T-79.148 **Autumn 2002**
Introduction to Theoretical Computer Science
Tutorial 6, 23–25 October
Problems

Homework problems:

1. Design a context-free grammar that characterises, at an appropriately general level, the structure of a typical newspaper article: headline, caption, text body, subheadings etc.

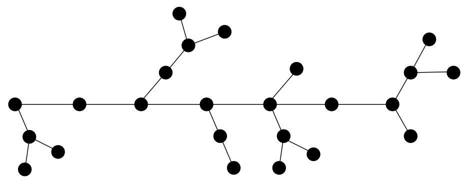
2. A *palindrome* is a string w such that $w = w^R$, e.g. "MADAMIMADAM", "ABLEWASIEREISAWELBA". Consider the set of palindromes over the alphabet $\{a, b\}$:

$$\text{PAL} = \{w \in \{a, b\}^* \mid w = w^R\}.$$

   (a) Prove that this language is not regular.

   (b) Design a context-free grammar generating the language.

3. 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 would be:

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

Design a context-free grammar describing the structure of ferns. (I.e. the grammar should generate all and only the valid fern strings.)

Demonstration problems:

4. *Pattern expressions* are a generalisation of regular expression used e.g. in some text editing tools of UN*X-type operating systems. In addition to the usual regular expression constructs, a pattern expression may contain string variables, inducing the constraint that any two appearances of the same variable must correspond to the same substring. Thus e.g. aXb^*Xa and $aX(a \cup b)^*YX(a \cup b)^*Ya$ are pattern expressions over the alphabet $\{a, b\}$. The first one of these describes the language $\{awb^nwa \mid w \in \{a, b\}^*,\ n \geq 0\}$. Prove that pattern expressions are a proper generalisation of regular expressions, i.e. that pattern expressions can be used to describe also some nonregular languages.

PLEASE TURN OVER

5. Prove that the language $\{w \in \{a,b\}^* \mid w \text{ contains equally many } a\text{'s and } b\text{'s}\}$ is not regular, and design a context-free grammar generating it.

6. Design a context-free grammar describing the syntax of simple "programs" of the following form: a program consists of nested `for` loops, compound statements enclosed by `begin-end` pairs and elementary operations `a`. Thus, a "program" in this language looks something like this:

```
a;
for 3 times do
begin
   for 5 times do a;
   a; a
end.
```

   For simplicity, you may assume that the loop counters are always integer constants in the range 0,..., 9.