Undecidable problems on CFLs

Video Lecture "Undecidable problems about CFLs", related practice problems and their solutions are on http://cse.iitkgp.ac.in/ abhij/course/theory/FLAT/Spring20/

A decidable problem on CFGs

Problem: Given a CFG G, does it generate some strings?

This problem is decidable due to the following algorithm. We design a marking scheme similar to the DFA minimisation algorithm. Let $G = (N, \Sigma, P, S)$. At the first step, we mark all terminal symbols in Σ. In consequent steps, when there is a production $A \to \beta$ such that all symbols in β are marked in previous steps, then in the current step we mark the non-terminal symbol A. We do this marking exhaustively as long as possible. When we stop, if we have marked the non-terminal start symbol S then we say that G generates some string. Otherwise we say that $L(G) = \emptyset$. (Try to prove the correctness of this algorithm using induction and the definition of generation of strings from S.)

An Undecidable problem on CFGs

Let us look at another variant:

Problem: Given a CFG G, does it generate all strings?

This problem is undecidable. We can give a reduction from the Halting Problem to this problem. Here are the steps to show that the problem is undecidable.

1. We introduce an intermediate problem called $VALCOMPS$ over an alphabet Δ.
2. We show that $\overline{VALCOMPS}$ is a CFG.
3. The definition of $VALCOMPS$ will imply that there is a reduction from HP to $VALCOMPS$ (in fact it is easier to get a reduction from $\overline{HP}$ to $\overline{VALCOMPS}$), such that
   for an instance $M\#x$ of HP,
   M halts on $x \implies L(VALCOMPS) \subseteq \Delta^* \implies L(\overline{VALCOMPS}) = \Delta^* - L(VALCOMPS) \subseteq \Delta^*$
   M does not halt on $x \implies L(VALCOMPS) = \emptyset \implies L(\overline{VALCOMPS}) = \Delta^*$.

4. Thus, we conclude that $\{G|$ the CFG G generates all strings $\}$ is a non-recursive set.

Please refer to Kozen Lecture 35 and the slides on the course page for full details of points 1 and 2 above.