-Wendling

Trakhtenbrot’s Theorem in Coq

IJCAR’20
Decidability Results

Lemmas used for decidability of monadic FOL and enumerability of FSAT:

**Lemma**

Given a discrete signature $\Sigma$ and a discrete and finite type $D$, one can decide whether or not a formula over $\Sigma$ has a (finite) model over $D$.

**Lemma**

A formula over a signature $\Sigma$ has a finite and discrete model if and only if it has a (finite) model over $\mathbb{F}_n$ for some $n : \mathbb{N}$.