LR-Analyse ist eine Bottomup-Analyse. D.h. es wird der Syntaxbaum bottomup aufgebaut.
Zur Analyse wird ein Kellerautomat eingesetzt.

Aktionen des Kellerautomaten:
==========
Shift Lesen und Kellern des nächsten Eingabesymbols
Reduce Reduzieren von Symbolen am Kellerende zu einem NTS
Accept Satz wurde erkannt
Error Satz wurde nicht erkannt (eventuell Fehlerbehandlung/Wiederaufsatz)

Wir beschäftigen uns hauptsächlich mit LALR(1)-Grammatiken (Lookahead-LR(1)). Die meisten Programmiersprachen lassen sich mit solchen Grammatiken beschreiben, und die Tabellen sind kleiner als bei LR(1). Der Unterschied zwischen LR(1) und LALR(1) ist das Verschmelzen der Kerne, wenn sie bis auf das Vorgriffssymbol gleich sind.

Ausgangsgrammatik:
==========
S = x A | y B z B . A = [A u] . B = y .

Umgeformte Grammatik:
==========
(0) S' = S # . (1) S = x A . (2) S = y B z B . (3) A = A u . (4) A = . (5) B = y .

Tabellenerzeugung:
==========

0	S' = . S #	shift S 1	
	S = . x A / #	shift x 2	
	S = . y B z B / #	shift y 3	
1	S' = S . #	acc #	
2	S = x . A / #	shift A 4	
	A = . A u / #,u	red #,u (4)	u ist Nachfolger wegen
			A-Rekursion
	A = . / #,u		
3	S = y . B z B / #	shift B 5	
	B = . y / z	shift y 6	
4	S = x A . / #	red # (1)	
	A = A . u / #,u	shift u 7	
5	S = y B . z B / #	shift z 8	
6	B = y . / z,#	red z,# (5)	Nachfolger: # kommt
			später dazu
7	A = A u . / #,u	red #,u (3)	
8	S = y B z . B / #	shift B 9	
	B = . y / #	shift y 6	nach 6, da Kern gleich
9	S = y B z B . / #	red # (2)	

Tabelle:
========

	x	y	z	u	#	S	A	B
0	s2	s3	-	-	-	s1	-	-
1	-	-	-	-	acc	-	-	-
2	-	-	-	r(4)	r(4)	-	s4	-
3	-	s6	-	-	-	-	-	s5
4	-	-	-	s7	r(1)	-	-	-
5	-	-	s8	-	-	-	-	-
6	-	-	r(5)	-	r(5)	-	-	-
7	-	-	-	r(3)	r(3)	-	-	-
8	-	s6	-	-	-	-	-	s9
9	-	-	-	-	r(2)	-	-	-

Simulation:
===========

Satz: y y z y

Keller	Eingabe	Aktion
0	y y z y #	s3
0 3	y z y #	s6
0 3 6	z y #	r(5) B = y .
0 3	B z y #	s5
0 3 5	z y #	s8
0 3 5 8	y #	s6
0 3 5 8 6	#	r(5) B = y .
0 3 5 8	B #	s9
0 3 5 8 9	#	r(2) S = y B z B .
0	S #	s1
0 1	#	acc

Satz: x

Keller	Eingabe	Aktion
0	x #	s2
0 2	#	r(4) A = .
0 2	A #	s4
0 2 4	#	r(1) S = x A .
0	S #	s1
0 1	#	acc

Satz: x u u

Keller	Eingabe	Aktion
0	x u u #	s2
0 2	u u #	r(4) A = .
0 2	A u u #	s4
0 2 4	u u #	s7
0 2 4 7	u #	r(3) A = A u .
0 2	A u #	s4
0 2 4	u #	s7
0 2 4 7	#	r(3) A = A u .
0 2	A #	s4
0 2 4	#	r(1) S = x A .
0	S #	s1
0 1	#	acc

Satz: x u y

Keller	Eingabe	Aktion
0	x u y #	s2
0 2	u y #	r(4) A = .
0 2	A u y #	s4
0 2 4	u y #	s7
0 2 4 7	y #	error