Exercise 5: LR Error Handling (2)

Given the following grammar with the corresponding parser table and the guide symbols:

	a	*b*	#	*S*	*X*	*Y*	*guide*
0	s1	-	-	s3	s2	-	a
1	s4	r3	-	-	-	-	b
2	-	s5	-	-	-	(s6)	b
3	s1	-	acc	-	s7	-	#
4	-	(s8)	-	-	-	-	b
5	r5	s9	r5	-	-	-	#
6	r1	-	r1	-	-	-	#
7	-	s5	-	-	-	(s10)	b
8	-	r4	-	-	-	-	b
9	(s11)	-	-	-	-	-	a
10	r2	-	r2	-	-	-	#
11	r6	-	r6	-	-	-	#

grammar

0 S' = S #
1 S = X Y
2 S = S X Y
3 X = a
4 X = a a b
5 Y = b
6 Y = b b a

The marked shift actions lead to pure reduce states. They can therefore be converted to shiftred actions and the corresponding reduce states can be deleted, resulting in the following table:

	a	*b*	#	*S*	*X*	*Y*	*guide*
0	s1	-	-	s3	s2	-	a
1	s4	r3	-	-	-	-	b
2	-	s5	-	-	-	**sr1**	b
3	s1	-	acc	-	s7	-	#
4	-	**sr4**	-	-	-	-	b
5	r5	s9	r5	-	-	-	#
7	-	s5	-	-	-	**sr2**	b
9	**sr6**	-	-	-	-	-	a

The guide symbols in the non-deleted states are retained. Show the analysis of the faulty input

b a b #

including recovery.

Solution

The analysis starts in state 0 with the complete input:

stack	*input*	*action*
0	b a b #	-- error

An error already occurs here because the symbol b is not valid in state 0. The recovery begins, in which the parser is steered from the error state to the final state (along the escape route) using the guide symbols. All valid symbols in the traversed states are collected as anchors. For better readability, the intermediate steps for reduce and shiftred actions are shown here in gray.

stack	*guide*		*action*	*anchors*
0	a		s1	a
0 1	b		r3 (X = a), s2	a, b
0 2	b		s5	b
0 2 5	#		r5 (Y = b)	a, b, #
0 2		Y	sr1 (S = X Y)	
0		S	s3	
0 3	#		acc	#

The set of anchors includes the symbols a, b and #. As the remaining input b a b # already begins with an anchor, no symbols are deleted from it. The symbol b is the anchor with which parsing can continue.

The parser now runs again from the error state using the guide symbols along the escape route until it reaches a state in which the anchor b is valid. All shifted guide symbols are inserted at the error location.

stack	*guide*	*action*	*inserted*
0	a	s1	a
0 1			

In state 1, the anchor b is already valid, so the analysis of the remaining input b a b # is continued here. The symbol a has been inserted into the source text at the error location. Here too, the intermediate steps of reduce and shiftred actions are shown in gray for the sake of clarity.

stack	*input*	*action*
0 1	b a b #	r3 (X = a), s2
0 2	b a b #	s5
0 2 5	a b #	r5 (Y = b)
0 2	Y a b #	sr1 (S = X Y)
0	S a b #	s3
0 3	a b #	s1
0 3 1	b #	r3 (X = a)
0 3	X b #	s7
0 3 7	b #	s5
0 3 7 5	#	r5 (Y = b)
0 3 7	Y #	sr2 (S = S X Y)
0	S #	s3
0 3	#	acc

The analysis has been successfully completed. The corrected input is

a b a b #

and the error message is

line ... col ...: "a" inserted