CS21004 Formal Languages and Automata Theory, Spring 2019–2020

Undecidable Problems about Context-Free Languages

---

1. You are given two CFGs G and G'. Prove that the following problems are undecidable.

(a) whether $\mathscr{L}(G) \subseteq \mathscr{L}(G')$,

Solution Suppose that the problem is decidable. Let D be a decider for the given problem. Let G' be a grammar over Σ. We want to decide whether $\mathscr{L}(G') = \Sigma^*$. Generate a grammar G with $\mathscr{L}(G) = \Sigma^*$. Invoke the decider D with G, G' as input. If D outputs *yes*, we have $\Sigma^* \subseteq \mathscr{L}(G')$, so $\mathscr{L}(G') = \Sigma^*$. If D outputs *no*, then $\Sigma^* \supsetneqq \mathscr{L}(G')$. Thus, we have a decider for the problem whether $\mathscr{L}(G') = \Sigma^*$, a contradiction. This is indeed a reduction

$$\{G' \mid \mathscr{L}(G') = \Sigma^*\} \leqslant_m \{G \,\#\, G' \mid \mathscr{L}(G) \subseteq \mathscr{L}(G')\}.$$

The reduction is correct because $\mathscr{L}(G) \subseteq \mathscr{L}(G')$ if and only if $\mathscr{L}(G') = \Sigma^*$.

(b) whether $\mathscr{L}(G) = \mathscr{L}(G)\mathscr{L}(G)$.

Solution Make a reduction $\overline{\text{HP}} \leqslant_m \{G \mid \mathscr{L}(G) = \mathscr{L}(G)\mathscr{L}(G)\}$. The input is $M \,\#\, w$ (an instance of $\overline{\text{HP}}$), and the output is a CFG G such that $\mathscr{L}(G) = \mathscr{L}(G)\mathscr{L}(G)$ if and only if M does not halt on w. We take G to be a grammar for $L = \overline{\text{VALCOMP}(M,w)}$.

If M does not halt on w, then $\text{VALCOMP}(M,w) = \emptyset$ and so $L = \mathscr{L}(G) = \Delta^*$, and we have $\Delta^* = \Delta^*\Delta^*$.

If M halts on w, then $\text{VALCOMP}(M,w) \neq \emptyset$. Let $\gamma = \#C_0\#C_1\#C_2\#\ldots\#C_N\#$ be a valid computation history of M on w. Take $\alpha = \#C_0$ and $\beta = \#C_1\#C_2\#\ldots\#C_N\#$. Then, α is not a valid computation history of M on w, because the string does not end with #. Moreover, β too is not a valid computation history of M on w, because the head is not at the leftmost cell (the left end-marker) in the first configuration. Therefore $\alpha, \beta \in L$, whereas $\gamma = \alpha\beta \notin L$. That is, in this case, $\mathscr{L}(G)$ does not satisfy $\mathscr{L}(G) = \mathscr{L}(G)\mathscr{L}(G)$.

2. Prove that the following problems are undecidable.

(a) whether a CFL is a DCFL.

Solution The same reduction from $M \,\#\, w$ (an instance of $\overline{\text{HP}}$) to a CFG G for $L = \overline{\text{VALCOMP}(M,w)}$ works.

If M does not halt on w, then $L = \mathscr{L}(G) = \Delta^*$ which is definitely a DCFL.

If M halts on w, then a string $\alpha \in \Delta^*$ may be in L for multiple reasons simultaneously, like:
(1) the final configuration is not a halting configuration,
(2) there is a halting state followed by a non-halting state,
(3) inconsistent head movement somewhere,
and so on. This means α may have multiple parse trees. So L is not a DCFL in this case.

(b) whether the complement of a CFL is a CFL.

Solution Again the reduction of $M \,\#\, w$ to a CFG G for $L = \overline{\text{VALCOMP}(M,w)}$ works.

If M does not halt on w, then $\mathscr{L}(G) = L = \Delta^*$, the complement of which is $\emptyset$ (a CFL).

If M halts on w, then $\overline{\mathscr{L}(G)} = \text{VALCOMP}(M,w) \neq \emptyset$ is not context-free.

3. For a TM M and an input w for M, define

$$\begin{aligned}\text{VALCOMP-ALT}(M,w) \;=\; & \{\#C_0\#C_1^R\#C_2\#C_3^R\#C_4\#C_5^R\#\ldots\#C'_N\# \mid C_0, C_1, C_2, \ldots, C_N \\ & \text{is a valid computation history of } M \text{ on } w\},\end{aligned}$$

where $C'_N = \begin{cases} C_N & \text{if } N \text{ is even,} \\ C_N^R & \text{if } N \text{ is odd,}\end{cases}$ (here α^R is the reverse of the string α). Like $\text{VALCOMP}(M,w)$, the language $\text{VALCOMP-ALT}(M,w)$, if non-empty, is not context-free. Prove the following facts.

