

Aufgabe 1: Tabellenerzeugung (1)

Erstellen Sie für folgende Grammatik Parsertabellen für die Bottom-up-Syntaxanalyse.

- A = A B c.
- A = a.
- B = b.

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' = A #
- 1 A = A B c
- 2 A = a
- 3 B = b

Die Tabellenerzeugung sieht dann wie folgt aus:

- 0 S = . A # shift a 1
- A = .a / #b shift A 2
- A = . A B c / #b
- 1 A = a . / #b red #,b 2 (A = a)
- 2 S = A . # acc #
- A = A . B c / #b shift b 3
- B = . b / c shift B 4
- 3 B = b . / c red c 3 (B = b)
- 4 A = A B . c / #b shift c 5
- 5 A = A B c . / #b red #,b 1 (A = A B c)

Das ergibt folgende Parser-Tabelle:

	a	b	c	#	S	A	B
0	s1					s2	
1		r2		r2			
2		s3		acc			s4
3			r3				
4			s5				
5		r1		r1			

Die Grammatik ist LR(0), weil es keinen Zustand gibt, in dem sowohl eine shift- als auch eine reduce-Aktion ausgeführt werden kann und weil in jedem Reduzierzustand nur nach einer einzigen Produktion reduziert werden kann. Somit kann sich der Parser in jedem Reduzierzustand ohne Vorgriffssymbol für die richtige Reduktion entscheiden.

Da sie LR(0) ist, ist sie auch LR(1): Wenn ohne Vorgriffssymbol zwischen shift und reduce unterschieden werden kann, ist das erstrecht mit 1 Vorgriffssymbol möglich.

Die Grammatik ist auch LALR(1). Es ist zu keiner Verschmelzung von Zuständen gekommen, daher kann auch die LALR(1)-Bedingung nicht von der LR(1)-Bedingung abweichen.