**T-79.1001/1002** **Autumn 2005**
**Introduction to Theoretical Computer Science T/Y**
**Tutorial 6, Problems**

**Week 43 (24–28 Oct) is exam week, and this course has neither lectures nor tutorials on that week. The final exam for the short version (T-79.1002, 2 cr) of the course takes place on Thu 27 Oct, and covers the topics discussed on the course so far (Lectures 1–6). You must register for the exam via the TOPI system by Fri 21 Oct. Also, all the compulsory Regis problems must have been solved by the time of the exam. The exam for the long version of the course (T-79.1001, 4 cr) takes place on Wed 14 Dec. Participants in the long version of the course should NOT go to the exam on 27 Oct.**

**Homework problems:**

1. The languages produced by the following context-free grammars are regular. Construct the regular expressions corresponding to them.

   (a) $\{S \to AS \mid \varepsilon, \quad A \to a \mid b\}$

   (b) $\{S \to aSa \mid aSb \mid bSa \mid bSb \mid \varepsilon\}$

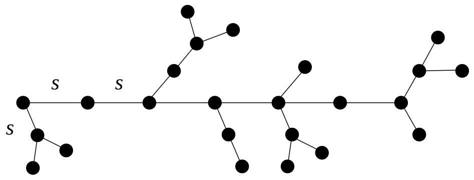
2. Design right-linear grammars for the following languages:

   (a) $\{w \in \{a,b\}^* \mid w$ does not contain $abb$ as a substring$\}$;

   (b) $\{w \in \{0,1\}^* \mid w$ contains an even number of both 0's and 1's$\}$.

   (Cf. Problem 2/3(c) & 3/1.)

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 such fern strings.

**PLEASE TURN OVER**

**Demonstration problems:**

4. In the modern WWW page description language XML, designers can construct their own "data type definitions" (abbr. DTD), which are essentially context free grammars describing the structure of the text or other data displayed on the page. Acquaint yourself with the notation used in this XML/DTD description language (from e.g. `http://www.rpbourret.com/xml/xmldtd.htm`), and give a context-free grammar corresponding to the following XML/DTD description:

```
<!DOCTYPE Book [
   <!ELEMENT Book (Title, Chapter+)>
   <!ATTLIST Book Author CDATA #REQUIRED>
   <!ELEMENT Title (#PCDATA)>
   <!ELEMENT Chapter (#PCDATA)>
   <!ATTLIST Chapter id ID #REQUIRED>
]>
```

5. Design a recursive-descent (top-down) parser for the grammar from Problem 5 of the previous tutorial.