Programmation Avancée et Compilation 2001
(Exercises)

5 Top-Down Parsing

Keywords 5

Give short explanations of the following keywords:

> *Recursive descent parsing (predictive parsing)* (Parsing 4), *LL(1)* (Parsing 13).

Exercise 9

In order to compute the first, follows, and nullable sets of a grammar (to determine whether it is LL(1) or, later, to present it to a parser generator), we need a representation in BNF. For this purpose, the lecture provides a transformation function translating EBNF expressions into equivalent BNF representations.

Translate the following EBNF grammars into BNF expressions.

(a) A ::= "a" { "b" } ("a" | A)

(b) B ::= { "(" [B | D] ")" }   D ::= { "[" [B | D] "]" }

(c) C ::= { ("a" | ["b"]) ("c" | "d") }

Exercise 10

Consider the following grammar in BNF:

A	::=	"a" A | "a" B "b"
B	::=	"b" B | C
C	::=	ϵ | B

(a) Which of the nonterminals are nullable?

(b) Determine the first and follows sets for A and C.

Exercise 11 (Programming exercise)

Recall the grammar recognizing arithmetic expressions from Exercise 3. At http://lampwww.epfl.ch/courses/progavancee01/HandParser.java, you find the frame of a a handwritten top-down parser checking inputs for correctness, where the parse functions are still missing.

(a) Complete the parser.

(b) Check your parser on a sample of correct and incorrect expressions.