Technische Universität München *Summer Semester 08*
Fakultät für Informatik *Vesal Vojdani*
Prof. Dr. H. Seidl *vojdanig@in.tum.de*

Compiler Construction

Exercise Sheet 4

Deadline: 14. May 2008, at the lecture, in room 02.07.053, or by e-mail.

Exercise 1: *Regular expressions* *8 Points*

Consider binary strings over the alphabet $\Sigma = \{0, 1\}$.

a) What language is denoted by the regular expression
$(00|11)^*((01|10)(00|11)^*(01|10)(00|11)^*)^*$

b) Give a regular expression matching all strings that do not contain the substring 100.

c) Give a regular expression for strings that do not contain the subsequence 100.

d) Give a regular expression that recognizes bit strings which interpreted as binary numbers are divisible by 3. (Difficult!)

The difference here between a *substring* (11100111) and *subsequence* (11010111110) is that a substring is a consecutive subsequence.

Exercise 2: *Identifiers and Constants* *4 Points*

Pick three programming languages you know well, which are fairly different from each other, e.g., Java, Haskell, and SQL.

a) For one of the languages, present the exact lexical forms for identifiers and numerical constants according to the language manuals/specification.

b) Give an example of a lexeme that is a valid identifier in one language, but not in one of the others.

Exercise 3: *Transforming regular expressions* *8 Points*

a) Rewrite the regular expression $(a \mid cb?)^*d$ without using the operators *, ? or ε. Instead, you may use $^+$ and the other regular expression operator.

b) Give a transformation T that performs systematically such a rewrite for any regular expression e, whose language does not contain the empty symbol, i.e., $\epsilon \notin L(e)$.

Exercise 4: *Finite Automata* *10 Points*

Make up your own original and reasonably interesting regular expression over the alphabet $\Sigma = \{a, b\}$. By reasonably interesting I mean something for which your NFA has about 4-8 states.

a) Construct the NFA to recognize the language of your regular expression.

b) Give the corresponding DFA.