Note: Because of the exam week, there are no tutorial sessions on 21–25 February. Sessions resume again in the following week.

Instructions

- Classroom Problems C6.1–C6.2 will be discussed and solved onsite at the tutorial sessions in lecture week 6. No credit is given for these problems.
- Homework Problems H6.1–H6.3 you should solve on your own, and be available to present your solutions at one of the tutorial sessions in lecture week 7. In order to get course credit, you need to indicate your solved problems on the signup sheet circulated at the beginning of the session.
- Supplementary Problems S6.1–S6.2 provide further illustration and extension of the course material, but will usually not be covered at the tutorials. You are however invited to work on these problems too, and discuss them with the course staff. Sample solutions are provided on MyCourses.

Classroom Problems

C6.1 Consider the following grammar generating a certain type of list structures:

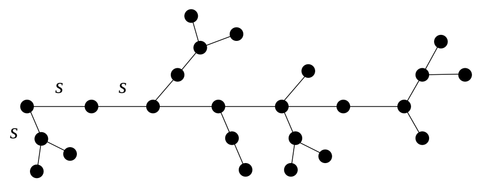
$$S \rightarrow (S) \mid S, S \mid a.$$

(i) Based on the above grammar, give a leftmost and rightmost derivation and a parse tree for the sentence "$(a, (a))$".

(ii) Prove that the grammar is ambiguous.

(iii) Design an unambiguous grammar generating the same language.

C6.2 A *fern* consists of a stem and a number of subferns rooted on the left and right sides of the stem. For instance, the following structure is a fern:



A fern structure can be described by a string where each unit of the stem is denoted by a letter s, and each subfern is described by a similar string in parentheses, located at the point where the subfern is rooted, and prefixed by l or r depending on whether the subfern occurs on the left or right side of the main stem, respectively. At most one subfern can be rooted to the left and to the right at each point, and each subfern must contain at least one stem unit. For instance, the string representation corresponding to the above example is

$$r(sl(s)r(s))ssl(ssl(s)r(s))sr(ss)sl(s)r(sl(s)r(s))ssl(sr(s)s)r(s).$$

Design a context-free grammar describing the structure of such fern strings.

Homework Problems

H6.1 (a) Design a context-free grammar for the language

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

(b) Show that the grammar you gave in part (a) is ambiguous.

H6.2 (a) Design a context-free grammar for the language

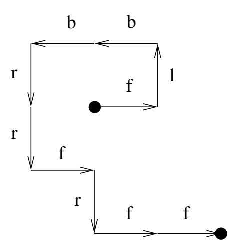
$$L = \{w \in \{a, b\}^* \mid w \text{ contains equally many } a\text{'s and } b\text{'s}\}.$$

Draw the corresponding parse trees for sentences *aabb*, *abab* and *baab*.

(b) Is the grammar you designed in part (a) ambiguous or unambiguous? If it is ambiguous, then try to design also an unambiguous grammar for the language.

(c) Prove (precisely!) that the language in part (a) is not regular.

H6.3 A *party walk* is a sequence of consequent steps, whose direction with respect to the starting point is either forward (abbr. f), backward (b), left (l) or right (r). For instance, the sequence *flbbrrfrff* describes the following walk, whose total result is to move the walker a distance of two steps forward (and concurrently two steps to the right):



Design a context-free grammar that generates all party walks whose total result is to move the walker at least one step forward from the starting point (ignoring any possible sideways movement).

Supplementary Problems

S6.1

(a) Prove that the following context-free grammar is ambiguous:

$$\begin{aligned} S &\rightarrow \textbf{if } b \textbf{ then } S \\ S &\rightarrow \textbf{if } b \textbf{ then } S \textbf{ else } S \\ S &\rightarrow s \end{aligned}$$

(b) Design an unambiguous grammar that is equivalent to the grammar in item (a), i.e. that generates the same language. (*Hint:* Introduce new variables B and U that generate, respectively, only "balanced" and "unbalanced" **if-then-else-**sequences.)

S6.2 Design a recursive-descent (top-down) parser for the grammar of the "programming language" discussed in Supplementary Problem S5.2 of Problem Set 5.