```
Terminals unused in grammar

   ROOT


Grammar

    0 $accept: program $end

    1 program: stmtseq

    2 stmtseq: stmtseq expr ';'
    3        | stmtseq error ';'
    4        | stmtseq ';'
    5        | %empty

    6 expr: expr '=' expr
    7     | expr '+' expr
    8     | expr '-' expr
    9     | expr '*' expr
   10     | expr '/' expr
   11     | expr '^' expr
   12     | '+' expr
   13     | '-' expr
   14     | '(' expr ')'
   15     | IDENT
   16     | NUMBER


Terminals, with rules where they appear

$end (0) 0
'(' (40) 14
')' (41) 14
'*' (42) 9
'+' (43) 7 12
'-' (45) 8 13
'/' (47) 10
';' (59) 2 3 4
'=' (61) 6
'^' (94) 11
error (256) 3
ROOT (258)
IDENT (259) 15
NUMBER (260) 16
POS (261)
NEG (262)


Nonterminals, with rules where they appear

$accept (17)
    on left: 0
program (18)
    on left: 1, on right: 0
stmtseq (19)
    on left: 2 3 4 5, on right: 1 2 3 4
expr (20)
```

```
    on left: 6 7 8 9 10 11 12 13 14 15 16, on right: 2 6 7 8 9 10 11
    12 13 14


State 0

    0 $accept: . program $end

    $default  reduce using rule 5 (stmtseq)

    program  go to state 1
    stmtseq  go to state 2


State 1

    0 $accept: program . $end

    $end  shift, and go to state 3


State 2

    1 program: stmtseq .
    2 stmtseq: stmtseq . expr ';'
    3        | stmtseq . error ';'
    4        | stmtseq . ';'

    error   shift, and go to state 4
    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    '+'     shift, and go to state 7
    '-'     shift, and go to state 8
    ';'     shift, and go to state 9
    '('     shift, and go to state 10

    $end  reduce using rule 1 (program)

    expr  go to state 11


State 3

    0 $accept: program $end .

    $default  accept


State 4

    3 stmtseq: stmtseq error . ';'

    ';'  shift, and go to state 12


State 5

   15 expr: IDENT .
```

```

    $default  reduce using rule 15 (expr)


State 6

   16 expr: NUMBER .

    $default  reduce using rule 16 (expr)


State 7

   12 expr: ’+’ . expr

    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    ’+’     shift, and go to state 7
    ’-’     shift, and go to state 8
    ’(’     shift, and go to state 10

    expr  go to state 13


State 8

   13 expr: ’-’ . expr

    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    ’+’     shift, and go to state 7
    ’-’     shift, and go to state 8
    ’(’     shift, and go to state 10

    expr  go to state 14


State 9

    4 stmtseq: stmtseq ’;’ .

    $default  reduce using rule 4 (stmtseq)


State 10

   14 expr: ’(’ . expr ’)’

    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    ’+’     shift, and go to state 7
    ’-’     shift, and go to state 8
    ’(’     shift, and go to state 10

    expr  go to state 15


State 11
```

```

    2 stmtseq: stmtseq expr . ’;’
    6 expr: expr . ’=’ expr
    7     | expr . ’+’ expr
    8     | expr . ’-’ expr
    9     | expr . ’*’ expr
   10     | expr . ’/’ expr
   11     | expr . ’^’ expr

    ’=’  shift, and go to state 16
    ’+’  shift, and go to state 17
    ’-’  shift, and go to state 18
    ’*’  shift, and go to state 19
    ’/’  shift, and go to state 20
    ’^’  shift, and go to state 21
    ’;’  shift, and go to state 22


State 12

    3 stmtseq: stmtseq error ’;’ .

    $default  reduce using rule 3 (stmtseq)


State 13

    6 expr: expr . ’=’ expr
    7     | expr . ’+’ expr
    8     | expr . ’-’ expr
    9     | expr . ’*’ expr
   10     | expr . ’/’ expr
   11     | expr . ’^’ expr
   12     | ’+’ expr .

    $default  reduce using rule 12 (expr)


State 14

    6 expr: expr . ’=’ expr
    7     | expr . ’+’ expr
    8     | expr . ’-’ expr
    9     | expr . ’*’ expr
   10     | expr . ’/’ expr
   11     | expr . ’^’ expr
   13     | ’-’ expr .

