

Aufgabe 3: Tabellenerzeugung (3)

Folgende Grammatik ist LR(1), aber nicht LALR(1):

- S = d X b.
- S = d Y a.
- S = X a.
- S = Y b.
- X = c.
- Y = c.

Erstellen Sie die Parsertabellen sowohl für die LR(1)-Analyse als auch für die LALR(1)-Analyse und zeigen Sie, wo der Konflikt auftritt. Zeichnen Sie auch die Kellerautomaten für die LR(1)- und die LALR(1)-Analyse und erklären Sie, warum der Konflikt bei den LALR(1)-Tabellen auftritt, nicht aber bei den LR(1)-Tabellen.

Lösung

Die Produktionen werden durchnummeriert, und es wird eine Pseudoproduktion für S' hinzugefügt:

- 0 S' = S #.
- 1 S = d X b.
- 2 S = d Y a.
- 3 S = X a.
- 4 S = Y b.
- 5 X = c.
- 6 Y = c.

Wir erstellen zuerst die LR(1)-Tabellen:

0	S' = . S #		shift d	1
	S = . d X b	/ #	shift c	2
	S = . d Y a	/ #	shift X	3
	S = . X a	/ #	shift Y	4
	X = . c	/ a	shift S	5
	S = . Y b	/ #		
	Y = . c	/ b		
1	S = d . X b	/ #	shift c	6
	X = . c	/ b	shift X	7
	S = d . Y a	/ #	shift Y	8
	Y = . c	/ a		
2	X = . c	/ a	red a	5 (X = c)
	Y = . c	/ b	red b	6 (Y = c)
3	S = X . a	/ #	shift a	9
4	S = Y . b	/ #	shift b	10
5	S' = S . #		acc #	
6	X = c .	/ b	red b	5 (X = c)
	Y = c .	/ a	red a	6 (Y = c)
7	S = d X . b	/ #	shift b	11
8	S = d Y . a	/ #	shift a	12
9	S = X a .	/ #	red #	3 (S = X a)
10	S = Y b .	/ #	red #	4 (S = Y b)
11	S = d X b .	/ #	red #	1 (S = d X b)
12	S = d Y a .	/ #	red #	2 (S = d Y a)

Hier tritt kein LR-Konflikt auf. Wir sehen aber, dass die beiden Zustände 2 und 6 die gleichen Kern-Items haben (wenn auch mit unterschiedlichen Vorgriffssymbolen). Wenn wir LALR(1)-Tabellen erstellen, werden diese beiden Zustände verschmolzen.

0	S' = . S #		shift d	1
	S = . d X b	/ #	shift c	2
	S = . d Y a	/ #	shift X	3
	S = . X a	/ #	shift Y	4
	X = . c	/ a	shift S	5
	S = . Y b	/ #		
	Y = . c	/ b		
1	S = d . X b	/ #	shift c	2 (!)
	X = . c	/ b	shift X	6
	S = d . Y a	/ #	shift Y	7
	Y = . c	/ a		
2	X = . c	/ ab	red a,b	5 (X = c)
	Y = . c	/ ab	red a,b	6 (Y = c)
3	S = X . a	/ #	shift a	8
4	S = Y . b	/ #	shift b	9
5	S' = S . #		acc #	
6	S = d X . b	/ #	shift b	10
7	S = d Y . a	/ #	shift a	11
8	S = X a .	/ #	red #	3 (S = X a)
9	S = Y b .	/ #	red #	4 (S = Y b)
10	S = d X b .	/ #	red #	1 (S = d X b)
11	S = d Y a .	/ #	red #	2 (S = d Y a)

Im Zustand 1 wird mit c in den bereits existierenden Zustand 2 gegangen und nicht in einen neuen Zustand 6 wie bei den LR(1)-Tabellen. Durch das Verschmelzen der Zustände 2 und 6 aus den LR(1)-Tabellen werden die Vorgriffssymbole der Items vereinigt. Beide Items im Zustand 2 haben nun als Vorgriffssymbole sowohl a als auch b, und es kann mit a und b sowohl nach Produktion 5 ($X = c$) als auch nach Produktion 6 ($Y = c$) reduziert werden. Wir haben also einen Reduce-Reduce-Konflikt, der in den LR(1)-Tabellen nicht aufgetreten ist. Die Kellerautomaten für LR(1) und LALR(1) machen das noch deutlicher.

