SAARLAND UNIVERSITY
COMPUTER SCIENCE

Prof. Dr. Reinhard Wilhelm
Jun.-Prof. Dr. Sebastian Hack
Jörg Herter, M.Sc.
Dipl.-Inform. Christoph Mallon

Compiler Construction WS09/10

Exercise Sheet 1

Please hand in the solutions to the theoretical exercises until the beginning of the lecture next Wednesday 2009-10-28, 10:00. Please write the number of your tutorial group or the name of your tutor on the first sheet of your solution.

Send your solution to the practical exercise to jherter@cs.uni-sb.de until 2009-10-28, 10:00. Please send just one e-mail per project group. Solutions submitted later will not be accepted.

Exercise 1.1: Regular Expressions and Finite Automata (Points: 3+3+3+2+2+1)

1. For the regular expression $(a|b)^*bab(a|b)^*$ over the alphabet $\Sigma = \{a, b\}$ construct:
   a) a nondeterministic finite automaton,
   b) a deterministic finite automaton, and
   c) a minimal deterministic finite automaton

   following the proof sketches from the lecture.

2. At first sight, the construction of an NFA from a regular expression as presented in the lecture seems to be too complicated in the case of r^* (Slide 20). Give examples where simplifying the rule by merging states
   a) q and q_1 fails.
   b) q_2 and p fails.
   c) q_1 and q_2 fails.

Exercise 1.2: Greed (Points: 3+3)

1. Provide an automaton and a corresponding input word, such that the maximal munch strategy has runtime in $\Omega(n^2)$. Prove all your claims!

2. Is there a sequence of regular expressions $\alpha_1, \ldots, \alpha_n$ over Σ and a word $w \in \Sigma^*$, such that
   - there exist $w_1, \ldots, w_k \in \Sigma^*$ and $t_1, \ldots, t_k \in \{1, \ldots, n\}$, such that $w = w_1 \ldots w_k$ and $w_i \in \alpha_{t_i}$ (that is, there is a match of the complete word), and
   - for each such dissection there exist $j \in \{1, \ldots, k-1\}$, $x \in \Sigma^+$, $y \in \Sigma^*$, and $s \in \{1, \ldots, n\}$, such that $w_{j+1} \ldots w_k = xy$, $w_j x \in \alpha_s$, and there is no dissection $y = w'_1 \ldots w'_m$ and $t'_1, \ldots, t'_m \in \{1, \ldots, n\}$, such that $w'_i \in \alpha_{t'_i}$ (that is, greedy scanning does not take us to a match)?

Exercise 1.3: Minimizing Automata (Points: 5)

Consider the following alternative algorithm for minimizing automata: Let $\langle \Sigma, Q, \Delta, q_0, F \rangle$ be a DFA. We merge two states $q_1, q_2 \in Q$ iff $q_1 \in F \Leftrightarrow q_2 \in F$ and for all $a \in \Sigma$ holds: $(q_1, a, p) \in \Delta \Leftrightarrow (q_2, a, p) \in \Delta$. We apply this rule until no further states can be merged.

Does this algorithm correctly minimize arbitrary deterministic finite automata? Find a proof or a counter example to back up your claim!

Exercise 1.4: Lexer (Project)

In the practical project, you are going to implement your own MiniJava compiler. You find a language specification for the MiniJava language on the compiler construction web page.

Use this specification to identify the symbols/tokens of the MiniJava language. Then implement a Lexer (lexical analysis) to read MiniJava source files and generate token streams for these files. Remember that whitespace and comments are not part of the token stream.

For example, the MiniJava line `int i = 42;` should produce the following token stream:

```
int
IDENT i
=
INTEGER_LITERAL 42
;
```

Your Lexer should be called via `mjava Source.java`.
Please use C/C++ to implement your Lexer and hand in your solution until 2009-10-28, 10.00!