    $default  reduce using rule 13 (expr)


State 15

    6 expr: expr . ’=’ expr
    7     | expr . ’+’ expr
    8     | expr . ’-’ expr
    9     | expr . ’*’ expr
   10     | expr . ’/’ expr
   11     | expr . ’^’ expr
   14     | ’(’ expr . ’)’

    ’=’  shift, and go to state 16
    ’+’  shift, and go to state 17
    ’-’  shift, and go to state 18
    ’*’  shift, and go to state 19
    ’/’  shift, and go to state 20
    ’^’  shift, and go to state 21
    ’)’  shift, and go to state 23


State 16

    6 expr: expr ’=’ . expr

    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    ’+’     shift, and go to state 7
    ’-’     shift, and go to state 8
    ’(’     shift, and go to state 10

    expr  go to state 24


State 17

    7 expr: expr ’+’ . expr

    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    ’+’     shift, and go to state 7
    ’-’     shift, and go to state 8
    ’(’     shift, and go to state 10

    expr  go to state 25


State 18

    8 expr: expr ’-’ . expr

    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    ’+’     shift, and go to state 7
    ’-’     shift, and go to state 8
    ’(’     shift, and go to state 10

    expr  go to state 26


State 19

    9 expr: expr ’*’ . expr

    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    ’+’     shift, and go to state 7
    ’-’     shift, and go to state 8
    ’(’     shift, and go to state 10

    expr  go to state 27


State 20

   10 expr: expr ’/’ . expr

    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    ’+’     shift, and go to state 7
    ’-’     shift, and go to state 8
    ’(’     shift, and go to state 10

    expr  go to state 28


State 21

   11 expr: expr ’^’ . expr

    IDENT   shift, and go to state 5
    NUMBER  shift, and go to state 6
    ’+’     shift, and go to state 7
    ’-’     shift, and go to state 8
    ’(’     shift, and go to state 10

    expr  go to state 29


State 22

    2 stmtseq: stmtseq expr ’;’ .

    $default  reduce using rule 2 (stmtseq)


State 23

   14 expr: ’(’ expr ’)’ .

    $default  reduce using rule 14 (expr)


State 24

    6 expr: expr . ’=’ expr
    6     | expr ’=’ expr .
    7     | expr . ’+’ expr
    8     | expr . ’-’ expr
    9     | expr . ’*’ expr
   10     | expr . ’/’ expr
   11     | expr . ’^’ expr

    ’=’  shift, and go to state 16
    ’+’  shift, and go to state 17
    ’-’  shift, and go to state 18
    ’*’  shift, and go to state 19
    ’/’  shift, and go to state 20
    ’^’  shift, and go to state 21

    $default  reduce using rule 6 (expr)


State 25

    6 expr: expr . ’=’ expr
    7     | expr . ’+’ expr
    7     | expr ’+’ expr .
    8     | expr . ’-’ expr
    9     | expr . ’*’ expr
   10     | expr . ’/’ expr
   11     | expr . ’^’ expr

    ’*’  shift, and go to state 19
    ’/’  shift, and go to state 20
    ’^’  shift, and go to state 21

    $default  reduce using rule 7 (expr)


State 26

    6 expr: expr . ’=’ expr
    7     | expr . ’+’ expr
    8     | expr . ’-’ expr
    8     | expr ’-’ expr .
    9     | expr . ’*’ expr
   10     | expr . ’/’ expr
   11     | expr . ’^’ expr

    ’*’  shift, and go to state 19
    ’/’  shift, and go to state 20
    ’^’  shift, and go to state 21

    $default  reduce using rule 8 (expr)


State 27

    6 expr: expr . ’=’ expr
    7     | expr . ’+’ expr
    8     | expr . ’-’ expr
    9     | expr . ’*’ expr
    9     | expr ’*’ expr .
   10     | expr . ’/’ expr
   11     | expr . ’^’ expr

    ’^’  shift, and go to state 21

    $default  reduce using rule 9 (expr)


State 28
```

```

    6 expr: expr . ’=’ expr
    7     | expr . ’+’ expr
    8     | expr . ’-’ expr
    9     | expr . ’*’ expr
   10     | expr . ’/’ expr
   10     | expr ’/’ expr .
   11     | expr . ’^’ expr

    ’^’  shift, and go to state 21

    $default  reduce using rule 10 (expr)


State 29

    6 expr: expr . ’=’ expr
    7     | expr . ’+’ expr
    8     | expr . ’-’ expr
    9     | expr . ’*’ expr
   10     | expr . ’/’ expr
   11     | expr . ’^’ expr
   11     | expr ’^’ expr .

