```
Conflict in state 13 between rule 6 and token '+' resolved as red
uce.
Conflict in state 13 between rule 6 and token '*' resolved as shi
ft.
Conflict in state 13 between rule 6 and token '^' resolved as shi
ft.
Conflict in state 14 between rule 7 and token '+' resolved as red
uce.
Conflict in state 14 between rule 7 and token '*' resolved as red
uce.
Conflict in state 14 between rule 7 and token '^' resolved as shi
ft.
Conflict in state 15 between rule 8 and token '+' resolved as red
uce.
Conflict in state 15 between rule 8 and token '*' resolved as red
uce.
Conflict in state 15 between rule 8 and token '^' resolved as shi
ft.

Grammar
rule 1    input ->                  /* empty */
rule 2    input -> input line
rule 3    line -> '\n'
rule 4    line -> exp '\n'
rule 5    exp -> NUM
rule 6    exp -> exp '+' exp
rule 7    exp -> exp '*' exp
rule 8    exp -> exp '^' exp
rule 9    exp -> '(' exp ')'

Terminals, with rules where they appear

$ (-1)
'\n' (10) 3 4
'(' (40) 9
')' (41) 9
'*' (42) 7
'+' (43) 6
'^' (94) 8
error (256)
NUM (258) 5

Nonterminals, with rules where they appear
```

```
input (10)    on left: 1 2, on right: 2
line (11)     on left: 3 4, on right: 2
exp (12)      on left: 5 6 7 8 9, on right: 4 6 7 8 9


state 0
    $default    reduce using rule 1 (input)

    input       go to state 1


state 1
    input  ->  input . line   (rule 2)

    $           go to state 16
    NUM         shift, and go to state 2
    '\n'        shift, and go to state 3
    '('         shift, and go to state 4

    line        go to state 5
    exp         go to state 6


state 2
    exp  ->  NUM .   (rule 5)

    $default    reduce using rule 5 (exp)


state 3
    line  ->  '\n' .   (rule 3)

    $default    reduce using rule 3 (line)


state 4
    exp  ->  '(' . exp ')'   (rule 9)

    NUM         shift, and go to state 2
    '('         shift, and go to state 4

    exp         go to state 7
```

```
state 5
    input  ->  input line .   (rule 2)

    $default    reduce using rule 2 (input)


state 6
    line  ->  exp . '\n'   (rule 4)
    exp  ->  exp . '+' exp   (rule 6)
    exp  ->  exp . '*' exp   (rule 7)
    exp  ->  exp . '^' exp   (rule 8)

    '+'         shift, and go to state 8
    '*'         shift, and go to state 9
    '^'         shift, and go to state 10
    '\n'        shift, and go to state 11


state 7
    exp  ->  exp . '+' exp   (rule 6)
    exp  ->  exp . '*' exp   (rule 7)
    exp  ->  exp . '^' exp   (rule 8)
    exp  ->  '(' exp . ')'   (rule 9)

    '+'         shift, and go to state 8
    '*'         shift, and go to state 9
    '^'         shift, and go to state 10
    ')'         shift, and go to state 12


state 8
    exp  ->  exp '+' . exp   (rule 6)

    NUM         shift, and go to state 2
    '('         shift, and go to state 4

    exp         go to state 13


state 9
    exp  ->  exp '*' . exp   (rule 7)
```

```
    NUM         shift, and go to state 2
    '('         shift, and go to state 4

    exp         go to state 14


state 10
    exp  ->  exp '^' . exp   (rule 8)

    NUM         shift, and go to state 2
    '('         shift, and go to state 4

    exp         go to state 15


state 11
    line  ->  exp '\n' .   (rule 4)

    $default    reduce using rule 4 (line)


state 12
    exp  ->  '(' exp ')' .   (rule 9)

    $default    reduce using rule 9 (exp)


state 13
    exp  ->  exp . '+' exp   (rule 6)
    exp  ->  exp '+' exp .   (rule 6)
    exp  ->  exp . '*' exp   (rule 7)
    exp  ->  exp . '^' exp   (rule 8)

    '*'         shift, and go to state 9
    '^'         shift, and go to state 10

    $default    reduce using rule 6 (exp)


state 14
    exp  ->  exp . '+' exp   (rule 6)
    exp  ->  exp . '*' exp   (rule 7)
```

```
    exp  ->  exp '*' exp .   (rule 7)
    exp  ->  exp . '^' exp   (rule 8)

    '^'          shift, and go to state 10

    $default     reduce using rule 7 (exp)


state 15
    exp  ->  exp . '+' exp   (rule 6)
    exp  ->  exp . '*' exp   (rule 7)
    exp  ->  exp . '^' exp   (rule 8)
    exp  ->  exp '^' exp .   (rule 8)

    '^'          shift, and go to state 10

    $default     reduce using rule 8 (exp)


state 16
    $            go to state 17


state 17
    $default     accept
```