

4 Compiler Construction (tgg22)

- (a) Show that the following grammar for the language of balanced parenthesis is ambiguous.

$$\begin{array}{l}
 S \rightarrow \epsilon \\
 \quad | (S) \\
 \quad | SS
 \end{array}$$

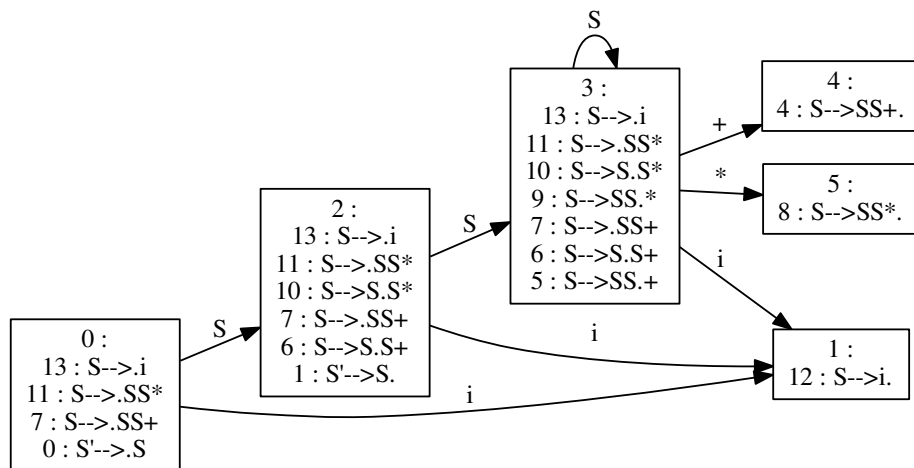
[5 marks]

- (b) An arithmetic expression such as $x * ((z * u) + y)$ can be represented without parenthesis in postfix notation as $xzu * y + *$. This representation is ideal for evaluation using a stack machine.

Below is a simple grammar for postfix expressions:

$$\begin{array}{l}
 S \rightarrow i \\
 \quad | SS+ \\
 \quad | SS*
 \end{array}$$

The terminal i represents the lexical class of identifiers. Here is a DFA for the LR(0) items of this grammar.



- (i) Using the DFA, construct the SLR(1) *ACTION* and *GOTO* tables for this grammar. Explain your work. [6 marks]

- (ii) Show a trace of a parsing of $w = iii * i + *$. Justify every step. [9 marks]