    ’^’  shift, and go to state 21

    $default  reduce using rule 11 (expr)
```

```
// $Id: test1.in,v 1.1 2015-07-08 13:29:32-07 - - $
a=b*c+d*e;
```

```
; 0.  1: # 1 "test1.in"; 1.  0:
; 1.  1: # 1 "<built-in>"
; 2.  1: # 1 "<command-line>"
; 3.  1: # 1 "/usr/include/stdc-predef.h" 1 3 4
; 4.  1: # 1 "<command-line>" 2
; 5.  1: # 1 "test1.in"
; 6.  1:
; 6.  2: a=b*c+d*e;
; ROOT "<<ROOT>>" (0.0.0)
;    ’=’ "=" (6.2.2)
;       IDENT "a" (6.2.1)
;       ’+’ "+" (6.2.6)
;          ’*’ "*" (6.2.4)
;             IDENT "b" (6.2.3)
;             IDENT "c" (6.2.5)
;          ’*’ "*" (6.2.8)
;             IDENT "d" (6.2.7)
;             IDENT "e" (6.2.9)

          pushvar   b                   ; test1.in 2.3
          pushvar   c                   ; test1.in 2.5
          mul                           ; test1.in 2.4
          pushvar   d                   ; test1.in 2.7
          pushvar   e                   ; test1.in 2.9
          mul                           ; test1.in 2.8
          add                           ; test1.in 2.6
          popvar    a                   ; test1.in 2.1
```

```
Command: zexprsm -ly test1.in
-- popen (/usr/bin/cpp test1.in), fileno(yyin) = 4
Starting parse
Entering state 0
Reducing stack by rule 5 (line 46):
-> $$ = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>":)
Stack now 0
Entering state 2
Reading a token: --(end of buffer or a NUL)
--accepting rule at line 31 ("# 1 "test1.in"")
--included # 1 "test1.in"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "<built-in>"")
--included # 1 "<built-in>"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "<command-line>"")
--included # 1 "<command-line>"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "/usr/include/stdc-predef.h" 1 3 4")
--included # 1 "/usr/include/stdc-predef.h"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "<command-line>" 2")
--included # 1 "<command-line>"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "test1.in"")
--included # 1 "test1.in"
--accepting rule at line 33 ("
")
--accepting rule at line 33 ("
")
--accepting rule at line 36 ("a")
Next token is token IDENT (0x5a279d0->{IDENT 6.2.1 "a":)
Shifting token IDENT (0x5a279d0->{IDENT 6.2.1 "a":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a279d0->{IDENT 6.2.1 "a":)
-> $$ = nterm expr (0x5a279d0->{IDENT 6.2.1 "a":)
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 37 ("=")
Next token is token ’=’ (0x5a27b80->{’=’ 6.2.2 "=":)
Shifting token ’=’ (0x5a27b80->{’=’ 6.2.2 "=":)
Entering state 16
Reading a token: --accepting rule at line 36 ("b")
Next token is token IDENT (0x5a27cc0->{IDENT 6.2.3 "b":)
Shifting token IDENT (0x5a27cc0->{IDENT 6.2.3 "b":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a27cc0->{IDENT 6.2.3 "b":)
-> $$ = nterm expr (0x5a27cc0->{IDENT 6.2.3 "b":)
Stack now 0 2 11 16
Entering state 24
Reading a token: --accepting rule at line 40 ("*")
```

```
Next token is token ’*’ (0x5a27e00->{’*’ 6.2.4 "*":)
Shifting token ’*’ (0x5a27e00->{’*’ 6.2.4 "*":)
Entering state 19
Reading a token: --accepting rule at line 36 ("c")
Next token is token IDENT (0x5a27fe0->{IDENT 6.2.5 "c":)
Shifting token IDENT (0x5a27fe0->{IDENT 6.2.5 "c":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a27fe0->{IDENT 6.2.5 "c":)
-> $$ = nterm expr (0x5a27fe0->{IDENT 6.2.5 "c":)
Stack now 0 2 11 16 24 19
Entering state 27
Reading a token: --accepting rule at line 38 ("+")
Next token is token ’+’ (0x5a28120->{’+’ 6.2.6 "+":)
Reducing stack by rule 9 (line 52):
   $1 = nterm expr (0x5a27cc0->{IDENT 6.2.3 "b":)
   $2 = token ’*’ (0x5a27e00->{’*’ 6.2.4 "*":)
   $3 = nterm expr (0x5a27fe0->{IDENT 6.2.5 "c":)
-> $$ = nterm expr (0x5a27e00->{’*’ 6.2.4 "*": 0x5a27cc0 0x5a27fe0)
Stack now 0 2 11 16
Entering state 24
Next token is token ’+’ (0x5a28120->{’+’ 6.2.6 "+":)
Shifting token ’+’ (0x5a28120->{’+’ 6.2.6 "+":)
Entering state 17
Reading a token: --accepting rule at line 36 ("d")
Next token is token IDENT (0x5a28300->{IDENT 6.2.7 "d":)
Shifting token IDENT (0x5a28300->{IDENT 6.2.7 "d":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a28300->{IDENT 6.2.7 "d":)
-> $$ = nterm expr (0x5a28300->{IDENT 6.2.7 "d":)
Stack now 0 2 11 16 24 17
Entering state 25
Reading a token: --accepting rule at line 40 ("*")
