

# Task 4: Theory

## a) Formally describe the regular grammar of the lexemes in task 2

If we have a grammar  $G = (N, E, P, S)$ , where:

$N$  = Non-Terminals

$E$  = Terminals

$P$  = Production Rules

$S$  = a start symbol  $S \in N$

We can describe each lexeme as:

$G\{\text{Integer}\} = (\{S.A\}, \{1,2,3,4,5,6,7,8,9,0\}, \{S \rightarrow 1A,$

$S \rightarrow 2A,$

$S \rightarrow 3A,$

$S \rightarrow 4A,$

$S \rightarrow 5A,$

$S \rightarrow 6A,$

$S \rightarrow 7A,$

$S \rightarrow 8A,$

$S \rightarrow 9A,$

$S \rightarrow 0A,$

$A \rightarrow 1A,$

$A \rightarrow 2A,$

$A \rightarrow 3A,$

$A \rightarrow 4A,$

$A \rightarrow 5A,$

$A \rightarrow 6A,$

$A \rightarrow 7A,$

$A \rightarrow 8A,$

$A \rightarrow 9A,$

$A \rightarrow 0A,$

$A \rightarrow \text{epsilon},$

$\}, S)$

$G_{\{\text{operator}\}} = (\{S\}, \{+, -, *, /, p, d, i\}, \{$

$S \rightarrow +,$

$S \rightarrow -,$

$S \rightarrow *,$

$S \rightarrow /,$

$S \rightarrow p,$

$S \rightarrow d,$

$S \rightarrow i,$

$\}, S)$

Here we are not describing the relation between the tokens, rather just the grammar of the lexemes themselves (Not the whole string fed into *fun {lex}*).

**b) Describe the grammar of the infix notation in task 3 using (E)BNF. Beware of operator precedence. Is the grammar ambiguous? Explain why it is or is not ambiguous?**

One example:

$\langle \text{exp} \rangle ::= \langle \text{int} \rangle \mid \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$

$\langle \text{op} \rangle ::= + \mid - \mid * \mid /$

This is ambiguous, because an expression on the form  $4*5-3$  could be written as either  $(4*5)-3$  or  $4*(5-3)$ , which will give two different results.

The book suggests to put the higher precedence operators deeper in the parser tree (farthest away from the root).

For example:

$\langle \text{exp} \rangle ::= \langle \text{term} \rangle + \langle \text{exp} \rangle \mid \langle \text{term} \rangle - \langle \text{exp} \rangle \mid \langle \text{term} \rangle$

$\langle \text{term} \rangle ::= \langle \text{int} \rangle * \langle \text{term} \rangle \mid \langle \text{int} \rangle / \langle \text{term} \rangle \mid \langle \text{int} \rangle$

**c) What is the difference between a context-sensitive and a context-free grammar?**

Context-sensitive grammar is a formal grammar in which left-hand sides and right hand sides of any production (rewrite) rules may be surrounded by a context of terminal and nonterminal symbols.

Context free grammar does not have this property.

d)

This is because Oz is strongly typed. This is useful to prevent unexpected behaviour and bugs