A theory which *really* doesn't have a computable model

Patrick Lutz

UC Berkeley

Joint work with James Walsh

Tennenbaum's Theorem

Gödel's completeness theorem. Every consistent theory has a model.

Natural question. Does every computably axiomatizable consistent theory have a computable model?

Answer. No!

Two methods.

Method 1: Direct construction.
E.g. a theory whose models all code paths through a computable tree with no computable paths.

Method 2: Tennenbaum's Theorem.
E.g. $\mathsf{PA} + \neg\,\mathsf{Con}(\mathsf{PA})$, RCA_0, ZF, $\ldots$

Method 1: Direct construction.

Fact. There is a computable infinite binary tree, T, with no computable infinite path.

Language.

0	Constant
S	Unary function
A	Unary relation

Idea. Truth values of $A(0), A(S(0)), A(S(S(0))), \ldots$ code a path through T

Theory. For all n,

$$\bigvee_{\sigma \in T_n} \left(\bigwedge_{\substack{i \leq n \\ \sigma(i)=1}} A(S^i(0)) \wedge \bigwedge_{\substack{i \leq n \\ \sigma(i)=0}} \neg A(S^i(0)) \right)$$

where $T_n =$ set of nodes in T of height n.

Method 2: Tennenbaum's Theorem.

Theorem (Tennenbaum). No nonstandard model of PA is computable.

Corollary. $\mathsf{PA} + \neg\,\mathsf{Con}(\mathsf{PA})$ has no computable model.

Proof. $\mathbb{N} \not\models \mathsf{PA} + \neg\,\mathsf{Con}(\mathsf{PA})$

$$\Longrightarrow \ \mathsf{PA} + \neg\,\mathsf{Con}(\mathsf{PA}) \text{ has only non-standard models}$$
$$\Longrightarrow \ \mathsf{PA} + \neg\,\mathsf{Con}(\mathsf{PA}) \text{ has only non-computable models.}$$

$\mathsf{RCA}_0, \mathsf{ZF}, \mathsf{ZFC}, \ldots$ can be proved to have no computable models by adapting the proof of Tennenbaum's Theorem

Pakhomov's Theorem

Tennenbaum: No nonstandard model of PA is computable

Pakhomov: That depends on what language you use to express PA!

Key notion. Definitional equivalence.

Informally. Theories T and T' in languages L and L' are definitionally equivalent if they are the same theory, but with different choices of what notions to take as primitive

Example. PA and ZF $-$ Infinity $+ \neg$ Infinity $+$ TransitiveClosure

Via the Ackermann interpretation (a number represents a set via its hereditary base 2 expansion)

Pakhomov's Theorem (informal version). All of the theories listed on the previous slide are definitionally equivalent to a theory with a computable model

Definition. Suppose we have theories $T \subseteq T'$ in languages $L \subseteq L'$. T' is a definitional extension of T if:

- For every constant symbol c in $L' \setminus L$, there is an L-formula $\varphi_c(x)$ such that
$$T' \vdash \forall x \, (\varphi_c(x) \longleftrightarrow x = c)$$
and similarly for function and relation symbols.
- T' is conservative over T.

Every symbol in L' has an "L-definition"

Example. Let $T = \mathrm{Th}(\mathbb{Z}, +, x)$ and $T' = \mathrm{Th}(\mathbb{Z}, +, x, \leq)$

$\varphi_{\leq}(x, y) = \exists z_1, z_2, z_3, z_4 \ \left[x + (z_1^2 + z_2^2 + z_3^2 + z_4^2) = y\right]$

Definition. Theories T and T' in (disjoint) languages L and L' are definitionally equivalent if they have a common definitional extension

Example. $T = \mathrm{Th}(\mathbb{Z}, +)$ and $T' = \mathrm{Th}(\mathbb{Z}, -)$

Definition. Theories T and T' in (disjoint) languages L and L' are definitionally equivalent if they have a common definitional extension

Comment. Definitional equivalence = bi-interpretability where the home sort stays the same

T, T' definitionally equivalent $\implies$ Every L symbol has an L'-definition and vice-versa

Important point. Suppose T and T' are definitionally equivalent

From a model $M \vDash T$ we get a model $N \vDash T'$ by interpreting symbols of L' according to their L-definitions

Similarly, from a model $N \vDash T'$ we get a model $M \vDash T$

Theorem (Pakhomov). There is a theory definitionally equivalent to ZF which has a computable model

Question (Pakhomov). Is every computably axiomatizable, consistent theory definitionally equivalent to a theory with a computable model?

