# Homework 5

**Please submit your solutions via email. Please submit only PDFs! Every solution should be preceeded by the corresponding question text, like:**

| | |
|---|---|
| **Q$n$:** | **…a difficult question from me…** |
| **A:** | **…an answer from you …** |
| **Q$n+1$** | **…another difficult question…** |
| **A:** | **…another brilliant answer from you…** |

**Solutions will only be accepted until 20th December! Please send only one homework per email.**

1. Consider the basic regular expressions

$$r ::= \mathbf{0} \mid \mathbf{1} \mid c \mid r_1 + r_2 \mid r_1 \cdot r_2 \mid r^*$$

   and suppose you want to show a property $P(r)$ for all regular expressions $r$ by structural induction. Write down which cases do you need to analyse. State clearly the induction hypotheses if applicable in a case.

2. Define a regular expression, written $ALL$, that can match every string. This definition should be in terms of the following extended regular expressions:

$$r ::= \mathbf{0} \mid \mathbf{1} \mid c \mid r_1 + r_2 \mid r_1 \cdot r_2 \mid r^* \mid \sim r$$

3. The *not*-regular expression is definitely useful for recognising comments for example, but can sometimes be quite unintuitive when it comes to deciding which strings are matched or not. Consider

$$(\sim a)^* \quad \text{and} \quad \sim (a^*)\,.$$

   What is the language of each regular expression?

4. Define the following regular expressions

| | |
|---|---|
| $r^+$ | (one or more matches) |
| $r^?$ | (zero or one match) |
| $r^{\{n\}}$ | (exactly $n$ matches) |
| $r^{\{m..n\}}$ | (at least $m$ and maximal $n$ matches, with the assumption $m \leq n$) |

   in terms of the usual basic regular expressions

$$r ::= \mathbf{0} \mid \mathbf{1} \mid c \mid r_1 + r_2 \mid r_1 \cdot r_2 \mid r^*$$

5. Give the regular expressions for lexing a language consisting of identifiers, left-parenthesis `(`, right-parenthesis `)`, numbers that can be either positive or negative, and the operations `+`, `-` and `*`.

   Decide whether the following strings can be lexed in this language?

   (a) `"(a3+3)*b"`

   (b) `")()++-33"`

   (c) `"(b42/3)*3"`

   In case they can, give the corresponding token sequences. (Hint: Observe the maximal munch rule and the priorities of your regular expressions that make the process of lexing unambiguous.)

6. Suppose the following context-free grammar $G$

   $$\begin{array}{l} \boldsymbol{S} ::= \boldsymbol{A} \cdot \boldsymbol{S} \cdot \boldsymbol{B} \mid \boldsymbol{B} \cdot \boldsymbol{S} \cdot \boldsymbol{A} \mid \epsilon \\ \boldsymbol{A} ::= a \mid \epsilon \\ \boldsymbol{B} ::= b \end{array}$$

   where the starting symbol is $\boldsymbol{S}$. Which of the following strings are in the language of $G$?

   - $a$
   - $b$
   - $ab$
   - $ba$
   - $bb$
   - $baa$

7. Suppose the following context-free grammar

   $$\boldsymbol{S} ::= a \cdot \boldsymbol{S} \cdot a \mid b \cdot \boldsymbol{S} \cdot b \mid \epsilon$$

   Describe which language is generated by this grammar.

8. Remember we have specified identifiers with regular expressions as strings that start with a letter followed by letters, digits and underscores. This can also be specified by a grammar rule or rules. What would the rule(s) look like for identifiers?

9. If we specify keywords, identifiers (see above) and programs by grammar rules, are there any problems you need to be careful about when using a parser for identifying tokens?

10. **(Optional)** Recall the definitions for *Der* and *der* from the lectures. Prove by induction on $r$ the property that

$$L(der\, c\, r) = Der\, c\, (L(r))$$

holds.

11. **(Optional)** This question is for you to provide regular feedback to me: for example what were the most interesting, least interesting, or confusing parts in this lecture? Any problems with my Scala code? Please feel free to share any other questions or concerns. Also, all my material is ~~crap~~ imperfect. If you have any suggestions for improvement, I am very grateful to hear.

    If *you* want to share anything (code, videos, links), you are encouraged to do so. Just drop me an email or send a message to the Forum.