T-79.148 **Spring 2005**
Introduction to Theoretical Computer Science
Tutorial 6, 1–4 March
Problems

Homework problems:

1. Consider the following context-free grammars:

   (a) $A \to aAcc \mid B$, $B \to bBc \mid \varepsilon$ (b) $S \to +S- \mid SS \mid \varepsilon$

   Give a derivation for the sentence *abccc* according to grammar (a), and a derivation for the sentence $++-+--+-$ according to grammar (b). Describe the language generated by each grammar verbally as simply as you can.

2. A *palindrome* is a string w such that $w = w^R$. (E.g. "MADAMIMADAM", "ABLEWASIEREISAWELBA," cf. `http://www.palindromes.org/`.) Consider the set of palindromes over the alphabet $\{a, b\}$:

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

   (a) Design a context-free grammar generating the language. (*Hint:* Note that a string $w \in \text{PAL}$, if and only if it is of the form $w = uXu^R$, where $X = a$, b or ε.)

   (b) Prove that this language is not regular. (*Hint:* Consider strings of the form $a^n b a^n$.)

3. Design context-free grammars for the following languages:

   (a) $\{a^m b^n \mid 0 \le m \le 2n\}$

   (b) $\{ucv \mid u, v \in \{a,b\}^* \text{ and } |u| = |v|\}$

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.h` 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. Prove that the language $\{w \in \{a,b\}^* \mid w$ contains equally many a's and b'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.