Weekly Challenge 08: Undecidability

mo08695

CS 212 Nature of Computation
Habib University
Fall 2025

1. The Biryani Theorem

Joseph Joestar does not believe in computability theory. He believes with the power of Hamon he can compute anything. Josuke however is well versed in theory of computation as he took CS 212 from Morioh university in fall 2025. Josuke keeps showing Koichi undecidable problems to mog him. In all this, their Koichi notices that there is a pattern in all the undecidable problems that Raahim and Taqi keep bringing. Whether it is from undecidability of Magic: The Gathering to E_{TM} there seem to be a pattern, that is that most of these problems are concerned with some questions regarding languages of Turing machines which are concerned with some general feature or quality regarding the language. Asking questions like "is the language of this Turing machine Empty" or "is the language of this Turing machine Regular". When discussing this with Josuke they conjecture that all such problems must be undecidable. If this statement is indeed true they can finally convince Joseph that Hamon can't compute everything.

Deciding a question Q regarding the language of a Turing machine means when given a Turing machine M we ask Q regarding the language of M, if the answer is "yes" then we accept M if the answer is "no" we reject. For example, if Q :"is the language of this Turing machine Regular", now if we ask this question regarding a Turing machine whose language is 0^n1^n then we reject that Turing machine but if we ask this regarding a Turing machine whose language is $\{0,1\}^*$ we accept that Turing machine. Now a question Q regarding the language of a Turing machine is said to be nontrivial if there exists at least one Turing machine whose language gives the answer "yes" on Q and there is at least one Turing machine whose language gives the answer "no" on Q. Show that any such nontrivial question Q regarding languages of Turing machines is undecidable.

To help them Jotaro Joestar a marine biologist has given a general idea for the proof. "For any such non trivial question regarding languages of Turing machines suppose the question is decidable then with the right construction you will have that if this question is decidable then a well known undecidable problem will alo be decidable therefore this question should not be decidable".

Solution: Let P be any nontrivial property of Turing-recognizable languages. Then the language

$$L_P = \{\langle M\rangle \mid L(M) \text{ has property } P\}$$

is undecidable.

Since P is nontrivial, there exist two Turing machines:
M_1 such that $L(M_1)$ has property P, and
M_0 such that $L(M_0)$ does *not* have property P.

we can show L_p is undecidable by reducing the undecidable language we all know:

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ accepts } w\}$$

to L_P. lets construct a new machine M' which works like this on an input x:

1. First, run M on w.
2. If M accepts w, then we, ignore M entirely and simulate M_1 on x. The output of M' will then match the behavior of M_1 on x.
3. If M never accepts w, then we, ignore M entirely and simulate M_0 on x. The output of M' will then match the behavior of M_0 on x.

Now given $\langle M, w\rangle$, using an algorithm we can produce $\langle M'\rangle$. So essentially the language that M' recognizes:

$$L(M') = \begin{cases} L(M_1), & \text{if } M \text{ accepts } w, \\ L(M_0), & \text{otherwise.} \end{cases}$$

Because P is a property of *languages*, the only thing that matters is which language is recognised by M'. So we can say that:

$$\langle M, w\rangle \in A_{TM} \quad \iff \quad M \text{ accepts } w \quad \iff \quad L(M') = L(M_1) \quad \iff \quad \langle M'\rangle \in L_P.$$

Since we were able to show a reduction from A_{TM} to L_p where L_p is the set of all turing machines who's lnagugage has the property P, so if L_p was decidable, A_{TM} would also be decidable but we know that its not therefore, L_p must be undecidable as well. Link to uploaded PDF (Sipser Notes)