Decidable instance, undecidable in general

Sophia Kolak

Problem: Suppose that for each positive integer t, we define a language $L(t)$ of binary strings. We can then define the language L_∞ corresponding to the countably infinite union

$$L = \cup_{t=1}^{\infty} L_t$$

to contain any binary string x that belongs to $L(t)$ for some value of t. Define languages $L(t)$ for each t such that each $L(t)$ is decidable (aka recursive), but L is not. You do not need to formally prove your answer, but you should intuitively explain why each $L(t)$ is decidable, what L is, and why L is not decidable.

Solution: Suppose we choose our language $L(t)$ to be:
$L(t) = \{< M, w, t > \ st\ M\ accepts\ w,\ but\ only\ in\ <\ t\ steps.\}$
Then for any specific choice of t, this algorithm would be decidable. All $L(t)$ would have to do, is keep track of the number of steps that have been performed on the Turing machine M. If the machine ever goes into an accepting state (even if less than t steps have been taken) then $L(t)$ outputs "yes". If M (being simulated on $L(t)$) takes t steps and is in a loop or is not in accepting state, then $L(t)$ outputs "no". There is no possibility of an infinite loop because t is an upper bound on the number of steps that the simulated machine has to take. After t steps have been taken, the machine is either in an accepting state or is not. This means there is always a clear output, which explains why the language $L(t)$ is in fact decidable. If, however, we take the infinite union of $L(t)$ for all t from 1-infinity, written as:

$$L = \bigcup_{t=1}^{\infty} L_t$$

Then we are taking the union of the sets:
$L_1 = \{M\ accepts\ w\ \ in\ <\ 1\ \ step\}$
$L_2 = \{M\ accepts\ w\ \ in\ <\ 2\ \ steps\}$
.
.
.
$L_{300000} = \{M\ accepts\ w\ \ in\ <\ 300000\ steps\}$
.
.
.

and so on forever....
Thus we have the set

$$L = L_1 \cup L_2 \cup ... \cup L_{300000} \cup ...$$

So L can be understood as the language which accepts w on M in any number of steps, “less than the infinite union” of possible steps. But deciding if a given string will be accepted in “less than an infinite number of steps” is the same as simply deciding if a string will be accepted in general. Based on this realization, it is clear that L_u (the language accepted by the universal TM) can be reduced to the language L. If we had a solution to L, we would necessarily have a solution to L_u, which is a contradiction, since L_u is undecidable. Since we know that L_u is not decidable, and L_u can be reduced to L, it must be the case that L too is not decidable.