1) Erstellen einer LALR(1)-Tabelle

Zustände

0)

S' = . Expr #		
Expr = .Term	/ #, "+"	shift Expr 1
Expr = . Expr "+" Term	/ #, "+"	shift Term 2
Term = . Factor	/ #, "+", "*"	shift Factor 3
Term = . Term "*" Factor	/ #, "+", "*"	shift id 4
Factor = . id	/ #, "+", "*"	shift num 5
Factor = . num	/ #, "+", "*"	
Factor = . id "(" Expr ")"	/ #, "+", "*"	

1)

S' = Expr . #		accept #
Expr = Expr . "+" Term	/ #, "+"	shift "+" 6

2)

Expr = Term.	/ #, ")", "+"	reduce #, ")", "+" (1)
Term = Term . "*" Factor	/ #, ")", "+", "*"	shift "*" 7

3)

Term = Factor .	/ #, ")", "+", "*"	reduce #, ")", "+", "*" (3)

4)

Factor = id .	/ #, ")", "+", "*"	reduce #, ")", "+", "*" (5)
Factor = id . “(“ Expr “)”	/ #, “)”, “+”, “*”	shift “(“, 8

5)

Factor = num .	/ #, ")", "+", "*"	reduce #, ")", "+", "*" (6)

6)

Expr = Expr "+" . Term	/ #, ")", "+"	shift Term 9
Term = . Factor	/ #, ")", "+", "*"	shift Factor 3
Term = . Term "*" Factor	/ #, ")", "+", "*"	shift id 4
Factor = . id	/ #, ")", "+", "*"	shift num 5
Factor = . num	/ #,")", "+", "*"	
Factor = . id "(" Expr ")"	/ #, ")", "+", "*"	

7)

Term = Term "*" . Factor	/ #, ")", "+", "*"	shift Factor 10
Factor = . id	/ #, ")", "+", "*"	shift id 4
Factor = . num	/ #, ")", "+", "*"	shift num 5
Factor = . id "(" Expr ")"	/ #, ")", "+", "*"	

8)

Factor = id "(". Expr ")"	/ #, ")", "+", "*"	shift Expr 11
Expr = . Term	/ ")", "+"	shift Term 2
Expr = . Expr "+" Term	/ ")", "+"	shift Factor 3
Term = . Factor	/ ")", "+", "*"	shift id 4
Term = . Term "*" Factor	/ ")", "+", "*"	shift num 5
Factor = . id	/ ")", "+", "*"	
Factor = . num	/ ")", "+", "*"	
Factor = . id "(" Expr ")"	/ ")", "+", "*"	

9)

Expr = Expr "+" Term .	/ #, ")", "+"	reduce #, ")", "+" (2)
Term = Term . "*" Factor	/ #, ")", "+", "*"	shift "*" 7

10)

Term = Term "*" Factor .	/ #, ")", "+", "*"	reduce #, ")", "+", "*" (4)

11)

Factor = id "(" Expr . ")"	/ #, ")", "+", "*"	shift ")" 12
Expr = Expr . "+" Term	/ ")", "+"	shift "+" 6

12)

Factor = "(" Expr ")" .	/ #, ")", "+", "*"	reduce #, ")", "+", "*" (7)

Gegebene EBNF-Grammatik

Expr = Term { "+" Term } .
Term = Factor { "*" Factor } .
Factor = id | num | "(" Expr ")" .

Transformation in BNF-Grammatik

(0) S' = Expr # .
(1) Expr = Term .
(2) Expr = Expr "+" Term .
(3) Term = Factor .
(4) Term = Term "*" Factor .
(5) Factor = id .
(6) Factor = num .
(7) Factor = id "(" Expr ")" .

