Trakhtenbrot's Theorem in Coq

A Constructive Approach to Finite Model Theory

Dominik Kirst, Dominique Larchey-Wendling

IJCAR'20, Paris, France
July 3, 2020

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Loria

cnrs

Fragments of the Entscheidungsproblem

Given a first-order formula φ, is there a model $\mathcal{M}$ with $\mathcal{M} \vDash \varphi$?

Church and Turing: general problem is undecidable[1]

Restrictions on φ:

- Quantifier prefix (complexity of alternations)
- Signature (monadic vs. at least binary symbols)

Restrictions on $\mathcal{M}$:

- 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 φ_M over custom signature
- Verify that the models of φ_M correspond to the runs of M
- Conclude that M halts if and only if φ_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 $p : X \to \mathbb{P}$...

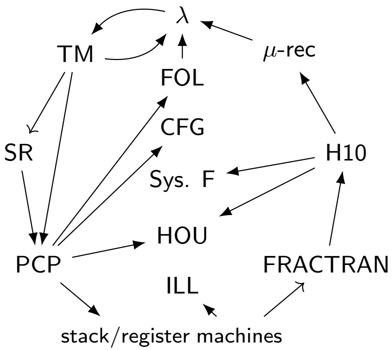
- is decidable: $\exists f : X \to \mathbb{B}. \forall x. f\, x = \mathsf{tt} \leftrightarrow p\, x$
- is enumerable: $\exists g : \mathbb{N} \to X. \forall x. \exists n. g\, n = x$
- reduces to $q : Y \to \mathbb{P}$: $\exists h : X \to Y. \forall x. p\, x \leftrightarrow q\, (h\, x)$

$\Rightarrow$ 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 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 $\Sigma = (\mathcal{F}_\Sigma; \mathcal{P}_\Sigma)$, we represent terms and formulas by:

$$t \;:\; \mathsf{Term}_\Sigma \;::=\; x \mid f\,\vec{t} \qquad\qquad (x : \mathbb{N},\ f : \mathcal{F}_\Sigma,\ \vec{t} : \mathsf{Term}_\Sigma^{|f|})$$

$$\varphi, \psi \;:\; \mathsf{Form}_\Sigma \;::=\; \dot\bot \mid P\,\vec{t} \mid \varphi \,\dot\Box\, \psi \mid \dot\nabla \varphi \qquad\qquad (P : \mathcal{P}_\Sigma,\ \vec{t} : \mathsf{Term}_\Sigma^{|P|})$$

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

$$\_^{\mathcal{M}} \;:\; \forall f : \mathcal{F}_\Sigma.\, D^{|f|} \to D \qquad\qquad \_^{\mathcal{M}} \;:\; \forall P : \mathcal{P}_\Sigma.\, D^{|P|} \to \mathbb{P}$$

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

$$\hat\rho\, x \;:=\; \rho x \qquad\qquad \hat\rho\,(f\,\vec{t}) \;:=\; f^{\mathcal{M}}\,(\hat\rho\,\vec{t})$$

$$\mathcal{M} \vDash_\rho \dot\bot \;:=\; \bot \qquad\qquad \mathcal{M} \vDash_\rho \varphi \,\dot\Box\, \psi \;:=\; \mathcal{M} \vDash_\rho \varphi \,\Box\, \mathcal{M} \vDash_\rho \psi$$

$$\mathcal{M} \vDash_\rho P\,\vec{t} \;:=\; P^{\mathcal{M}}\,(\hat\rho\,\vec{t}) \qquad\qquad \mathcal{M} \vDash_\rho \dot\nabla \varphi \;:=\; \nabla a : D.\, \mathcal{M} \vDash_{a \cdot \rho} \varphi$$

$\mathsf{SAT}(\Sigma)\,\varphi$:= there are $\mathcal{M}$ and ρ 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 $\mathbb{F}_n$

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

$\mathsf{FSATEQ}(\Sigma;\equiv)\,\varphi$ if $x \equiv^{\mathcal{M}} y \leftrightarrow x = y$ 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:

$$\frac{(s,t) \in R}{R \triangleright (s,t)} \qquad \frac{(s,t) \in R \quad R \triangleright (u,v)}{R \triangleright (s \mathbin{+\!\!+} u, t \mathbin{+\!\!+} v)} \qquad \frac{R \triangleright (s,s)}{\mathrm{PCP}\ R}$$