(a) $\overline{\text{VALCOMP-ALT}(M,w)} = \Delta^* \setminus \text{VALCOMP-ALT}(M,w)$ is context-free.

Solution Now, since two consecutive configurations are in opposite order, a DPDA can check whether two consecutive configurations *are* consistent or not (elaborate the construction). Therefore if $\alpha \in \Delta^*$ is syntactically correct but contains some inconsistent changes, one such change can be nondeterministically guessed by an NPDA. The NPDA guesses i, and then simulates the above DPDA for finding the inconsistency between C_i and C_{i+1}.

(b) VALCOMP-ALT(M,w) is the intersection of two DCFLs.

Solution We can use the consistency-checking DPDA of Part (a) to accept the languages

$$\text{VALCOMP-ALT}_{even}(M,w) = \left\{\#C_0\#C_1\#C_2\#\ldots\#C_N \mid C_{i+1} \text{ is consistent with } C_i \text{ for all even } i\right\}$$

and

$$\text{VALCOMP-ALT}_{odd}(M,w) = \left\{\#C_0\#C_1\#C_2\#\ldots\#C_N \mid C_{i+1} \text{ is consistent with } C_i \text{ for all odd } i\right\}.$$

So VALCOMP-ALT$_{even}(M,w)$ and VALCOMP-ALT$_{odd}(M,w)$ are DCFLs. Moreover,

$$\text{VALCOMP-ALT}(M,w) = \text{VALCOMP-ALT}_{even}(M,w) \bigcap \text{VALCOMP-ALT}_{odd}(M,w).$$

4. Prove that the following problems are undecidable.

(a) Whether the intersection of two CFLs is empty.

Solution Reduction from $\overline{\text{HP}}$. Given $M \# w$, generate the grammars G_{even}, G_{odd} for VALCOMP-ALT$_{even}(M,w)$ and VALCOMP-ALT$_{odd}(M,w)$, respectively, and output $G_{even} \# G_{odd}$.

If M does not halt on w, there are no valid computation histories of M on w, so

$$\mathscr{L}(G_{even}) \cap \mathscr{L}(G_{odd}) = \text{VALCOMP-ALT}(M,w) = \emptyset.$$

(Note that in this case, $\mathscr{L}(G_{even})$ is not empty, because C_{2i+2} need not be consistent with C_{2i+1}. Likewise, for $\mathscr{L}(G_{odd})$.)

If M halts on w, there are valid computation histories of M on w, so

$$\mathscr{L}(G_{even}) \cap \mathscr{L}(G_{odd}) = \text{VALCOMP-ALT}(M,w) \neq \emptyset.$$

(b) Whether the intersection of two CFLs is a CFL.

Solution The same reduction of Part (a) works, since $\emptyset$ is a CFL, whereas a non-empty VALCOMP-ALT(M,w) is not.

(c) Whether the union of two DCFLs is a DCFL.

Solution Since DCFLs are closed under complement, the complements of the languages VALCOMP-ALT$_{even}(M,w)$ and VALCOMP-ALT$_{odd}(M,w)$ in Δ^* are also DCFLs. Let $\hat{G}_{even}$ and $\hat{G}_{odd}$ be DCFGs for these complements.

Use reduction from $\overline{\text{HP}}$. Given $M \# w$, generate and output $\hat{G}_{even} \# \hat{G}_{odd}$.

If M does not halt on w, $\mathscr{L}(\hat{G}_{even}) \cup \mathscr{L}(\hat{G}_{odd}) = \overline{\text{VALCOMP-ALT}(M,w)} = \Delta^*$ is a DCFL.

If M halts on w, $\mathscr{L}(\hat{G}_{even}) \cup \mathscr{L}(\hat{G}_{odd}) = \overline{\text{VALCOMP-ALT}(M,w)} \neq \Delta^*$ is not a DCFL.

5. Prove that the finiteness problem for regular and context-free languages is decidable.

Solution Use the pumping lemma. Let L be a regular/context-free language, and k a pumping-lemma constant for L. We know (see class test) that L is infinite if and only if L contains a string of length in the range $[k, 2k-1]$. So it suffices to check the membership in L of all the strings of lengths in this range.

Remark: In many of these exercises, you have a reduction algorithm that generates one or more CFGs. Note that a CFG cannot simulate a TM on an input. So you need to generate CFGs that can describe certain assertions about the working of M on w in a context-free manner. $\overline{\text{VALCOMP}}$ and its cousins are typical examples. Note also that the reduction algorithm R is a TM, and can simulate M on w for producing the output. But R must be a total TM, so it cannot wait for an infinite simulation (looping) of M on w.

Suppose that M halts on w. In some of the exercises, we have used the following results without proofs.

- VALCOMP(M,w) and VALCOMP-ALT(M,w) are not CFL's, in general.
- Their complements are not DCFL's, in general.
- A DCFG is necessarily unambiguous.