Next token is token ’*’ (0x5a28440->{’*’ 6.2.8 "*":)
Shifting token ’*’ (0x5a28440->{’*’ 6.2.8 "*":)
Entering state 19
Reading a token: --accepting rule at line 36 ("e")
Next token is token IDENT (0x5a28520->{IDENT 6.2.9 "e":)
Shifting token IDENT (0x5a28520->{IDENT 6.2.9 "e":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a28520->{IDENT 6.2.9 "e":)
-> $$ = nterm expr (0x5a28520->{IDENT 6.2.9 "e":)
Stack now 0 2 11 16 24 17 25 19
Entering state 27
Reading a token: --accepting rule at line 45 (";")
Next token is token ’;’ (0x5a28660->{’;’ 6.2.10 ";":)
Reducing stack by rule 9 (line 52):
   $1 = nterm expr (0x5a28300->{IDENT 6.2.7 "d":)
   $2 = token ’*’ (0x5a28440->{’*’ 6.2.8 "*":)
   $3 = nterm expr (0x5a28520->{IDENT 6.2.9 "e":)
-> $$ = nterm expr (0x5a28440->{’*’ 6.2.8 "*": 0x5a28300 0x5a28520)
Stack now 0 2 11 16 24 17
Entering state 25
Next token is token ’;’ (0x5a28660->{’;’ 6.2.10 ";":)
Reducing stack by rule 7 (line 50):
   $1 = nterm expr (0x5a27e00->{’*’ 6.2.4 "*": 0x5a27cc0 0x5a27fe0)
   $2 = token ’+’ (0x5a28120->{’+’ 6.2.6 "+":)
   $3 = nterm expr (0x5a28440->{’*’ 6.2.8 "*": 0x5a28300 0x5a28520)
-> $$ = nterm expr (0x5a28120->{’+’ 6.2.6 "+": 0x5a27e00 0x5a28440)
Stack now 0 2 11 16
Entering state 24
Next token is token ’;’ (0x5a28660->{’;’ 6.2.10 ";":)
Reducing stack by rule 6 (line 49):
   $1 = nterm expr (0x5a279d0->{IDENT 6.2.1 "a":)
   $2 = token ’=’ (0x5a27b80->{’=’ 6.2.2 "=":)
   $3 = nterm expr (0x5a28120->{’+’ 6.2.6 "+": 0x5a27e00 0x5a28440)
-> $$ = nterm expr (0x5a27b80->{’=’ 6.2.2 "=": 0x5a279d0 0x5a28120)
Stack now 0 2
Entering state 11
Next token is token ’;’ (0x5a28660->{’;’ 6.2.10 ";":)
Shifting token ’;’ (0x5a28660->{’;’ 6.2.10 ";":)
Entering state 22
Reducing stack by rule 2 (line 43):
   $1 = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>":)
   $2 = nterm expr (0x5a27b80->{’=’ 6.2.2 "=": 0x5a279d0 0x5a28120)
   $3 = token ’;’ (0x5a28660->{’;’ 6.2.10 ";":)
Deleting astree (0x5a28660->{’;’ 6.2.10 ";":)
-> $$ = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27b80)
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 33 ("
")
--(end of buffer or a NUL)
--EOF (start condition 0)
Now at end of input.
Reducing stack by rule 1 (line 40):
   $1 = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27b80)
-> $$ = nterm program (nullptr)
Stack now 0
Entering state 1
Now at end of input.
Shifting token $end ()
Entering state 3
Stack now 0 1 3
Cleanup: popping token $end ()
Cleanup: popping nterm program (nullptr)
Dumping parser::root:
0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27b80
   0x5a27b80->{’=’ 6.2.2 "=": 0x5a279d0 0x5a28120
      0x5a279d0->{IDENT 6.2.1 "a":
      0x5a28120->{’+’ 6.2.6 "+": 0x5a27e00 0x5a28440
         0x5a27e00->{’*’ 6.2.4 "*": 0x5a27cc0 0x5a27fe0
            0x5a27cc0->{IDENT 6.2.3 "b":
            0x5a27fe0->{IDENT 6.2.5 "c":
         0x5a28440->{’*’ 6.2.8 "*": 0x5a28300 0x5a28520
            0x5a28300->{IDENT 6.2.7 "d":
            0x5a28520->{IDENT 6.2.9 "e":
Dumping string_set:
string_set[   0]:   10959529184379665549 0x5a280c8->"c"
                    3729804957429652673 0x5a23418->"<<ROOT>>"
string_set[   1]:    2540012008095083820 0x5a28608->"e"
                   11597697714117577063 0x5a27c68->"="
                   10838281452030117757 0x5a27da8->"b"
string_set[   3]:    1370730858159036685 0x5a28208->"+"
```

```
                    6637313742931709005 0x5a27ee8->"*"
string_set[   4]:    5344662657899890615 0x5a28748->";"
string_set[   5]:   14494284460613645429 0x5a283e8->"d"
string_set[   7]:    4993892634952068459 0x5a27ab8->"a"
load_factor = 0.909
bucket_count = 11
max_bucket_size = 3
Deleting astree (0x5a28520->{IDENT 6.2.9 "e":)
Deleting astree (0x5a28300->{IDENT 6.2.7 "d":)
Deleting astree (0x5a28440->{'*' 6.2.8 "*":)
Deleting astree (0x5a27fe0->{IDENT 6.2.5 "c":)
Deleting astree (0x5a27cc0->{IDENT 6.2.3 "b":)
Deleting astree (0x5a27e00->{'*' 6.2.4 "*":)
Deleting astree (0x5a28120->{'+' 6.2.6 "+":)
Deleting astree (0x5a279d0->{IDENT 6.2.1 "a":)
Deleting astree (0x5a27b80->{'=' 6.2.2 "=":)
Deleting astree (0x5a23320->{ROOT 0.0.0 "<<ROOT>>":)
```

```
==11301== Memcheck, a memory error detector
==11301== Copyright (C) 2002-2017, and GNU GPL’d, by Julian Seward et al.
