

Grammar:
 0. start ::= statement
 1. statement ::= PRINT expression
 2. expression ::= expression PLUS expression
 3. expression ::= INT

State 0
 start ::= . statement
 statement ::= . PRINT expression

State 1
 statement ::= PRINT . expression
 expression ::= . expression PLUS expression
 expression ::= . INT

State 2
 start ::= statement .

State 4
 expression ::= INT .
 end, reduce by rule 3
 PLUS, reduce by rule 3

State 3
 statement ::= PRINT expression .
 expression ::= expression . PLUS expression
 end, reduce by rule 1

State 6
 expression ::= expression PLUS expression .
 expression ::= expression . PLUS expression
 end, reduce by rule 2
PLUS, reduce by rule 2

State 5
 expression ::= expression PLUS . expression
 expression ::= . expression PLUS expression
 expression ::= . INT

PLUS, shift

expression, goto

Example parse of 'print 1 + 2'

Stack	Input	Action
[]	'print 1 + 2'	shift to state 1
[(1,PRINT)]	'1 + 2'	shift to state 4
[(1,PRINT),(4,INT)]	'+ 2'	reduce by rule 3 to state 1, goto 3
[(1,PRINT),(3,expression)]	'+ 2'	shift to state 5
[(1,PRINT),(3,expression),(5,+)]	'2'	shift to state 4
[(1,PRINT),(3,expression),(5,+),(4,INT)]	"	reduce by rule 3 to state 5, goto 6
[(1,PRINT),(3,expression),(5,+),(6,expression)]	"	reduce by rule 2 to state 1, goto 3
[(1,PRINT),(3,expression)]	"	reduce by rule 1 to state 0, goto 2
[(2,statement)]	"	accept