Programmation Avancée et Compilation 2001
(Solutions)

5 Top-Down Parsing

Exercise 9

The translation scheme consists in eliminating bracketing operators and internal choices by introducing new non-terminals. For instance, { E } is converted into a fresh nonterminal X, and the production X ::= ϵ | E'X is added, where E' is a transformation of E. (Try to recall the rest of the transformation scheme.)

(a)

	A	::=	"a" A1 A2	
	A1	::=	ϵ | "b" A1	describing { "b" }
	A2	::=	"a" | A	describing ("a" | A)

Alternatively, we could (and would have for a parser-generator, to keep the grammar small) also have written,

A	::=	"a" A1 "a" | "a" A1 A
A1	::=	ϵ | "b" A1

(b)

B	::=	B1	for { ... }
B1	::=	ϵ | "(" ")" B1 | "(" B2 ")" B1	for "(" [B | D] ")"
B2	::=	B | D	for (...)
D	::=	D1	for { ... }
D1	::=	ϵ | "[" "]" D1 | "[" D2 "]" D1	for "[" [B | D] "]"
D2	::=	B | D	for (...)

(c)

C	::=	C1	for { ... }
C1	::=	ϵ | C2 C3 C1	for (...)(...)
C2	::=	"a" | ϵ | "b"	for "a" | ["b"]
C3	::=	"c" | "d"	for "c" | "d"

Exercise 10

(a) The non-terminals B and C can both produce ϵ, that is, they are nullable (B can produce the empty word via C). The non-terminal A is certainly not nullable, with each of its right-hand side clauses containing at least one terminal.

(b) The first and follows sets are obtained by a fixpoint construction: starting with empty sets for each of the terminals, these are augmented according to certain rules until they do not change any more (that is, until a fixpoint is reached). Intuitively, the rules express the following facts:

- If from a nonterminal X, a sequence starting with some terminal t can be derived, this terminal should certainly belong to first(X). This can have two reasons (this is a bit formal, nothing to be reproduced):
  1. The rule for X is of the form X ::= ... | "t"... | ...; or
  2. the rule for X is of the form X ::= ... | Y1...Yk... | ..., all of the Y1...Yk can produce ϵ (that is, they are nullable!), and what follows next can produce something starting with t. That is, what follows is either "t" or some nonterminal Y with t $\in$ first(Y).

– The set follows(X) expresses that X can be followed by t in a derived sequence. This can have three reasons (again, this is a bit formal, nothing to be reproduced):

  1. There is some rule of the form Y ::= ... | ...X "t"... | ...; or
  2. there is some rule of the form Y ::= ... | ...X Y1...Yk... | ..., where all of the Y1...Yk can produce ϵ (that is, they are nullable!), and what follows can derive a sequence starting with t. That is, what follows is either "t" or some nonterminal Z with t $\in$ first(Z); or
  3. there is some rule of the form Y ::= ... | ...X Y1...Yk | ..., where all of the Y1...Yk are nullable, in which case X can be followed by all terminals in follows(Y). To illustrate this case, consider a derivation yielding ...Y t.... Applying the above production, we obtain ...X Y1...Yk t..., from which we can derive ...X t... by applying that all Y1...Yk are nullable.

Let us first compute the first sets for the non-terminals:

– Starting with empty sets, we obtain in a first step that a must be in first(A), and b in first(B).

– In a second step, we can derive from C ::= ... | B that b must be in first(C) as well.

– In a third step, we could for instance augment first(B) due to B ::= ... | C. Or else we could repeat the second step. But this does not change the sets any more, hence we have reached a fixpoint and can terminate.

We hence have first(A) = { a }, and first(C) = first(B) = { b }.

Let us now have a look at the follows sets:

– Starting with empty sets, we obtain in a first step that b must be in follows(B).

– In a second step, we can derive from B ::= ... | C that b must be in follows(C) as well.

– In a third step, we could for instance augment follows(B) due to C ::= ... | B. Or else we could repeat the second step. But this does not change the sets any more, hence we have reached a fixpoint and can terminate.

With none of the rules applying to A, we hence have follows(A) = $\emptyset$, and follows(C) = follows(B) = { b }.

Hint: In order to get a feeling why follows(A) = $\emptyset$, perform a derivation, for example of aab or aaabb. Together with the above-mentioned intuitions (first(X) = what terminals a sequence derived from X can start with; follows(X) = what terminals can follow X is a sequence derived from some Y), one can obtain a good feeling about all of these sets by executing some (beginnings of) derivations.