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

Priv.-Doz. T. Noll noll@cs.rwth-aachen.de
Ch. Jansen christina.jansen@cs.rwth-aachen.de

4. Exercise sheet *Compiler Construction 2010*

Due Wed., 17 November 2010, *before* the exercise course begins.

Exercise 4.1: **(3 points)**

To conclude the lecture and exercise part about lexical analysis, answer the following questions:

(a) Briefly sketch the structure of a compiler! Describe the functionality of each component. Where does the lexical analysis fit in here?

(b) Describe the meaning and differences between lexemes, symbols, symbol classes and tokens!

(c) Which problem does the simple matching problem solve?

(d) What is the output of the FLM approach handling the extended matching problem as presented in the lecture? Is the result unique? If yes, why?

(e) Where and why do we need to partition the final states? How does the partitioning work, argue why this partitioning is suitable!

Exercise 4.2: **(3 points)**

Consider the following grammar G:

$$\begin{array}{lcl} S & \rightarrow & aBSS \mid bBS \mid cSB \mid dLe \mid f \\ L & \rightarrow & SaL \mid \epsilon \\ B & \rightarrow & aC \mid bC \\ C & \rightarrow & eB \mid \epsilon \end{array}$$

Give the corrresponding NTA $\mathfrak{A}$ of G. Provide the complete NTA run of $\mathfrak{A}$ for the input $dbaabffae$.

Exercise 4.3: **(1+2 points)**

a) Show that the following grammar is ambiguous:

$$S \rightarrow (S) \mid S \vee S \mid S \wedge S \mid \neg S \mid \text{true}$$

b) Devise a method that chooses a reasonable derivation out of the set of all leftmost derivations for a given word $w \in \Sigma^*$ by investigating the leftmost analysis information. Reasonable in this context means that

$\neg$ binds stronger than $\vee$ and $\wedge$,

$\wedge$ binds stronger than $\vee$.

Exercise 4.4: **(1+3+1 points)**

In the lecture two characterizations of $LL(1)$ have been given:

- $G \in LL(1)$ iff for all leftmost derivations of the form

$$S \Rightarrow_l^* wA\alpha \begin{cases} \Rightarrow_l w\beta\alpha \\ \Rightarrow_l w\gamma\alpha \end{cases}$$

such that $\beta \neq \gamma$, it follows that $\mathrm{fi}(\beta\alpha) \cap \mathrm{fi}(\gamma\alpha) = \emptyset$.

- $G \in LL(1)$ iff for all pairs of rules $A \to \beta \mid \gamma \in P$ (where $\beta \neq \gamma$):

$$\mathrm{la}(A \to \beta) \cap \mathrm{la}(A \to \gamma) = \emptyset$$

a) Lift the second definition to $LL(k)$ for $k \in \mathbb{N}^+$. (The first definition was given for $k \in \mathbb{N}^+$ in the lecture.)

b) Show that the definitions are not equivalent by showing that the following grammar is in $LL(2)$ according to the first definition but not according to the second definition (also refered to as *strong* $LL(2)$ property).

$$\begin{aligned} S &\to aAab \mid bAbb \\ A &\to a \mid \varepsilon \end{aligned}$$

c) Explain (in a few words) why the definitions are not equivalent.