Software-Modellierung und Verifikation
Informatik 2
Prof. J.-P. Katoen
RWTH Aachen

Priv.-Doz. T. Noll noll@cs.rwth-aachen.de
D. Klink klink@cs.rwth-aachen.de

2. Exercise sheet *Compiler Construction 2008*

Due to Wed., 7 May 2008, *before* the exercise course begins.

Hand in your solutions in groups of three!

Exercise 2.1:

a) For extended matching two principles have been introduced to resolve nondeterminism during analysis, the *longest match* principle and the *first match* principle. Instead, we could have insisted on an unambiguous definition of the symbol classes, i.e. for regular expressions $\alpha_1, \ldots, \alpha_n$ it should hold $\bigcap_{i=1}^{n} [\![\alpha_i]\!] = \emptyset$. Why is this not a good idea from a practical point of view? Give an example to support your explainations.

b) Let $\alpha_1, \ldots, \alpha_n$ be regular expressions over Σ and $w \in \Sigma^*$. In the lecture it was assumed that $\varepsilon \notin [\![\alpha_i]\!] \neq \emptyset$ for all $i \in \{1, \ldots, n\}$. Show that these are reasonable assumptions by proving the following proposition:

- If $[\![\alpha_i]\!] = \emptyset$ for some $i \in \{1, \ldots, n\}$ there exists no *flm*-analysis of w w.r.t. $\alpha_1, \ldots, \alpha_n$ that is not a *flm*-analysis of w w.r.t. $\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n$ as well.
- If $\varepsilon \in [\![\alpha_i]\!]$ for some $i \in \{1, \ldots, n\}$ then the *flm*-analysis of w w.r.t. $\alpha_1, \ldots, \alpha_n$ is not unique (if it exists).

Exercise 2.2:

Let $\alpha_1 = (\ominus + \varepsilon)N$, $\alpha_2 = N(e \ominus N + \varepsilon)$ where N is a macro for $(1+\ldots+9)(0+\ldots+9)^*$ and $\alpha_3 = (a+\ldots+z)(a+\ldots+z)^*$ be regular expressions over the alphabet $\Sigma = \{\ominus, 0, \ldots, 9, a, \ldots, z\}$.

a) Construct DFAs $\mathfrak{A}_i$ for α_i such that $\mathcal{L}(\mathfrak{A}_i) = [\![\alpha_i]\!]$.

b) Construct DFA $\mathfrak{A}$ such that $\mathcal{L}(\mathfrak{A}) = \mathcal{L}(\mathfrak{A}_1) \cup \mathcal{L}(\mathfrak{A}_2) \cup \mathcal{L}(\mathfrak{A}_3)$.

c) Determine the *first match* partitioning of the set of final states in $\mathfrak{A}$.
(The regular expressions are ordered $(\alpha_1, \alpha_2, \alpha_3)$.)

d) Compute the (accepting) run of the corresponding backtracking DFA for input $w = 23exp\ominus 42$.

Exercise 2.3: (optional)

Download *ANTLRWorks* from `www.antlr.org/works`.

a) Enter the following grammar file and save it as `Rational.g`:

```
grammar Rational;
start           :   (RATIONAL ' ')* RATIONAL;
RATIONAL        :   a regular expression;
```

Replace *a regular expression* by the regular expression from Exercise 0.1a (in *ANTLRWorks* notation).
Select the *Interpreter* tab, enter `-1/2 0/4 +5` as input and press the start button.

b) Generate code (see menu) and write a java application that takes the first command line argument as input lexeme and prints the token generated by the generated `RationalLexer` to the console.