

Compiler Construction

Exercise Sheet 7

Deadline: 11. June 2008, at the lecture, in room 02.07.053, or by e-mail.

Exercise 1: FIRST and FOLLOW sets

5 Points

Consider the grammar:

$$\begin{aligned} E &\rightarrow T A \\ A &\rightarrow + T A \mid - T A \mid \epsilon \\ T &\rightarrow F B \\ B &\rightarrow * F B \mid / F B \mid \epsilon \\ F &\rightarrow - S \mid S \\ S &\rightarrow v \mid (E) \end{aligned}$$

- Set up a constraint system to compute the $FIRST_1$ and $FOLLOW_1$ sets.
- Solve the system using fix point iteration.

Exercise 2: LL(1)

6 Points

Consider the following grammars:

$$\begin{array}{lll} S \rightarrow A a \mid B & S \rightarrow A a \mid b & S \rightarrow a b A \mid b \\ A \rightarrow a \mid \epsilon & A \rightarrow c \mid \epsilon & A \rightarrow b S b \mid \epsilon \\ B \rightarrow b & & \end{array}$$

For each grammar, find out whether it is an LL(1) grammar. Justify your answer.

Exercise 3: LL(2)

5 Points

Show that the following grammar is LL(2), but not strong LL(2).

$$\begin{aligned} S &\rightarrow a A a \mid b B a \\ A &\rightarrow C a \\ B &\rightarrow C c \\ C &\rightarrow c \mid \epsilon \end{aligned}$$

Exercise 4: From LL(k) to strong LL(k)

8 Points

For every LL(k) grammar there exists an equivalent strong LL(k) grammar. Give an algorithm that performs this transformation and apply it to the example of the previous exercise.

Exercise 5: Not LL(k)

6 Points

Give an example of a grammar that is not LL(k). Prove that it is not LL(k) for any $k > 0$.