(By Pakhomov's Theorem, Tennenbaum's Theorem no longer answers this question)

Answer. No.

Theorem (L.-Walsh). There is a computably axiomatizable, consistent theory T such that no theory which is definitionally equivalent to T has a computable model

The proof uses non-trivial model theory

How does model theory help?

Theorem (L.-Walsh). There is a computably axiomatizable, consistent theory T such that no theory which is definitionally equivalent to T has a computable model

Idea. Suppose we have

- T theory (in a finite language) with no computable models
- T' definitionally equivalent to T
- N model of T'

Goal: Show N is not computable

Inside N, we can define a model $M \vDash T$

We would like to show that if N is computable, so is M

Assume: N has quantifier elimination

N has QE $\implies$ M is definable by quantifier-free formulas
$\implies$ M is computable from N
$\implies$ N is not computable

Problem. QE not preserved by definitional equivalence

Summary. Suppose T and T' are definitionally equivalent and no model of T is computable and let $N \vDash T'$

N has QE $\implies$ N computes a model of T $\implies$ N not computable

Problem. QE not preserved by definitional equivalence

Example. $T = \mathrm{Th}(\mathbb{Z}, \leq, S)$ and $T' = \mathrm{Th}(\mathbb{Z}, \leq)$.

The successor function is definable in T', but not without quantifiers

Solution. Identify some model-theoretic tameness property which:

1. implies QE
2. is preserved by definitional equivalence

(Almost) Perfect tool: Laskowski's theory of mutual algebraicity
(Thanks, James Hanson)

Actually, only gives a weak form of QE

Definition. $\varphi(\overline{x})$ is mutually algebraic over M if there is some $k \in \mathbb{N}$ such that for all nontrivial partitions $\overline{x} = \overline{x}_0 \cup \overline{x}_1$ and all $\overline{a}$ in M,

$$|\{\overline{b} \mid M \models \varphi(\overline{a}, \overline{b})\}| \leq k.$$

Example. $M = (\mathbb{Z}, +)$
$x = y + 3$ mutually algebraic
$x = y + z + 3$ not mutually algebraic

Definition. M is mutually algebraic if every formula is equivalent to a Boolean combination of formulas mutually algebraic over M

Example. $(\mathbb{Z}, S)$ mutually algebraic
$(\mathbb{Q}, \leq)$ not mutually algebraic

Two key facts

Fact. Mutual algebraicity is preserved by definitional equivalence

Theorem (Laskowski). If M is mutually algebraic then every formula is equivalent to a Boolean combination of mutually algebraic formulas with only existential quantifiers

T theory with no computable models
T' definitionally equivalent to T
N model of T'

Old strategy. Hope that N has QE

$$N \text{ has QE} \implies N \text{ computes a model of } T$$
$$\implies N \text{ not computable}$$

New strategy. Pick T such that all its models are mutually algebraic

$$N \vDash T' \implies N \text{ is mutually algebraic}$$
$$\implies N \text{ has weak QE}$$
$$\implies N \text{ can computably approximate a model of } T$$

Idea. Pick T such that all its models are mutually algebraic and none of its models can be computably approximated

The counterexample

Idea. Pick a theory such that all its models are mutually algebraic and none of its models can be computably approximated

Definition. Given $f : \mathbb{N} \to \mathbb{N}$, $x \in 2^\omega$ is f-guessable if there is an algorithm which, for every n, enumerates a list of at most $O(f(n))$ strings, one of which is $x \upharpoonright n$

Proposition. There is a computable infinite binary tree, T, with no n^2-guessable paths

E.g. a tree whose paths are all Martin-Löf random

Language.

0	Constant
S	Unary function
A	Unary relation

Theory. Two parts:

1. $\mathrm{Th}(\mathbb{Z}, 0, S)$
2. $A(0), A(S(0)), A(S^2(0)), \ldots$ codes a path through T

1 ensures mutual algebraicity, 2 ensures no models computably approximable