We use the signature $\Sigma_{\mathrm{BPCP}} := (\{\star^0, e^0, f_{\mathrm{tt}}^1, f_{\mathrm{ff}}^1\}; \{P^2, \prec^2, \equiv^2\})$:

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

$$\begin{aligned} \varphi_P &:= \dot\forall xy.\, P\,x\,y \mathrel{\dot\to} x \not\equiv \star \mathrel{\dot\wedge} y \not\equiv \star \\ \varphi_\prec &:= (\dot\forall x.\, x \not\prec x) \mathrel{\dot\wedge} (\dot\forall xyz.\, x \prec y \mathrel{\dot\to} y \prec z \mathrel{\dot\to} x \prec z) \end{aligned}$$

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

$$\varphi_f := \begin{pmatrix} f_{\mathrm{tt}}\,\star \equiv \star \mathrel{\dot\wedge} f_{\mathrm{ff}}\,\star \equiv \star \\ \dot\forall x.\, f_{\mathrm{tt}}\,x \not\equiv e \\ \dot\forall x.\, f_{\mathrm{ff}}\,x \not\equiv e \end{pmatrix} \mathrel{\dot\wedge} \begin{pmatrix} \dot\forall xy.\, f_{\mathrm{tt}}\,x \not\equiv \star \mathrel{\dot\to} f_{\mathrm{tt}}\,x \equiv f_{\mathrm{tt}}\,y \mathrel{\dot\to} x \equiv y \\ \dot\forall xy.\, f_{\mathrm{ff}}\,x \not\equiv \star \mathrel{\dot\to} f_{\mathrm{ff}}\,x \equiv f_{\mathrm{ff}}\,y \mathrel{\dot\to} x \equiv y \\ \dot\forall xy.\, f_{\mathrm{tt}}\,x \equiv f_{\mathrm{ff}}\,y \mathrel{\dot\to} f_{\mathrm{tt}}\,x \equiv \star \mathrel{\dot\wedge} f_{\mathrm{ff}}\,y \equiv \star \end{pmatrix}$$

Trakhtenbrot's Theorem

Given an instance R of PCP, we construct a formula φ_R by setting

$$\varphi_R := \varphi_P \mathbin{\dot\land} \varphi_\prec \mathbin{\dot\land} \varphi_f \mathbin{\dot\land} \varphi_\triangleright \mathbin{\dot\land} \dot\exists x.\, P\, x\, x.$$

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

$$\varphi_\triangleright := \dot\forall xy.\, P\, x\, y \mathbin{\dot\to} \dot{\bigvee_{(s,t)\in R}} \mathbin{\dot\lor} \begin{cases} x \equiv \overline{s} \mathbin{\dot\land} y \equiv \overline{t} \\ \dot\exists uv.\, P\, u\, v \mathbin{\dot\land} x \equiv \overline{s} \mathbin{+\!\!\!+\!\!\!+} u \mathbin{\dot\land} y \equiv \overline{t} \mathbin{+\!\!\!+\!\!\!+} v \mathbin{\dot\land} u/v \prec x/y \end{cases}$$

Theorem

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

Proof.

If R has a solution of length n, then φ_R is satisfied by the model of strings of length bounded by n. Conversely, if $\mathcal{M} \vDash_\rho \varphi_R$ we can extract a solution of R from $\varphi_\triangleright$ by well-founded induction on $\prec^{\mathcal{M}}$ (which is applicable since $\mathcal{M}$ is finite). $\square$

Signature Transformations

Given a finite and discrete signature Σ with arities bounded by n, we have:

$$\mathsf{FSATEQ}(\Sigma; \equiv) \preceq \mathsf{FSAT}(\Sigma) \preceq \mathsf{FSAT}(\mathbb{0}; P^{n+2}) \preceq \mathsf{FSAT}(\mathbb{0}; \in^2)$$

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

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

$\mathsf{FSAT}'(\Sigma)\,\varphi$ if $\mathsf{FSAT}(\Sigma)\,\varphi$ on a discrete model

Can every finite model $\mathcal{M}$ be transformed to a discrete finite model $\mathcal{M}'$?

Idea: first-order indistinguishability $x \doteq y \;:=\; \forall \varphi \rho.\, \mathcal{M} \vDash_{x\cdot\rho} \varphi \leftrightarrow \mathcal{M} \vDash_{y\cdot\rho} \varphi$

Lemma

The relation $x \doteq y$ is a decidable congruence for the symbols in Σ.

Fact

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

Proof.

