Aufgabe 2: Tabellenerzeugung (2)

Erstellen Sie für folgende Grammatik Parsertabellen für die Bottom-up-Syntaxanalyse und bestimmen Sie dabei die Wegweisersymbole für die Fehlerbehandlung.

S = a.
S = a X.
X = b.
X = b X c.

Ist diese Grammatik LR(0), LR(1) oder LALR(1)? Begründen Sie Ihre Antwort.

Lösung

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

0 S' = S #.
1 S = a.
2 S = a X.
3 X = b.
4 X = b X c.

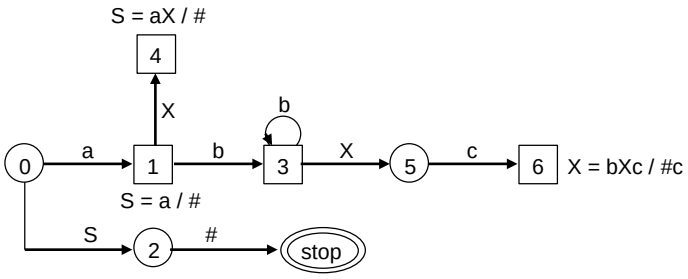
Bei der Tabellenerzeugung bestimmt die erste Terminalsymbolaktion in jedem Zustand das Wegweisersymbol, das in der rechtesten Spalte dargestellt ist.

0	S'= . S #		shift	**a**	1		**a**
	S = . a	/ #	shift	S	2		
	S = . a X	/ #					
1	S = a .	/ #	red	#	1	(S = a)	#
	S = a . X	/ #	shift	b	3		
	X = . b	/ #	shift	X	4		
	X = . b X c	/ #					
2	S' = S . #		acc	#			#
3	X = b .	/ #c	red	#,c	3	(X = b)	#
	X = b . X c	/ #c	shift	b	3 (!)		
	X = . b	/ c	shift	X	5		
	X = . b X c	/ c					
4	S = a X .	/ #	red	#	2	(S = a X)	#
5	X = b X . c	/ #c	shift	**c**	6		**c**
6	X = b X c .	/ #c	red	#,c	4	(X = b X c)	#

Das ergibt folgende Parser-Tabelle:

	a	b	c	#	S	X	guide
0	s1				s2		a
1		s3		r1		s4	#
2				acc			#
3		s3	r3	r3		s5	#
4				r2			#
5			s6				c
6			r4	r4			#

Der Kellerautomat dazu sieht wie folgt aus:



Die Grammatik ist nicht LR(0), weil in den Zuständen 1 und 3 sowohl shift- als auch reduce-Aktionen vorkommen. Man braucht also ein Vorgriffssymbol, um zwischen diesen Aktionen zu unterscheiden.

Sie ist LR(1), weil in jedem Zustand mithilfe des Vorgriffssymbols entschieden werden kann, ob ein shift oder ein reduce ausgeführt werden soll.

Sie ist auch LALR(1), weil durch das Verschmelzen von Zuständen kein LR-Konflikt entstanden ist. Im Zustand 3 gibt es ein shift in einen (potentiell neuen) Zustand, der aber mit dem Zustand 3 verschmolzen wurde, weil die Kern-Items die gleichen sind.