

Exercise 1 (LR(1)):

(3 Points)

Let grammar G be given by:

$$\begin{aligned} S' &\rightarrow S \\ S &\rightarrow Aa \mid bAc \mid Bc \mid bBa \\ A &\rightarrow d \\ B &\rightarrow d \end{aligned}$$

- a) Check whether $G \in LR(1)$ by computing the $LR(1)$ -sets of G .
- b) Is $G \in LALR(1)$? Justify your answer.

Exercise 2 (LL vs. LALR):

(3 Points)

Show that $LL(1) \not\subseteq LALR(1)$.

Exercise 3 (Attribute Grammars):

(4 Points)

Consider the following grammar for a subset of WHILE:

```
start  → program
program → statement program | statement
statement → declaration SEM | assignment SEM | branch
declaration → INT ID
assignment → ID ASSIGN expr
branch → IF LBRAC guard RBRAC LCBRAC program RCBRAC |
        IF LBRAC guard RBRAC LCBRAC program RCBRAC ELSE LCBRAC program RCBRAC
expr → NUM | ID | LBRAC subexpr RBRAC
subexpr → expr PLUS expr | expr MINUS expr | expr TIMES expr | expr DIV expr
guard → relation | LBRAC subguard RBRAC | NOT LBRAC guard RBRAC
subguard → guard AND guard | guard OR guard
relation → expr LT expr | expr LEQ expr | expr EQ expr | expr NEQ expr | expr GEQ expr | expr GT expr
```

Provide grammar attributes, which generate a symbol table holding every declared variable. Test whether variables are declared multiple times! The symbol table should contain at least the address of the variable. For simplicity you may assume, that every variable needs one storing unit, i.e. n variables obtain addresses 1 to n .

Your attributed grammar should test whether all variables occurring in a program were declared before.