Decidable problems for CFLs.

Richard Cole

November 5, 2007

Problem 1 \( A_{CFG} = \{ < G, w > \mid G \text{ is a CFG which can generate } w \} \).

Claim \( A_{CFG} \) is decidable.

Proof The following procedure decides \( A_{CFG} \).

Step 1 Convert \( G \) to a CNF grammar \( \tilde{G} \), with start symbol \( S \).

Step 2 If \( w = \epsilon \) check if \( S \rightarrow \epsilon \) is a rule of \( \tilde{G} \) and if so accept (output “accept”) and otherwise reject.

Step 3 If \( w \neq \epsilon \), the derivation of \( w \) takes \( 2|w| - 1 \) steps. Simply generate, one by one, all possible derivations in \( \tilde{G} \) of length \( 2|w| - 1 \). If any of them yield \( w \) then accept; otherwise, reject.

\( \square \)

Problem 2 \( E_{CFG} = \{ < G > \mid G \text{ is a CFG and } L(G) = \phi \} \).

Claim \( E_{CFG} \) is decidable.

Proof Note that \( L(G) \neq \phi \) if and only if \( G \) can generate some string of terminals, or the empty string. We simply determine this property for each variable \( u \) in \( G \): can \( u \) generate a string in \( \Sigma^* \)? This can be done by means of the following marking procedure.

Step 1 Convert the grammar to CNF (this just simplifies the rest of the description).

Step 2 Mark each variable \( A \) for which there is a rule \( A \rightarrow a \) or \( A \rightarrow \epsilon \) (the latter could only apply to \( S \), the start variable).

Step 3 Iteratively, mark each variable \( A \) such that there is a rule \( A \rightarrow BC \) and \( B \) and \( C \) are already marked. (We leave an efficient implementation to the reader). Stop when no more variables can be marked.

Step 4 Accept if \( S \) is marked and reject otherwise.

\( \square \)