© Manuel Cargaleiro

Exercises About LL(K) Grammars

Compilers course

Masters in Informatics and Computing Engineering (MIEIC), 3rd Year

João M. P. Cardoso

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Dep. de Engenharia Informática
Faculdade de Engenharia (FEUP)
Universidade do Porto,
Porto, Portugal
Email:jmpc@acm.org

Exercise 1

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid NUM$

- **Group 1. *First*, *Follow* Sets and *Parsers* LL(k) (5 pts)**
- Consider the CFG1 in the left. **NUM** is a terminal symbol representing numbers.
- **1.a)** [1pt] Give the *First* and *Follow* sets for each grammar variable;
- **1.b)** [2pts] Is the grammar LL(1)? Show the table for the *parser* LL(1);
- **1.c)** [2pts] Modify the grammar in order it can be implemented as a top-down recursive parser with K=1 (*lookahead*).

Source: First midterm exam (MT1): April 6, 2017

Exercise 2 (cont.)

- **1d)** Show the grammar is LL(1) by using the table for the *parser* LL(1)
- **1e)** Show the concrete syntax tree (CST) for 2+3*4+5
- **1f)** Show a possible abstract syntax tree (AST) for 2+3*4+5

$E \rightarrow T\ E1$

$E1 \rightarrow +\ E \mid \varepsilon$

$T \rightarrow F\ T1$

$T1 \rightarrow *\ T \mid \varepsilon$

$F \rightarrow (\ E\ ) \mid NUM$

Source: First midterm exam (MT1): April 6, 2017