If $\mathcal{M} \vDash_\rho \varphi$ pick $\mathcal{M}'$ to be the quotient of $\mathcal{M}$ under $x \doteq y$. $\square$

Compressing Relations: $\mathsf{FSAT}(\mathbb{0}; P^n) \preceq \mathsf{FSAT}(\mathbb{0}; \in^2)$

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 $\varphi'_\in$:

$$\begin{aligned} (P\,x_1 \ldots x_n)'_\in &:= \text{“}(x_1, \ldots, x_n) \in p\text{”} & (\dot\forall z.\,\varphi)'_\in &:= \dot\forall z.\, z \in d \mathrel{\dot\to} (\varphi)'_\in \\ (\varphi \mathbin{\dot\square} \psi)'_\in &:= (\varphi)'_\in \mathbin{\dot\square} (\psi)'_\in & (\dot\exists z.\,\varphi)'_\in &:= \dot\exists z.\, z \in d \mathbin{\dot\wedge} (\varphi)'_\in \end{aligned}$$

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

Fact

$\mathsf{FSAT}(\mathbb{0}; P^n)\,\varphi$ *iff* $\mathsf{FSAT}(\mathbb{0}; \in^2)\,\varphi_\in$, *hence* $\mathsf{FSAT}(\mathbb{0}; P^n) \preceq \mathsf{FSAT}(\mathbb{0}; \in^2)$.

Proof.

The hard direction is to construct a model of $\varphi_\in$ given a model $\mathcal{M}$ of φ. 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 Σ contains either an at least binary relation or a unary relation together with an at least binary function, then PCP *reduces to* $\mathsf{FSAT}(\Sigma)$.

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

Theorem

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

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

Fact

If Σ is discrete and enumerable, then $\mathsf{FSAT}(\Sigma)$ *is enumerable.*

Coq Mechanisation[4]

- 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 $\mathsf{FSAT}(\mathbb{0}; P^n) \preceq \mathsf{FSAT}(\mathbb{0}; \in^2)$
  - Less than 500 loc for $\mathsf{PCP} \preceq \mathsf{FSATEQ}(\Sigma_{\mathsf{BPCP}}; \equiv)$
- 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]

[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

Börger, E., Grädel, E., and Gurevich, Y. (1997).
The Classical Decision Problem.
Perspectives in Mathematical Logic. Springer-Verlag Berlin Heidelberg.

Forster, Y., Kirst, D., and Smolka, G. (2019).
On synthetic undecidability in Coq, with an application to the Entscheidungsproblem.
In *International Conference on Certified Programs and Proofs*, pages 38–51. ACM.

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, 2020, Deerfield Beach, Florida, U.S.A.*

Forster, Y. and Kunze, F. (2019).
A Certifying Extraction with Time Bounds from Coq to Call-By-Value Lambda Calculus.
In Harrison, J., O'Leary, J., and Tolmach, A., editors, *10th International Conference on Interactive Theorem Proving*, volume 141 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:19, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Libkin, L. (2010).
Elements of Finite Model Theory.
Springer Publishing Company, Incorporated, 1st edition.

Smolka, G. and Stark, K. (2016).
Hereditarily Finite Sets in Constructive Type Theory.
In *Interactive Theorem Proving - 7th International Conference, ITP 2016, Nancy, France, August 22-27, 2016*, volume 9807 of *LNCS*, pages 374–390. Springer.

Trakhtenbrot, B. A. (1950).
The impossibility of an algorithm for the decidability problem on finite classes.
Dokl. Akad. Nok. SSSR, 70(4):569–572.

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 PCP $\preceq$ 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 $\mathrm{F}_{\mathcal{F}}, \mathrm{F}_{\mathcal{P}} : (D \to D \to \mathbb{P}) \to (D \to D \to \mathbb{P})$ by :

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

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

Theorem

First-order indistinguishability $\doteq$ up to $l_{\mathcal{F}}/l_{\mathcal{P}}$ is extensionally equivalent to $\equiv_{\mathrm{F}}$ (Kleene's greatest fixpoint of F), i.e. for any $x, y : D$ we have

$$x \doteq y \;\leftrightarrow\; x \equiv_{\mathrm{F}} y \quad \textit{where} \quad x \equiv_{\mathrm{F}} y \;:=\; \forall n : \mathbb{N}.\, \mathrm{F}^n(\lambda uv.\top)\, x\, y.$$

Moreover, the relation $x \equiv_{\mathrm{F}} y$ is decidable and hence so is $x \doteq y$.

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. □

Decidability Results

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

Lemma

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

Lemma

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