Theory of Computation, CSCI 438 spring 2022
Decidability Problems Concerning Context-Free Languages, pg. 198-201, April 6

1. Acceptance problem for CFG.

    A_{CFG} = {<G, w> | G is a CFG, w is a string in the language of G, and w $\in \mathcal{L}$ (G)} decidable?

    Is A_{CFG} decidable?

    M= “On input <G,w> that encodes a CFG and a string in the grammar

    1. Convert the grammar to Chomsky Normal Form (if we’ve covered it) otherwise… one limited to rules of the form $S \rightarrow \varepsilon$ or $v \rightarrow x$ for $v \in V$, $x \in (V \cup \Sigma)^+$.
    2. Generate all possible derivation trees of 2*|w| steps. If w is generated by one of these derivations accept. Otherwise reject.”

2. Empty CFG.

    E_{CFG} = {<G> | G is a CFG whose language is empty}

    Is E_{CFG} decidable?

    M= “On input <G> that encodes a CFG

    1. Consider terminals as marked.
    2. While (number of marked items is increasing)
        For each rule where all variables and symbols on the right are marked, mark the variable on the left.
    3. If the start variable is marked, reject, otherwise, accept.”

3. Equivalent CFGs

   EQ_{CFG} = {<G,F> | G and F are CFGs that generate the same language} decidable?

   Is EQ_{CFG} decidable?

   We'd like to the fact that A and B are equivalent iff $(A \cap \overline{B}) \cup (\overline{A} \cap B) = \Phi$. However, while $\cup$ is easy with grammars, complementation is not. In fact, the class of context-free languages is not closed under complementation.

   The answer to this is no. EQ_{CFG} is not decidable.

4.3 Let ALL_{DFA} = {<A> | A is a DFA and $\mathcal{L}(A) = \Sigma^*$}.
Show that ALL_{DFA} is decidable.

D = "On input <M> that encodes a DFA

1. Translate M to a machine that accepts the complement of what M accepted (i.e. make non-accepting states accepting, and vice versa)
   i. Run E_{DFA} on the new machine.
   ii. If E_{DFA} accepts the complement of M, accept; If E_{DFA} rejects the complement of M, reject"

4.24 A **useless state** in a pushdown automaton is never entered on any input string. Consider the problem of determining whether a pushdown automaton has any useless states. Formulate this problem as a language and show that it is decidable.

Let $USELESS_{PDA}$ = {<M> | M is PDA which contains at least one useless state}

The following defines a decider for $USELESS_{PDA}$.

D = "On input <M> that encodes a PDA

1. Do for each state q in M
   Modify M so that q is the only accept state in M.
   Run E_{PDA} (decider is defined in the text). If E_{PDA} accepts, accept.
2. Reject because no states are useless."