und.1 Trakhtenbrot's Theorem

tur:und:tra:sec

In ?? we defined sentences $\tau(M, w)$ and $\alpha(M, w)$ for a Turing machine M and input string w. Then we showed in ?? and ?? that $\tau(M, w) \to \alpha(M, w)$ is valid iff M, started on input w, eventually halts. Since the Halting Problem is undecidable, this implies that validity and satisfiability of sentences of first-order logic is undecidable (????). explanation

But validity and satisfiability of sentences is defined for arbitrary structures, finite or infinite. You might suspect that it is easier to decide if a sentence is satisfiable in a finite structure (or valid in all finite structures). We can adapt the proof of the unsolvability of the decision problem so that it shows this is not the case.

First, if you go back to the proof of ??, you'll see that what we did there is produce a model $\mathfrak{M}$ of $\tau(M, w)$ which describes exactly what machine M does when started on input w. The domain of that model was $\mathbb{N}$, i.e., infinite. But if M actually halts on input w, we can build a finite model $\mathfrak{M}'$ in the same way. Suppose M started on input w halts after k steps. Take as domain $|\mathfrak{M}'|$ the set $\{0, \dots, n\}$, where n is the larger of k and the length of w, and let

$$\prime^{\mathfrak{M}'}(x) = \begin{cases} x+1 & \text{if } x < n \\ n & \text{otherwise,} \end{cases}$$

and $\langle x, y\rangle \in <^{\mathfrak{M}'}$ iff $x < y$ or $x = y = n$. Otherwise $\mathfrak{M}'$ is defined just like $\mathfrak{M}$. By the definition of $\mathfrak{M}'$, just like in the proof of ??, $\mathfrak{M}' \vDash \tau(M, w)$. And since we assumed that M halts on input w, $\mathfrak{M}' \vDash \alpha(M, w)$. So, $\mathfrak{M}'$ is a finite model of $\tau(M, w) \land \alpha(M, w)$ (note that we've replaced $\to$ with $\land$).

We are halfway to a proof: we've shown that if M halts on input w, then $\tau(M, w) \land \alpha(M, w)$ has a finite model. Unfortunately, the converse of this does not hold, i.e., there are Turing machines that don't halt on some input w, but $\tau(M, w) \land \alpha(M, w)$ still has a finite model. For instance, consider the machine M with the single state q_0 and instruction $\delta(q_0, 0) = \langle q_0, 0, N\rangle$. Started on empty input $w = \Lambda$, this machine never halts: it is in an infinite loop, but does not change the tape or move the head. All configurations are the same (same state, same head position, same tape contents). We can define a finite structure $\mathfrak{M}''$ that satisfies $\tau(M, \Lambda) \land \alpha(M, \Lambda)$ (exercise). We can, however, change $\tau(M, w)$ in a suitable way so that such structures are ruled out.

Problem und.1. Let M be a Turing machine with the single state q_0 and single instruction $\delta(q_0, 0) = \langle q, 0, N\rangle$. Let $|\mathfrak{M}''| = \{0, 1, 2\}$, $\prime^{\mathfrak{M}''}(0) = \prime^{\mathfrak{M}'}(1) = 1$ and $\prime^{\mathfrak{M}''}(2) = 2$, and $<^{\mathfrak{M}''} = \{\langle 0, 1\rangle, \langle 1, 1\rangle, \langle 2, 2\rangle\}$. Define $Q_{q_0}^{\mathfrak{M}''}$, $S_0^{\mathfrak{M}''}$, and $S_{\triangleright}^{\mathfrak{M}''}$ so that $\tau(M, \Lambda)$ and $\alpha(M, \Lambda)$ become true and explain why they are. Hint: Observe that $\delta(q_0, \triangleright)$ is undefined. Ensure that

$$Q_{q_0}(\overline{1}, \overline{n}) \land S_{\triangleright}(\overline{0}, \overline{n}) \land \forall x\, (\overline{0} < x \to S_0(x, \overline{n})) \quad \text{for all } n \in \mathbb{N}$$
$$\exists y\, (Q_{q_0}(\overline{0}, y) \land S_{\triangleright}(\overline{0}, y))$$

are both true in $\mathfrak{M}''$.

Consider the sentences describing the operation of the Turing machine M on input $w = \sigma_{i_1} \dots \sigma_{i_k}$:

1. Axioms describing numbers and $<$ (just like in the definition of $\tau(M, w)$ in **??**).

2. Axioms describing the input configuration: just like in the definition of $\tau(M, w)$.

3. Axioms describing the transition from one configuration to the next:

   For the following, let $\varphi(x, y)$ be as before, and let

$$\psi(y) \equiv \forall x\,(x < y \rightarrow x \neq y).$$

   a) For every instruction $\delta(q_i, \sigma) = \langle q_j, \sigma', R\rangle$, the sentence: tur:und:tra:rep-right

$$\forall x\,\forall y\,((Q_{q_i}(x, y) \land S_\sigma(x, y)) \rightarrow \\ (Q_{q_j}(x', y') \land S_{\sigma'}(x, y') \land \varphi(x, y) \land \psi(y')))$$

   b) For every instruction $\delta(q_i, \sigma) = \langle q_j, \sigma', L\rangle$, the sentence tur:und:tra:rep-left