==11301== Using Valgrind-3.14.0.GIT and LibVEX; rerun with -h for copyright info
==11301== Command: zexprsm -ly test1.in
==11301== Parent PID: 11300
==11301==
==11301==
==11301== HEAP SUMMARY:
==11301==     in use at exit: 0 bytes in 0 blocks
==11301==   total heap usage: 61 allocs, 61 frees, 18,675 bytes allocated
==11301==
==11301== All heap blocks were freed -- no leaks are possible
==11301==
==11301== For counts of detected and suppressed errors, rerun with: -v
==11301== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
EXIT STATUS = 0
```

```
// $Id: test4.in,v 1.2 2015-07-08 13:43:56-07 - - $
#include "test4a.inh"
#include "test4b.inh"
3*4;
```

```
; 0.  1: # 1 "test4.in"; 1.  0:
; 1.  1: # 1 "<built-in>"
; 2.  1: # 1 "<command-line>"
; 3.  1: # 1 "/usr/include/stdc-predef.h" 1 3 4
; 4.  1: # 1 "<command-line>" 2
; 5.  1: # 1 "test4.in"
; 6.  1:
; 6.  2: # 1 "test4a.inh" 1
; 7.  1:
; 7.  2:
; 7.  3: pi=3.141592653589793238462643383280;
; 7.  4: pi;
; 7.  5: # 3 "test4.in" 2
; 8.  3: # 1 "test4b.inh" 1
; 9.  1:
; 9.  2: a=pi*r^2;
; 9.  3: 3.141592653589793238462643383280;
; 9.  4: # 4 "test4.in" 2
;10.  4: 3*4;
; ROOT "<<ROOT>>" (0.0.0)
;    ’=’ "=" (7.3.3)
;       IDENT "pi" (7.3.1)
;       NUMBER "3.141592653589793238462643383280" (7.3.4)
;    IDENT "pi" (7.4.1)
;    ’=’ "=" (9.2.2)
;       IDENT "a" (9.2.1)
;       ’*’ "*" (9.2.5)
;          IDENT "pi" (9.2.3)
;          ’^’ "^" (9.2.7)
;             IDENT "r" (9.2.6)
;             NUMBER "2" (9.2.8)
;    NUMBER "3.141592653589793238462643383280" (9.3.1)
;    ’*’ "*" (10.4.2)
;       NUMBER "3" (10.4.1)
;       NUMBER "4" (10.4.3)

          pushnum   3.141592653589793238462643383280; test4a.inh 3.4
          popvar    pi                  ; test4a.inh 3.1
          pushvar   pi                  ; test4a.inh 4.1
          pushvar   pi                  ; test4b.inh 2.3
          pushvar   r                   ; test4b.inh 2.6
          pushnum   2                   ; test4b.inh 2.8
          pow                           ; test4b.inh 2.7
          mul                           ; test4b.inh 2.5
          popvar    a                   ; test4b.inh 2.1
          pushnum   3.141592653589793238462643383280; test4b.inh 3.1
          pushnum   3                   ; test4.in 4.1
          pushnum   4                   ; test4.in 4.3
          mul                           ; test4.in 4.2
```

```
Command: zexprsm -ly test4.in
-- popen (/usr/bin/cpp test4.in), fileno(yyin) = 4
Starting parse
Entering state 0
Reducing stack by rule 5 (line 46):
-> $$ = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>":)
Stack now 0
Entering state 2
Reading a token: --(end of buffer or a NUL)
--accepting rule at line 31 ("# 1 "test4.in"")
--included # 1 "test4.in"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "<built-in>"")
--included # 1 "<built-in>"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "<command-line>"")
--included # 1 "<command-line>"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "/usr/include/stdc-predef.h" 1 3 4")
--included # 1 "/usr/include/stdc-predef.h"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "<command-line>" 2")
--included # 1 "<command-line>"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "test4.in"")
--included # 1 "test4.in"
--accepting rule at line 33 ("
")
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "test4a.inh" 1")
--included # 1 "test4a.inh"
--accepting rule at line 33 ("
")
--accepting rule at line 33 ("
")
--accepting rule at line 33 ("
")
--accepting rule at line 36 ("pi")
Next token is token IDENT (0x5a27a40->{IDENT 7.3.1 "pi":)
Shifting token IDENT (0x5a27a40->{IDENT 7.3.1 "pi":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a27a40->{IDENT 7.3.1 "pi":)
-> $$ = nterm expr (0x5a27a40->{IDENT 7.3.1 "pi":)
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 37 ("=")
Next token is token ’=’ (0x5a27bf0->{’=’ 7.3.3 "=":)
Shifting token ’=’ (0x5a27bf0->{’=’ 7.3.3 "=":)
Entering state 16
Reading a token: --accepting rule at line 35 ("3.141592653589793238462643383280")
```

```
Next token is token NUMBER (0x5a27d30->{NUMBER 7.3.4 "3.141592653589793238462643383280":)
Shifting token NUMBER (0x5a27d30->{NUMBER 7.3.4 "3.141592653589793238462643383280":)
Entering state 6
Reducing stack by rule 16 (line 59):
   $1 = token NUMBER (0x5a27d30->{NUMBER 7.3.4 "3.141592653589793238462643383280":)
