

9640-5-1E AID: 1112 | 27/06/2014

RID: 6385 | 07/08/2015

Undecidable language:

The problem of determining whether a string or input can be accepted by a Turing machine or not is called Undecidability. The decidability of the Context-free grammar depends on the decidability of the Turing machine.

Proof to show that EQ_{CFG} is undecidable:

Step-1:

Consider a context-free grammar CFG $G_0 = (V, \Sigma, R, S)$ where $V = \{S\}$ and S is a starting variable. Assume that there is a rule $S \rightarrow IS$ in R for every terminal $I \in \Sigma$. The grammar G_0 includes a ϵ notation by using the rule $S \rightarrow \epsilon$.

Example:

For the CFG, the rules in G_0 are defined as $S \rightarrow aS \mid bS \mid \epsilon$ over the alphabet set $\Sigma = \{a, b\}$. So, the grammar CFG G_0 satisfies all the alphabets in the alphabet set Σ .

So, $L(G_0) = \Sigma^*$. Thus, the Turing Machine is decidable.

Step-2:

Assume that the CFG is decidable by using the Turing machine R that decides EQ_{CFG} . Construct another Turing machine S which uses R to decide ALL_{CFG} by using the following procedure:

$S = \text{On input } \langle G_0 \rangle,$

1. Run R on the input $\langle G_0, G_1 \rangle$. G_1 is a CFG, which generates Σ^* .
2. Accept the grammar, when R accepts.
3. Otherwise reject.

Thus, if the Turing machine R decides EQ_{CFG} , S also decides ALL_{CFG} which is impossible. So, EQ_{CFG} is also undecidable.