Analysetabelle

	"+"	"*"	"("	")"	id	num	#	Expr	Term	Factor
0	-	-	-	-	s4	s5	-	s1	s2	s3
1	s6	-	-	-	-	-	acc	-	-	-
2	red(1)	s7	-	red(1)	-	-	red(1)	-	-	-
3	red(3)	red(3)	-	red(3)	-	-	red(3)	-	-	-
4	red(5)	red(5)	s8	red(5)	-	-	red(5)	-	-	-
5	red(6)	red(6)	-	red(6)	-	-	red(6)	-	-	-
6	-	-		-	s4	s5	-	-	s9	s3
7	-	-	-	-	s4	s5	-	-	-	s10
8	-	-	-	-	s4	s5	-	s11	s2	s3
9	red(2)	s7	-	red(2)	-	-	red(2)	-	-	-
10	red(4)	red(4)	-	red(4)	-	-	red(4)	-	-	-
11	s6	-	-	s12	-	-	-	-	-	-
12	red(7)	red(7)	-	red(7)	-	-	red(7)	-	-	-

2) Tabellenverkleinerung

In dieser Tabelle sind die Zustände markiert, die nur reduzieren und das nur nach einer Regel. (Das ist nicht verlangt in der Übung.)

	"+"	"*"	"("	")"	id	num	#	Expr	Term	Factor
0	-	-	-	-	s4	s5	-	s1	s2	s3
1	s6	-	-	-	-	-	acc	-	-	-
2	red(1)	s7	-	red(1)	-	-	red(1)	-	-	-
3	red(3)	red(3)	-	red(3)	-	-	red(3)	-	-	-
4	red(5)	red(5)	s8	red(5)	-	-	red(5)	-	-	-
5	red(6)	red(6)	-	red(6)	-	-	red(6)	-	-	-
6	-	-		-	s4	s5	-	-	s9	s3
7	-	-	-	-	s4	s5	-	-	-	s10
8	-	-	-	-	s4	s5	-	s11	s2	s3
9	red(2)	s7	-	red(2)	-	-	red(2)	-	-	-
10	red(4)	red(4)	-	red(4)	-	-	red(4)	-	-	-
11	s6	-	-	s12	-	-	-	-	-	-
12	red(7)	red(7)	-	red(7)	-	-	red(7)	-	-	-

Shift-Reduce-Verschmelzung

	"+"	"*"	"("	")"	id	num	#	Expr	Term	Factor
0	-	-	-	-	s4	sr(6)	-	s1	s2	sr(3)
1	s6	-	-	-	-	-	acc	-	-	-
2	red(1)	s7	-	red(1)	-	-	red(1)	-	-	-
4	red(5)	red(5)	s8	red(5)	-	-	red(5)	-	-	-
6	-	-		-	s4	sr(6)	-	-	s9	sr(3)
7	-	-	-	-	s4	sr(6)	-	-	-	sr(4)
8	-	-	-	-	s4	sr(6)	-	s11	s2	sr(3)
9	red(2)	s7	-	red(2)	-	-	red(2)	-	-	-
11	s6	-	-	sr(7)	-	-	-	-	-	-

Shift-Reduce-Verschmelzung mit Neu-Nummerierung

	"+"	"*"	"("	")"	id	num	#	Expr	Term	Factor
0	-	-	-	-	s3	sr(6)	-	s1	s2	sr(3)
1	s4	-	-	-	-	-	acc	-	-	-
2	red(1)	s5	-	red(1)	-	-	red(1)	-	-	-
3 (4)	red(5)	red(5)	s6	red(5)	-	-	red(5)	-	-	-
4 (6)	-	-		-	s3	sr(6)	-	-	s7	sr(3)
5 (7)	-	-	-	-	s3	sr(6)	-	-	-	sr(4)
6 (8)	-	-	-	-	s3	sr(6)	-	s8	s2	sr(3)
7 (9)	red(2)	s5	-	red(2)	-	-	red(2)	-	-	-
8 (11)	s4	-	-	sr(7)	-	-	-	-	-	-