-> $$ = nterm expr (0x5a27d30->{NUMBER 7.3.4 "3.141592653589793238462643383280":)
Stack now 0 2 11 16
Entering state 24
Reading a token: --accepting rule at line 45 (";")
Next token is token ';' (0x5a27e90->{';' 7.3.36 ";":)
Reducing stack by rule 6 (line 49):
   $1 = nterm expr (0x5a27a40->{IDENT 7.3.1 "pi":)
   $2 = token '=' (0x5a27bf0->{'=' 7.3.3 "=":)
   $3 = nterm expr (0x5a27d30->{NUMBER 7.3.4 "3.141592653589793238462643383280":)
-> $$ = nterm expr (0x5a27bf0->{'=' 7.3.3 "=": 0x5a27a40 0x5a27d30)
Stack now 0 2
Entering state 11
Next token is token ';' (0x5a27e90->{';' 7.3.36 ";":)
Shifting token ';' (0x5a27e90->{';' 7.3.36 ";":)
Entering state 22
Reducing stack by rule 2 (line 43):
   $1 = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>":)
   $2 = nterm expr (0x5a27bf0->{'=' 7.3.3 "=": 0x5a27a40 0x5a27d30)
   $3 = token ';' (0x5a27e90->{';' 7.3.36 ";":)
Deleting astree (0x5a27e90->{';' 7.3.36 ";":)
-> $$ = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0)
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 33 ("
")
--accepting rule at line 36 ("pi")
Next token is token IDENT (0x5a28160->{IDENT 7.4.1 "pi":)
Shifting token IDENT (0x5a28160->{IDENT 7.4.1 "pi":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a28160->{IDENT 7.4.1 "pi":)
-> $$ = nterm expr (0x5a28160->{IDENT 7.4.1 "pi":)
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 45 (";")
Next token is token ';' (0x5a28240->{';' 7.4.3 ";":)
Shifting token ';' (0x5a28240->{';' 7.4.3 ";":)
Entering state 22
Reducing stack by rule 2 (line 43):
   $1 = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0)
   $2 = nterm expr (0x5a28160->{IDENT 7.4.1 "pi":)
   $3 = token ';' (0x5a28240->{';' 7.4.3 ";":)
Deleting astree (0x5a28240->{';' 7.4.3 ";":)
-> $$ = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0 0x5a28160)
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 3 "test4.in" 2")
--included # 3 "test4.in"
--accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 1 "test4b.inh" 1")
--included # 1 "test4b.inh"
--accepting rule at line 33 ("
")
--accepting rule at line 33 ("
")
--accepting rule at line 36 ("a")
Next token is token IDENT (0x5a28510->{IDENT 9.2.1 "a":)
Shifting token IDENT (0x5a28510->{IDENT 9.2.1 "a":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a28510->{IDENT 9.2.1 "a":)
-> $$ = nterm expr (0x5a28510->{IDENT 9.2.1 "a":)
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 37 ("=")
Next token is token ’=’ (0x5a28650->{’=’ 9.2.2 "=":)
Shifting token ’=’ (0x5a28650->{’=’ 9.2.2 "=":)
Entering state 16
Reading a token: --accepting rule at line 36 ("pi")
Next token is token IDENT (0x5a28730->{IDENT 9.2.3 "pi":)
Shifting token IDENT (0x5a28730->{IDENT 9.2.3 "pi":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a28730->{IDENT 9.2.3 "pi":)
-> $$ = nterm expr (0x5a28730->{IDENT 9.2.3 "pi":)
Stack now 0 2 11 16
Entering state 24
Reading a token: --accepting rule at line 40 ("*")
Next token is token ’*’ (0x5a28810->{’*’ 9.2.5 "*":)
Shifting token ’*’ (0x5a28810->{’*’ 9.2.5 "*":)
Entering state 19
Reading a token: --accepting rule at line 36 ("r")
Next token is token IDENT (0x5a28950->{IDENT 9.2.6 "r":)
Shifting token IDENT (0x5a28950->{IDENT 9.2.6 "r":)
Entering state 5
Reducing stack by rule 15 (line 58):
   $1 = token IDENT (0x5a28950->{IDENT 9.2.6 "r":)
-> $$ = nterm expr (0x5a28950->{IDENT 9.2.6 "r":)
Stack now 0 2 11 16 24 19
Entering state 27
Reading a token: --accepting rule at line 42 ("^")
Next token is token ’^’ (0x5a28a90->{’^’ 9.2.7 "^":)
Shifting token ’^’ (0x5a28a90->{’^’ 9.2.7 "^":)
Entering state 21
Reading a token: --accepting rule at line 35 ("2")
Next token is token NUMBER (0x5a28bd0->{NUMBER 9.2.8 "2":)
Shifting token NUMBER (0x5a28bd0->{NUMBER 9.2.8 "2":)
Entering state 6
Reducing stack by rule 16 (line 59):
   $1 = token NUMBER (0x5a28bd0->{NUMBER 9.2.8 "2":)
-> $$ = nterm expr (0x5a28bd0->{NUMBER 9.2.8 "2":)
Stack now 0 2 11 16 24 19 27 21
```

```
Entering state 29
Reading a token: --accepting rule at line 45 (";")
Next token is token ’;’ (0x5a28d10->{’;’ 9.2.9 ";":)
Reducing stack by rule 11 (line 54):
   $1 = nterm expr (0x5a28950->{IDENT 9.2.6 "r":)
   $2 = token ’^’ (0x5a28a90->{’^’ 9.2.7 "^":)
   $3 = nterm expr (0x5a28bd0->{NUMBER 9.2.8 "2":)
-> $$ = nterm expr (0x5a28a90->{’^’ 9.2.7 "^": 0x5a28950 0x5a28bd0)
Stack now 0 2 11 16 24 19
Entering state 27
