

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	/ #, "+", "**"	shift "(" 6	
Factor = . "(" Expr ")"	/ #, "+", "**"		
1)			
S' = Expr . #		accept #	
Expr = Expr . "+" Term	/ #, "+"	shift "+" 7	
2)			
Expr = Term .	/ #, ")", "+"	reduce #, ")", "+" (1)	
Term = Term . "*" Factor	/ #, ")", "+", "**"	shift "*" 8	
3)			
Term = Factor .	/ #, ")", "+", "**"	reduce #, ")", "+", "**" (3)	
4)			
Factor = id .	/ #, ")", "+", "**"	reduce #, ")", "+", "**" (5)	
5)			
Factor = num .	/ #, ")", "+", "**"	reduce #, ")", "+", "**" (6)	
6)			
Factor = "(" . Expr ")"	/ #, ")", "+", "**"	shift Expr 9	
Expr = . Term	/ ")", "+"	shift Term 2	
Expr = . Expr "+" Term	/ ")", "+"	shift Factor 3	
Term = . Factor	/ ")", "+", "**"	shift id 4	
Term = . Term "*" Factor	/ ")", "+", "**"	shift num 5	
Factor = . id	/ ")", "+", "**"	shift "(" 6	
Factor = . num	/ ")", "+", "**"		
Factor = . "(" Expr ")"	/ ")", "+", "**"		
7)			
Expr = Expr "+" . Term	/ #, ")", "+"	shift Term 10	
Term = . Factor	/ #, ")", "+", "**"	shift Factor 3	
Term = . Term "*" Factor	/ #, ")", "+", "**"	shift id 4	
Factor = . id	/ #, ")", "+", "**"	shift num 5	
Factor = . num	/ #, ")", "+", "**"	shift "(" 6	
Factor = . "(" Expr ")"	/ #, ")", "+", "**"		
8)			
Term = Term "*" . Factor	/ #, ")", "+", "**"	shift Factor 11	
Factor = . id	/ #, ")", "+", "**"	shift id 4	
Factor = . num	/ #, ")", "+", "**"	shift num 5	
Factor = . "(" Expr ")"	/ #, ")", "+", "**"	shift "(" 6	
9)			
Factor = "(" Expr . ")"	/ #, ")", "+", "**"	shift ")" 12	
Expr = Expr . "+" Term	/ ")", "+"	shift "+" 7	
10)			
Expr = Expr "+" Term .	/ #, ")", "+"	reduce #, ")", "+" (2)	
Term = Term . "*" Factor	/ #, ")", "+", "**"	shift "*" 8	
11)			
Term = Term "*" Factor .	/ #, ")", "+", "**"	reduce #, ")", "+", "**" (4)	
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 = "(" Expr ")" .

Analysetabelle

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

Shift-Reduce-Verschmelzung

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

Shift-Reduce-Verschmelzung mit Neu-Nummerierung

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

3) Simulation einer LALR(1)-Analyse

	a	b	c	d	e	#	S	A	B	
0	s2	-	r(4)	-	s4	r(4)	s1	-	s3	(1) S = a A.
1	-	-	s6	-	-	acc	-	s5	-	(2) S = B.
2	-	-	s6	-	-	-	-	s7	-	(3) S = S A b.
3	-	-	r(2)	-	s8	r(2)	-	-	-	(4) S = .
4	r(7)	r(7)	r(7)	r(7)	r(7)	r(7)	-	-	-	(5) A = c B d B.
5	s10	s9	-	-	-	-	-	-	-	(6) A = A a.
6	-	-	-	-	s4	-	-	-	s11	(7) B = e.
7	s10	-	r(1)	-	-	r(1)	-	-	-	(8) B = B e.
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)	-	-	-	

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 d e e e a b

Keller	Eingabe	Aktion	
0	e c e e d e e e a b #	s4	
0,4	c e e d e e e a b #	r(7)	B=e. Länge 1 → ein Element vom Keller
0	B c e e d e e e a b #	s3	
0,3	c e e d e e e a b #	r(2)	S=B. Länge 1 → ein Element vom Keller
0	S c e e d e e e a b #	s1	
0,1	c e e d e e e a b #	s6	
0,1,6	e e d e e e a b #	s4	
0,1,6,4	e d e e e a b #	r(7)	B=e. Länge 1 → ein Element vom Keller
0,1,6	B e d e e e a b #	s11	
0,1,6,11	e d e e e a b #	s8	
0,1,6,11,8	d e e e a b #	r(8)	B=Be. Länge 2 → zwei Elemente vom Keller
0,1,6	B d e e e a b #	s11	
0,1,6,11	d e e e a b #	s12	
0,1,6,11,12	e e e a b #	s4	
0,1,6,11,12,4	e e a b #	r(7)	B=e. Länge 1 → ein Element vom Keller
0,1,6,11,12	B e e a b #	s13	
0,1,6,11,12,13	e e a b #	s8	
0,1,6,11,12,13,8	e a b #	r(8)	B=Be. Länge 2 → zwei Elemente 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 c

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