3) Simulation einer LALR(1)-Analyse

	a	b	c	d	e	#	S	A	B
0	s2	-	r(4)	-	s4	r(4)	s1	-	s3
1	-	-	s6	-	-	acc	-	s5	-
2	-	-	s6	-	-	-	-	s7	-
3	-	-	r(2)	-	s8	r(2)	-	-	-
4	r(7)	r(7)	r(7)	r(7)	r(7)	r(7)	-	-	-
5	s10	s9	-	-	-	-	-	-	-
6	-	-	-	-	s4	-	-	-	s11
7	s10	-	r(1)	-	-	r(1)	-	-	-
8	r(8)	r(8)	r(8)	r(8)	r(8)	r(8)	-	-	-
9	-	-	r(3)	-	-	r(3)	-	-	-
10	r(6)	r(6)	r(6)	-	-	r(6)	-	-	-
11	-	-	-	s12	s8	-	-	-	-
12	-	-	-	-	s4	-	-	-	s13
13	r(5)	r(5)	r(5)	-	s8	r(5)	-	-	-

(1) S = a A.
(2) S = B.
(3) S = S A b.
(4) S = .
(5) A = c B d B.
(6) A = A a.
(7) B = e.
(8) B = B e.

Erkennung von ε (leerer Satz)

Keller	Eingabe	Aktion	
0	#	r(4)	Regel (4), "S=." hat Länge 0 → Keller unverändert
0	S#	s1	
01	#	acc	

Erkennung von e c e e e d e e a b

Keller		Eingabe	Aktion	
0		e c e e e d e e a b #	s4	
0,4		c e e e d e e a b #	r(7)	B=e. Länge 1 → ein Element vom Keller
0	B	c e e e d e e a b #	s3	
0,3		c e e e d e e a b #	r(2)	S=B. Länge 1 → ein Element vom Keller
0	S	c e e e d e e a b #	s1	
0,1		c e e e d e e a b #	s6	
0,1,6		e e e d e e a b #	s4	
0,1,6,4		e e d e e a b #	r(7)	B=e. Länge 1 → ein Element vom Keller
0,1,6	B	e e d e e a b #	s11	
0,1,6,11		e e d e e a b #	s8	
0,1,6,11,8		e d e e a b #	r(8)	B=Be. Länge 2 → zwei Elemente vom Keller
0,1,6	B	e d e e a b #	s11	
0,1,6,11		e d e e a b #	s8	
0,1,6,11,8		d e e a b #	r(8)	B=Be. Länge 2 → zwei Elemente vom Keller
0,1,6	B	d e e a b #	s11	
0,1,6,11		d e e a b #	s12	
0,1,6,11,12		e e a b #	s4	
0,1,6,11,12,4		e a b #	r(7)	B=e. Länge 1 → ein Element vom Keller
0,1,6,11,12	B	e a b #	s13	
0,1,6,11,12,13		e a b #	s8	
0,1,6,11,12,13,8		a b #	r(8)	B=Be. Länge 2 → zwei Elemente vom Keller
0,1,6,11,12	B	a b #	s13	
0,1,6,11,12,13		a b #	r(5)	A=cBdB. Länge 4 → vier Elemente vom Keller
0,1	A	a b #	s5	
0,1,5		a b #	s10	
0,1,5,10		b #	r(6)	A=Aa. Länge 2 → zwei Elemente vom Keller
0,1	A	b #	s5	
0,1,5		b #	s9	
0,1,5,9		#	r(3)	S=SAb. Länge 3 → drei Elemente vom Keller
0	S	#	s1	
0,1		#	acc	

Erkennung von a c e d b

Keller		Eingabe	Aktion	
0		a c e d b #	s2	
0,2		c e d b #	s6	
0,2,6		e d b #	s4	
0,2,6,4		d b #	r(7)	B=e. Länge 1 → ein Element vom Keller
0,2,6	B	d b #	s11	
0,2,6,11		d b #	s12	
0,2,6,11,12		b #	err	

Punkteverteilung

- Transformation in BNF-Grammatik: 2 Punkte
- Ermitteln der Zustände: 10 Punkte
- Analysetabelle: 2 Punkte

- Tabellenverkleinerung: 4 Punkte

- Beispielsatz ε: 2 Punkte
- Beispielsatz a c e d b: 2 Punkte
- Beispielsatz e c e e e d e e a b: 2 Punkte