582206 Models of Computation (Autumn 2010)
Exercise 8 (8–11 November)

Basic exercises

The first three problems are basic applications of the material from the text book. Solve them by yourself; if there is anything unclear you can ask about it during the exercise session.

1. Consider the context-free grammar

$$\begin{aligned} S &\to SAB \mid \varepsilon \\ A &\to \mathrm{a}A \mid \mathrm{a} \\ B &\to \mathrm{b}B \mid \varepsilon \end{aligned}$$

   Give two parse trees and the corresponding leftmost derivations for the string aa.

2. Convert the grammar

$$\begin{aligned} S &\to ASB \mid B \\ C &\to \mathrm{c}C \mid \varepsilon \\ A &\to C\mathrm{a}AC \mid \mathrm{a} \\ B &\to \mathrm{abc} \mid A \end{aligned}$$

   into Chomsky normal form using the procedure given in the textbook.

3. [Sipser Exercise 2.9] Give a context-free grammar that generates the language

$$\{\, \mathrm{a}^i \mathrm{b}^j \mathrm{c}^k \mid i = j \text{ or } j = k \,\} .$$

   Is your grammar ambiguous? Why or why not?

Continues on the next page!

Discussion problems

Read the following problems and make sure you are familiar with the necessary basic concepts. You are not expected to solve the problems by yourself; we shall discuss them together.

4. Consider the grammar

$$\begin{aligned} \langle \texttt{stmt} \rangle &\rightarrow \langle \texttt{if-then-else} \rangle \mid \langle \texttt{if-then} \rangle \mid \texttt{p} \\ \langle \texttt{if-then-else} \rangle &\rightarrow \textbf{if}\ \texttt{b}\ \textbf{then}\ \langle \texttt{stmt} \rangle\ \textbf{else}\ \langle \texttt{stmt} \rangle \\ \langle \texttt{if-then} \rangle &\rightarrow \textbf{if}\ \texttt{b}\ \textbf{then}\ \langle \texttt{stmt} \rangle \quad . \end{aligned}$$

   To clarify, the set of terminal is $\{\, \textbf{if}, \textbf{then}, \textbf{else}, \texttt{b}, \texttt{p} \,\}$; for readability, some space has been added between the terminals.

   Show that the grammar is ambiguous. Desing an unambiguous grammar for the same language.

5. Call a variable A of a context-free grammar

   - *unreachable* if from the start symbol of the grammar it is impossible to derive any string containing A, and
   - *unproductive* if from A it is impossible to derive any string consisting entirely of terminals.

   As an example, in the grammar

$$\begin{aligned} S &\rightarrow A \mid BC \\ A &\rightarrow \mathrm{a}A \mid \varepsilon \\ B &\rightarrow \mathrm{b}B \mid \varepsilon \\ D &\rightarrow \mathrm{ab} \\ E &\rightarrow C \end{aligned}$$

   the variable C is unproductive, D unreachable and E both unproductive and unreachable.

   (a) Give an algorithm for finding all unreachable variables in a grammar.

   (b) Give an algorithm for finding all unproductive variables in a grammar.

   (c) Suppose we try to remove all "useless" variables by first removing all unreachable variables and rules involving them, and then, from what remains, further removing all the unproductive variables and rules involving them. Does this lead to the desired result? Why?

6. [Sipser Problem 2.25] For a language A over the alphabet Σ, define its set of suffixes

$$\textsc{Suffix}(A) = \{\, v \in \Sigma^* \mid uv \in A \text{ for some } u \in \Sigma^* \,\} \,.$$

   Prove that if A in context free, then so is $\textsc{Suffix}(A)$.