

und.1 The Decision Problem is Unsolvable

tur:und:uns:
sec
tur:und:uns:
thm:decision-prob

Theorem und.1. *The decision problem is unsolvable: There is no Turing machine D , which when started on a tape that contains a sentence ψ of first-order logic as input, D eventually halts, and outputs 1 iff ψ is valid and 0 otherwise.*

Proof. Suppose the decision problem were solvable, i.e., suppose there were a Turing machine D . Then we could solve the halting problem as follows. We construct a Turing machine E that, given as input the number e of Turing machine M_e and input w , computes the corresponding sentence $\tau(M_e, w) \rightarrow \alpha(M_e, w)$ and halts, scanning the leftmost square on the tape. The machine $E \frown D$ would then, given input e and w , first compute $\tau(M_e, w) \rightarrow \alpha(M_e, w)$ and then run the decision problem machine D on that input. D halts with output 1 iff $\tau(M_e, w) \rightarrow \alpha(M_e, w)$ is valid and outputs 0 otherwise. By ?? and ??, $\tau(M_e, w) \rightarrow \alpha(M_e, w)$ is valid iff M_e halts on input w . Thus, $E \frown D$, given input e and w halts with output 1 iff M_e halts on input w and halts with output 0 otherwise. In other words, $E \frown D$ would solve the halting problem. But we know, by ??, that no such Turing machine can exist. \square

tur:und:uns:
cor:undecidable-sat

Corollary und.2. *It is undecidable if an arbitrary sentence of first-order logic is satisfiable.*

Proof. Suppose satisfiability were decidable by a Turing machine S . Then we could solve the decision problem as follows: Given a sentence B as input, move ψ to the right one square. Return to square 1 and write the symbol \neg .

Now run the Turing machine S . It eventually halts with output either 1 (if $\neg\psi$ is satisfiable) or 0 (if $\neg\psi$ is unsatisfiable) on the tape. If there is a 1 on square 1, erase it; if square 1 is empty, write a 1, then halt.

This Turing machine always halts, and its output is 1 iff $\neg\psi$ is unsatisfiable and 0 otherwise. Since ψ is valid iff $\neg\psi$ is unsatisfiable, the machine outputs 1 iff ψ is valid, and 0 otherwise, i.e., it would solve the decision problem. \square

So there is no Turing machine which always gives a correct “yes” or “no” answer to the question “Is ψ a valid sentence of first-order logic?” However, there is a Turing machine that always gives a correct “yes” answer—but simply does not halt if the answer is “no.” This follows from the soundness and completeness theorem of first-order logic, and the fact that derivations can be effectively enumerated. explanation

tur:und:uns:
thm:valid-ce

Theorem und.3. *Validity of first-order sentences is semi-decidable: There is a Turing machine E , which when started on a tape that contains a sentence ψ of first-order logic as input, E eventually halts and outputs 1 iff ψ is valid, but does not halt otherwise.*

Proof. All possible **derivations** of first-order logic can be generated, one after another, by an effective algorithm. The machine E does this, and when it finds **a derivation** that shows that $\vdash \psi$, it halts with output 1. By the soundness theorem, if E halts with output 1, it's because $\models \psi$. By the completeness theorem, if $\models \psi$ there is **a derivation** that shows that $\vdash \psi$. Since E systematically generates all possible **derivations**, it will eventually find one that shows $\vdash \psi$, so will eventually halt with output 1. \square

Photo Credits

Bibliography