Next token is token ’;’ (0x5a28d10->{’;’ 9.2.9 ";":)
Reducing stack by rule 9 (line 52):
   $1 = nterm expr (0x5a28730->{IDENT 9.2.3 "pi":)
   $2 = token ’*’ (0x5a28810->{’*’ 9.2.5 "*":)
   $3 = nterm expr (0x5a28a90->{’^’ 9.2.7 "^": 0x5a28950 0x5a28bd0)
-> $$ = nterm expr (0x5a28810->{’*’ 9.2.5 "*": 0x5a28730 0x5a28a90)
Stack now 0 2 11 16
Entering state 24
Next token is token ’;’ (0x5a28d10->{’;’ 9.2.9 ";":)
Reducing stack by rule 6 (line 49):
   $1 = nterm expr (0x5a28510->{IDENT 9.2.1 "a":)
   $2 = token ’=’ (0x5a28650->{’=’ 9.2.2 "=":)
   $3 = nterm expr (0x5a28810->{’*’ 9.2.5 "*": 0x5a28730 0x5a28a90)
-> $$ = nterm expr (0x5a28650->{’=’ 9.2.2 "=": 0x5a28510 0x5a28810)
Stack now 0 2
Entering state 11
Next token is token ’;’ (0x5a28d10->{’;’ 9.2.9 ";":)
Shifting token ’;’ (0x5a28d10->{’;’ 9.2.9 ";":)
Entering state 22
Reducing stack by rule 2 (line 43):
   $1 = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0 0x5a28160)
   $2 = nterm expr (0x5a28650->{’=’ 9.2.2 "=": 0x5a28510 0x5a28810)
   $3 = token ’;’ (0x5a28d10->{’;’ 9.2.9 ";":)
Deleting astree (0x5a28d10->{’;’ 9.2.9 ";":)
-> $$ = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0 0x5a28160 0x5a28650)
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 33 ("
")
--accepting rule at line 35 ("3.14159265358979323846264338328O")
Next token is token NUMBER (0x5a29030->{NUMBER 9.3.1 "3.14159265358979323846264338328O":)
Shifting token NUMBER (0x5a29030->{NUMBER 9.3.1 "3.14159265358979323846264338328O":)
Entering state 6
Reducing stack by rule 16 (line 59):
   $1 = token NUMBER (0x5a29030->{NUMBER 9.3.1 "3.14159265358979323846264338328O":)
-> $$ = nterm expr (0x5a29030->{NUMBER 9.3.1 "3.14159265358979323846264338328O":)
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 45 (";")
Next token is token ’;’ (0x5a29130->{’;’ 9.3.33 ";":)
Shifting token ’;’ (0x5a29130->{’;’ 9.3.33 ";":)
Entering state 22
```

```
Reducing stack by rule 2 (line 43):
   $1 = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0 0x5a28160 0x5a28650)
   $2 = nterm expr (0x5a29030->{NUMBER 9.3.1 "3.14159265358979323846264338328 0":)
   $3 = token ’;’ (0x5a29130->{’;’ 9.3.33 ";":)
Deleting astree (0x5a29130->{’;’ 9.3.33 ";":)
-> $$ = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0 0x5a28160 0x5a28650 0x5a29030)
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 33 ("
")
--accepting rule at line 31 ("# 4 "test4.in" 2")
--included # 4 "test4.in"
--accepting rule at line 33 ("
")
--accepting rule at line 35 ("3")
Next token is token NUMBER (0x5a29280->{NUMBER 10.4.1 "3":)
Shifting token NUMBER (0x5a29280->{NUMBER 10.4.1 "3":)
Entering state 6
Reducing stack by rule 16 (line 59):
   $1 = token NUMBER (0x5a29280->{NUMBER 10.4.1 "3":)
-> $$ = nterm expr (0x5a29280->{NUMBER 10.4.1 "3":)
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 40 ("*")
Next token is token ’*’ (0x5a294c0->{’*’ 10.4.2 "*":)
Shifting token ’*’ (0x5a294c0->{’*’ 10.4.2 "*":)
Entering state 19
Reading a token: --accepting rule at line 35 ("4")
Next token is token NUMBER (0x5a295a0->{NUMBER 10.4.3 "4":)
Shifting token NUMBER (0x5a295a0->{NUMBER 10.4.3 "4":)
Entering state 6
Reducing stack by rule 16 (line 59):
   $1 = token NUMBER (0x5a295a0->{NUMBER 10.4.3 "4":)
-> $$ = nterm expr (0x5a295a0->{NUMBER 10.4.3 "4":)
Stack now 0 2 11 19
Entering state 27
Reading a token: --accepting rule at line 45 (";")
Next token is token ’;’ (0x5a296e0->{’;’ 10.4.4 ";":)
Reducing stack by rule 9 (line 52):
   $1 = nterm expr (0x5a29280->{NUMBER 10.4.1 "3":)
   $2 = token ’*’ (0x5a294c0->{’*’ 10.4.2 "*":)
   $3 = nterm expr (0x5a295a0->{NUMBER 10.4.3 "4":)
-> $$ = nterm expr (0x5a294c0->{’*’ 10.4.2 "*": 0x5a29280 0x5a295a0)
Stack now 0 2
Entering state 11
Next token is token ’;’ (0x5a296e0->{’;’ 10.4.4 ";":)
Shifting token ’;’ (0x5a296e0->{’;’ 10.4.4 ";":)
Entering state 22
Reducing stack by rule 2 (line 43):
   $1 = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0 0x5a28160 0x5a28650 0x5a29030)
   $2 = nterm expr (0x5a294c0->{’*’ 10.4.2 "*": 0x5a29280 0x5a295a0)
   $3 = token ’;’ (0x5a296e0->{’;’ 10.4.4 ";":)
Deleting astree (0x5a296e0->{’;’ 10.4.4 ";":)
-> $$ = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0 0x5a
28160 0x5a28650 0x5a29030 0x5a294c0)
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 33 ("
")
--(end of buffer or a NUL)
--EOF (start condition 0)
Now at end of input.
