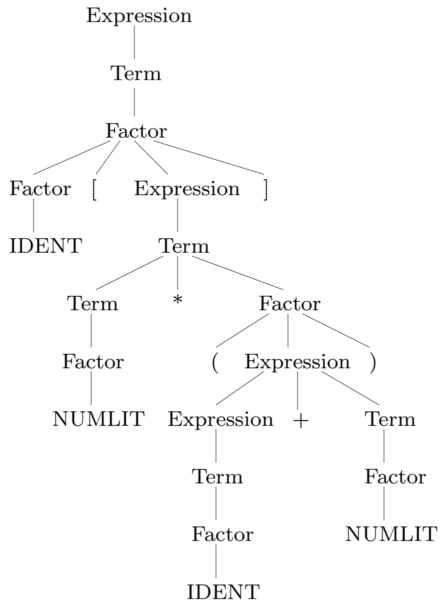
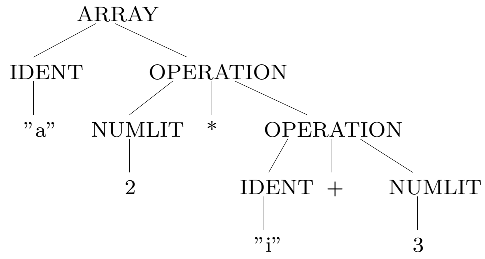
Programmation Avancée et Compilation 2001
(Solutions)

7 Abstract Syntax

Exercise 13

(a) The concrete and abstract syntax-trees are as follows:





(b) As is obvious from the example in part (a) of the exercise, abstract syntax trees are generally much simpler than concrete syntax trees, omitting superfluous information (grouping by parentheses is already expressed in terms of the tree structure).

(c) This is not a question of compiler construction but rather of language design. There exist languages based on a tree-like syntax (LISP, Scheme), but usually such a syntax is considered hard to read and maintain.

Exercise 14

(a) The grammar exhibits two typical problems: the first production for Content is left-recursive, and the two productions for List start with the same symbol.

(b) In general, a left-recursive pair of productions $\mathsf{A} ::= \mathsf{A}\ \alpha \mid \beta$ can be replaced by $\mathsf{A} ::= \beta\ \{\ \alpha\ \}$. In our case, this yields:

$$\mathsf{Content} \quad ::= \quad \mathsf{IDENT}\ \{\ \text{","}\ \mathsf{IDENT}\ \ \}$$

The second problem can be eliminated by left-factoring:

$$\mathsf{List} \quad ::= \quad \text{"["}\ \ [\ \mathsf{Content}\ ]\ \text{"]"}$$