Exercise 1: Formal Grammars and Languages

Formal Methods II, Fall Semester 2013

Distributed: 27.09.2013
Due Date: 11.10.2013
Send your solutions to tobias.klauser@uzh.ch or deliver them in class.

Formal Languages

1. (6 points) For each of the following languages, indicate whether the language is regular, context-free or context-sensitive, and provide a generative grammar.

   E.g. $L = \{\mathrm{a}^n\mathrm{b}^n \mid n \geq 0\}$

   Solution: This *context-free* language can be generated by the grammar

   $$S \to aSb \mid \epsilon$$

   (a) $L = \{\mathrm{a}^m\mathrm{b}^n \mid m > 0 \wedge n \geq 0\}$

   (b) All strings over $\{$a, b, c$\}$ that contain an even number of a's.

   (c) All strings over $\{$0,1$\}$ with the following property: The string read from left to right shall be the complement of the string read from right to left. A complement of a string is formed by replacing every 0 by an 1 and vice versa (e.g. 101 is the complement of 010). For example, 01, 0101 and 0011 are in the language while 1, 11 and 101 are not.

   (d) (2 bonus points) All strings over $\{$a,b$\}$ that *are* palindromes. A palindrome is a string which reads the same in either direction. For example, aba, abbaabba, b are all in the language, while ab, aab, aabaaa are all not in the language.

Regular Expressions

2. (3 points) For each of the following regular expressions, give a minimum-length string over the alphabet $\{$0,1$\}$ that is *not* in the language:

   (a) 1*(01)*0*

   (b) 0*(10 | 1)*0*

   (c) (0* | 1*)(1* | 0*)(0* | 1*)

3. (6 points) Give a regular expression to generate each of the following languages:

   (a) All strings over $\{0{,}1\}$ that represent positive binary numbers divisible by 4

   (b) All strings over $\{0{,}1\}$ that have an even number of 0's

   (c) All strings over $\{0{,}1\}$ in which every 1 is followed immediately by 00

   (d) All strings over $\{a{,}b\}$ that contain either the substring aa or aba

   (e) All strings over $\{a{,}b\}$ that do *not* contain the substring ab

   (f) All strings over $\{a{,}b\}$ that do *not* contain the substring aa

4. (3 points) For each of the following regular expressions, give an as short regular grammar as possible that defines the corresponding language:

   (a) (ab*a)+ (b) a*b*a* (c) 01(0 | 1)*

   Note: the grammar should be written in a *regular* form (either left or right regular – see section 2.4 in the lecture script).

5. (2 points) Imagine that you, as a brilliant hacker, want to find out the login name of a particular person. You have found, on a Unix system, a directory with several log files that you suspect contain the information you're looking for. You remember that you can use the `grep` utility to search for text patterns. For instance, you can enter the command

```
grep -E "abc" *
```

   to search all files for lines containing the sequence of characters `abc`. (The option `-E` is used to indicate that you're using the extended syntax for regular expressions – see page 2-13 of the lecture script.)

   (a) What command would you enter to search all files for lines that *start* with a potential login name? (Remember that a login name consists only of letters and digits, and must start with a letter.)

   (b) You realize that you obtain far too many hits with the above command. Let us thus assume that the login name you're looking for is at least 5 characters long. How would you refine your command accordingly?

   (c) (1 bonus point) How would you further refine your command to search for potential login names that are not necessarily at the beginning of the line, but that are either at the beginning or the end of the line, or preceded or followed by whitespace characters?

Have fun!