Reducing stack by rule 1 (line 40):
   $1 = nterm stmtseq (0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0 0x5a28160 0x5a28650 0x5a29030 0x5a294c0)
-> $$ = nterm program (nullptr)
Stack now 0
Entering state 1
Now at end of input.
Shifting token $end ()
Entering state 3
Stack now 0 1 3
Cleanup: popping token $end ()
Cleanup: popping nterm program (nullptr)
Dumping parser::root:
0x5a23320->{ROOT 0.0.0 "<<ROOT>>": 0x5a27bf0 0x5a28160 0x5a28650 0x5a29030 0x5a294c0
   0x5a27bf0->{'=' 7.3.3 "=": 0x5a27a40 0x5a27d30
      0x5a27a40->{IDENT 7.3.1 "pi":
      0x5a27d30->{NUMBER 7.3.4 "3.141592653589793238462643383280":
   0x5a28160->{IDENT 7.4.1 "pi":
   0x5a28650->{'=' 9.2.2 "=": 0x5a28510 0x5a28810
      0x5a28510->{IDENT 9.2.1 "a":
      0x5a28810->{'*' 9.2.5 "*": 0x5a28730 0x5a28a90
         0x5a28730->{IDENT 9.2.3 "pi":
         0x5a28a90->{'^' 9.2.7 "^": 0x5a28950 0x5a28bd0
            0x5a28950->{IDENT 9.2.6 "r":
            0x5a28bd0->{NUMBER 9.2.8 "2":
   0x5a29030->{NUMBER 9.3.1 "3.141592653589793238462643383280":
   0x5a294c0->{'*' 10.4.2 "*": 0x5a29280 0x5a295a0
      0x5a29280->{NUMBER 10.4.1 "3":
      0x5a295a0->{NUMBER 10.4.3 "4":
Dumping string_set:
string_set[   3]:    8248777770799913213 0x5a29368->"3"
                     6637313742931709005 0x5a288f8->"*"
                    11597697714117577063 0x5a27cd8->"="
                     5344662657899890615 0x5a27f78->";"
string_set[   6]:    3729804957429652673 0x5a23418->"<<ROOT>>"
string_set[   9]:    4551451650890805270 0x5a28cb8->"2"
string_set[  11]:    1998636459596678225 0x5a28b78->"^"
string_set[  12]:   15530967369654070964 0x5a27b28->"pi"
                    14518329498511883088 0x5a28a38->"r"
string_set[  17]:    4993892634952068459 0x5a285f8->"a"
string_set[  20]:   17027087115628340017 0x5a27e38->"3.141592653589793238462643383280"
string_set[  21]:   16215888864653804456 0x5a29688->"4"
load_factor = 0.522
bucket_count = 23
max_bucket_size = 4
Deleting astree (0x5a295a0->{NUMBER 10.4.3 "4":)
Deleting astree (0x5a29280->{NUMBER 10.4.1 "3":)
Deleting astree (0x5a294c0->{'*' 10.4.2 "*":)
Deleting astree (0x5a29030->{NUMBER 9.3.1 "3.141592653589793238462643383280":)
Deleting astree (0x5a28bd0->{NUMBER 9.2.8 "2":)
Deleting astree (0x5a28950->{IDENT 9.2.6 "r":)
Deleting astree (0x5a28a90->{'^' 9.2.7 "^":)
Deleting astree (0x5a28730->{IDENT 9.2.3 "pi":)
Deleting astree (0x5a28810->{'*' 9.2.5 "*":)
Deleting astree (0x5a28510->{IDENT 9.2.1 "a":)
Deleting astree (0x5a28650->{'=' 9.2.2 "=":)
Deleting astree (0x5a28160->{IDENT 7.4.1 "pi":)
Deleting astree (0x5a27d30->{NUMBER 7.3.4 "3.141592653589793238462643383280":)
Deleting astree (0x5a27a40->{IDENT 7.3.1 "pi":)
Deleting astree (0x5a27bf0->{'=' 7.3.3 "=":)
Deleting astree (0x5a23320->{ROOT 0.0.0 "<<ROOT>>":)
```

```
==11305== Memcheck, a memory error detector
==11305== Copyright (C) 2002-2017, and GNU GPL’d, by Julian Seward et al.
==11305== Using Valgrind-3.14.0.GIT and LibVEX; rerun with -h for copyright info
==11305== Command: zexprsm -ly test4.in
==11305== Parent PID: 11304
==11305==
==11305==
==11305== HEAP SUMMARY:
==11305==     in use at exit: 0 bytes in 0 blocks
==11305==   total heap usage: 94 allocs, 94 frees, 20,272 bytes allocated
==11305==
==11305== All heap blocks were freed -- no leaks are possible
==11305==
==11305== For counts of detected and suppressed errors, rerun with: -v
==11305== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
EXIT STATUS = 0
```