$$\forall x\,\forall y\,((Q_{q_i}(x', y) \land S_\sigma(x', y)) \rightarrow \\ (Q_{q_j}(x, y') \land S_{\sigma'}(x', y') \land \varphi(x, y))) \land \\ \forall y\,((Q_{q_i}(\overline{0}, y) \land S_\sigma(\overline{0}, y)) \rightarrow \\ (Q_{q_j}(\overline{0}, y') \land S_{\sigma'}(\overline{0}, y') \land \varphi(\overline{0}, y) \land \psi(y')))$$

   c) For every instruction $\delta(q_i, \sigma) = \langle q_j, \sigma', N\rangle$, the sentence: tur:und:tra:rep-stay

$$\forall x\,\forall y\,((Q_{q_i}(x, y) \land S_\sigma(x, y)) \rightarrow \\ (Q_{q_j}(x, y') \land S_{\sigma'}(x, y') \land \varphi(x, y) \land \psi(y')))$$

   As you can see, the sentences describing the transitions of M are the same as the corresponding sentence in $\tau(M, w)$, except we add $\psi(y')$ at the end. $\psi(y')$ ensures that the number y' of the "next" configuration is different from all previous numbers o, o', ...

Let $\tau'(M, w)$ be the conjunction of all the above sentences for Turing machine M and input w.

Lemma und.1. *If M started on input w halts, then $\tau'(M, w) \land \alpha(M, w)$ has a finite model.* tur:und:tra:lem:halts-sat

Proof. Let $\mathfrak{M}'$ be as in the proof of **??**, except

$$|\mathfrak{M}'| = \{0, \dots, n\},$$

$$\prime^{\mathfrak{M}'}(x) = \begin{cases} x + 1 & \text{if } x < n \\ n & \text{otherwise,} \end{cases}$$

$$\langle x, y\rangle \in <^{\mathfrak{M}'} \text{iff } x < y \text{ or } x = y = n,$$

where $n = \max(k, \mathrm{len}(w))$ and k is the least number such that M started on input w has halted after k steps. We leave the verification that $\mathfrak{M}' \vDash \tau'(M, w) \wedge E(M, w)$ as an exercise. □

Problem und.2. Complete the proof of Lemma und.1 by proving that $\mathfrak{M}' \vDash \tau(M, w) \wedge E(M, w)$.

tur:und:tra: lem:sat-halts

Lemma und.2. *If $\tau'(M, w) \wedge \alpha(M, w)$ has a finite model, then M started on input w halts.*

Proof. We show the contrapositive. Suppose that M started on w does not halt. If $\tau'(M, w) \wedge \alpha(M, w)$ has no model at all, we are done. So assume $\mathfrak{M}$ is a model of $\tau(M, w) \wedge \alpha(M, w)$. We have to show that it cannot be finite.

We can prove, just like in **??**, that if M, started on input w, has not halted after n steps, then $\tau'(M, w) \vDash \chi(M, w, n) \wedge \psi(\overline{n})$. Since M started on input w does not halt, $\tau'(M, w) \vDash \chi(M, w, n) \wedge \psi(\overline{n})$ for all $n \in \mathbb{N}$. Note that by **??**, $\tau'(M, w) \vDash \overline{k} < \overline{n}$ for all $k < n$. Also $\psi(\overline{n}) \vDash \overline{k} < \overline{n} \rightarrow \overline{k} \neq \overline{n}$. So, $\mathfrak{M} \vDash \overline{k} \neq \overline{n}$ for all $k < n$, i.e., the infinitely many terms $\overline{k}$ must all have different values in $\mathfrak{M}$. But this requires that $|\mathfrak{M}|$ be infinite, so $\mathfrak{M}$ cannot be a finite model of $\tau'(M, w) \wedge \alpha(M, w)$. □

Problem und.3. Complete the proof of Lemma und.2 by proving that if M, started on input w, has not halted after n steps, then $\tau'(M, w) \vDash \psi(\overline{n})$.

tur:und:tra: thm:trakhtenbrodt

Theorem und.3 (Trakhtenbrot's Theorem). *It is undecidable if an arbitrary sentence of first-order logic has a finite model (i.e., is finitely satisfiable).*

Proof. Suppose there were a Turing machine F that decides the finite satisfiability problem. Then given any Turing machine M and input w, we could compute the sentence $\tau'(M, w) \wedge \alpha(M, w)$, and use F to decide if it has a finite model. By Lemmata und.1 and und.2, it does iff M started on input w halts. So we could use F to solve the halting problem, which we know is unsolvable. □

tur:und:tra: cor:fproof-incomp

Corollary und.4. *There can be no derivation system that is sound and complete for* finite *validity, i.e., a derivation system which has $\vdash \psi$ iff $\mathfrak{M} \vDash \psi$ for every finite structure $\mathfrak{M}$.*

Proof. Exercise. □

Problem und.4. Prove Corollary und.4. Observe that ψ is satisfied in every finite structure iff $\neg\psi$ is not finitely satisfiable. Explain why finite satisfiability is semi-decidable in the sense of **??**. Use this to argue that if there were a derivation system for finite validity, then finite satisfiability would be decidable.

Photo Credits

Bibliography