

Grammar:

0. $\text{start} ::= \text{stmt}$
1. $\text{stmt} ::= \text{"print"} \text{ exp}$
2. $\text{exp} ::= \text{exp "+" exp}$
3. $\text{exp} ::= \text{INT}$

State 0

$\text{start} ::= . \text{stmt}$
 $\text{stmt} ::= . \text{"print"} \text{ exp}$

State 1

$\text{stmt} ::= \text{"print"} . \text{exp}$
 $\text{exp} ::= . \text{exp "+" exp}$
 $\text{exp} ::= . \text{INT}$

State 2

$\text{start} ::= \text{stmt} .$

State 4

$\text{exp} ::= \text{INT} .$

end, reduce by rule 3
"+", reduce by rule 3

State 3

$\text{stmt} ::= \text{"print"} \text{ exp} .$
 $\text{exp} ::= \text{exp} . \text{"+" exp}$

end, reduce by rule 1

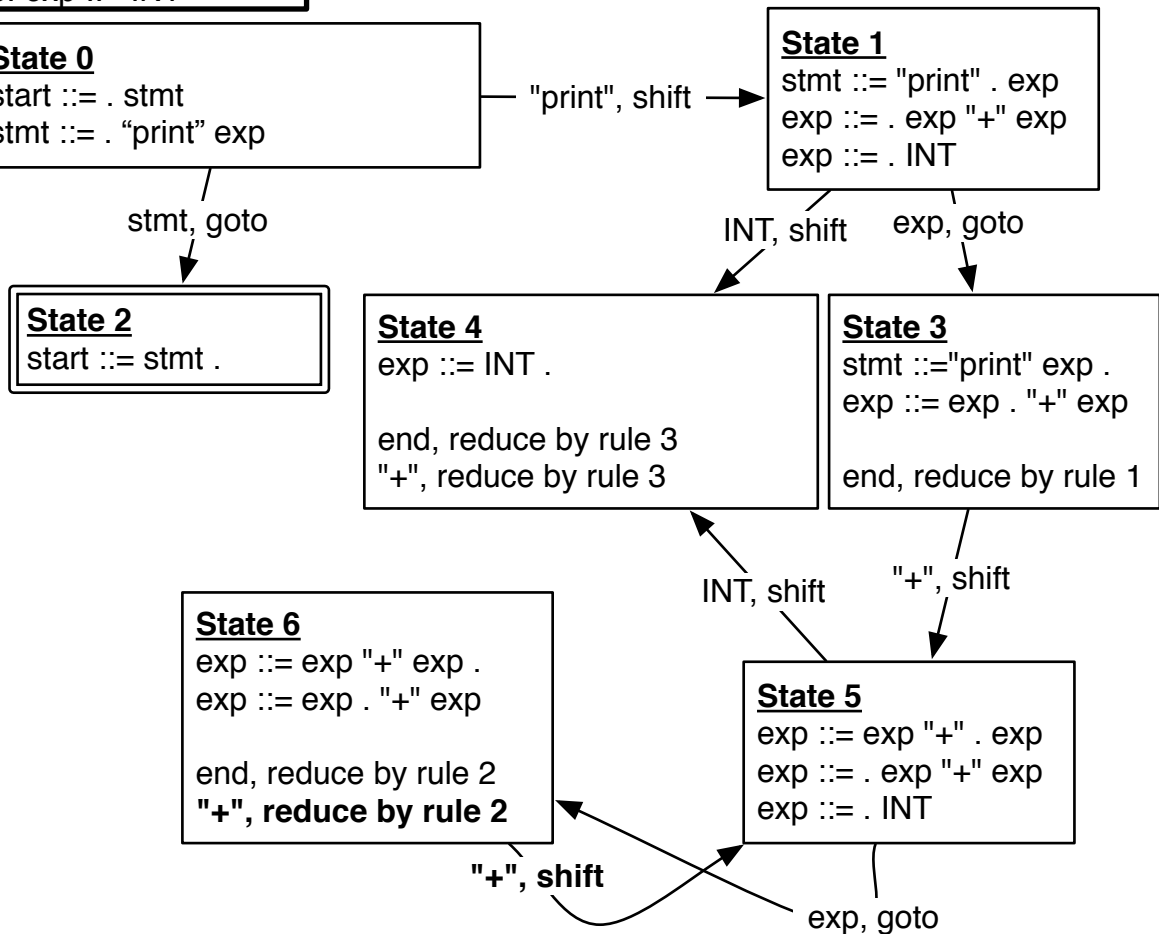
State 6

$\text{exp} ::= \text{exp "+" exp} .$
 $\text{exp} ::= \text{exp} . \text{"+" exp}$

end, reduce by rule 2
"+", reduce by rule 2

State 5

$\text{exp} ::= \text{exp "+"} . \text{exp}$
 $\text{exp} ::= . \text{exp "+" exp}$
 $\text{exp} ::= . \text{INT}$

**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,exp)]	' + 2'	shift to state 5
[(1,"print"),(3,exp),(5,+)]	'2'	shift to state 4
[(1,"print"),(3,exp),(5,+),(4,INT)]	"	reduce by rule 3 to state 5, goto 6
[(1,"print"),(3,exp),(5,+),(6,exp)]	"	reduce by rule 2 to state 1, goto 3
[(1,"print"),(3,exp)]	"	reduce by rule 1 to state 0, goto 2
[(2,stmt)]	"	accept