```
flex version 2.5.35 usage statistics:
  scanner options: -dsvI8 -Cem -oyylex.cc
  106/2000 NFA states
  29/1000 DFA states (158 words)
  16 rules
  Compressed tables always back-up
  1/40 start conditions
  73 epsilon states, 39 double epsilon states
  18/100 character classes needed 197/500 words of storage, 0 reused
  92 state/nextstate pairs created
  57/35 unique/duplicate transitions
  33/1000 base-def entries created
  59/2000 (peak 77) nxt-chk entries created
  20/2500 (peak 68) template nxt-chk entries created
  0 empty table entries
  5 protos created
  4 templates created, 5 uses
  17/256 equivalence classes created
  5/256 meta-equivalence classes created
  0 (2 saved) hash collisions, 31 DFAs equal
  0 sets of reallocations needed
  457 total table entries needed
```

```
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
NEG (261)
POS (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)
```

```
astree.o:
     $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
     $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
     $Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
     $Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
     $Id: astree.cc,v 1.14 2013-10-10 18:48:18-07 - - $
     $Compiled: astree.cc Apr 16 2015 17:21:13 $

emit.o:
     $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
     $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
     $Id: emit.h,v 1.1 2013-09-19 16:38:25-07 - - $
     $Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
     $Id: emit.cc,v 1.3 2013-09-20 17:52:13-07 - - $
     $Compiled: emit.cc Apr 16 2015 17:21:13 $

lyutils.o:
     $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
     $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
     $Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
     $Id: lyutils.cc,v 1.3 2013-10-10 18:17:45-07 - - $
     $Compiled: lyutils.cc Apr 16 2015 17:21:13 $

auxlib.o:
     $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
     $Id: auxlib.cc,v 1.3 2013-09-20 17:52:13-07 - - $
     $Compiled: auxlib.cc Apr 16 2015 17:21:14 $

stringset.o:
     $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
     $Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
     $Id: stringset.cc,v 1.7 2015-04-16 17:19:22-07 - - $
     $Compiled: stringset.cc Apr 16 2015 17:21:14 $

main.o:
     $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
     $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
     $Id: emit.h,v 1.1 2013-09-19 16:38:25-07 - - $
     $Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
     $Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
     $Id: main.cc,v 1.4 2013-09-20 17:52:13-07 - - $
     $Compiled: main.cc Apr 16 2015 17:21:14 $

yylex.o:
     $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
     $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
     $Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
     $Id: scanner.l,v 1.5 2015-04-16 17:17:51-07 - - $
     $Compiled: scanner.l Apr 16 2015 17:21:14 $

yyparse.o:
     $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
     $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
     $Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
     $Id: parser.y,v 1.6 2015-04-16 17:17:22-07 - - $
     $Compiled: parser.y Apr 16 2015 17:21:15 $

zexprsm:
```

```
$Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
$Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
$Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
$Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
$Id: astree.cc,v 1.14 2013-10-10 18:48:18-07 - - $
$Compiled: astree.cc Apr 16 2015 17:21:13 $
$Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
$Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
$Id: emit.h,v 1.1 2013-09-19 16:38:25-07 - - $
$Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
$Id: emit.cc,v 1.3 2013-09-20 17:52:13-07 - - $
$Compiled: emit.cc Apr 16 2015 17:21:13 $
$Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
$Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
$Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
$Id: lyutils.cc,v 1.3 2013-10-10 18:17:45-07 - - $
$Compiled: lyutils.cc Apr 16 2015 17:21:13 $
$Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
$Id: auxlib.cc,v 1.3 2013-09-20 17:52:13-07 - - $
$Compiled: auxlib.cc Apr 16 2015 17:21:14 $
$Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
$Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
$Id: stringset.cc,v 1.7 2015-04-16 17:19:22-07 - - $
$Compiled: stringset.cc Apr 16 2015 17:21:14 $
$Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
$Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
$Id: emit.h,v 1.1 2013-09-19 16:38:25-07 - - $
$Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
$Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
$Id: main.cc,v 1.4 2013-09-20 17:52:13-07 - - $
$Compiled: main.cc Apr 16 2015 17:21:14 $
$Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
$Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
$Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
$Id: scanner.l,v 1.5 2015-04-16 17:17:51-07 - - $
$Compiled: scanner.l Apr 16 2015 17:21:14 $
$Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
$Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
$Id: lyutils.h,v 1.6 2015-04-16 17:19:58-07 - - $
$Id: parser.y,v 1.6 2015-04-16 17:17:22-07 - - $
$Compiled: parser.y Apr 16 2015 17:21:15 $
```

$cmps104a-wm/Examples/e08.expr-smc/
test1.in

```
// $Id: test1.in,v 1.1 2013-09-19 16:38:25-07 - - $
a=b*c+d*e;
```

```
;    1: # 1 "test1.in";# 1 "test1.in"
;    0:
;    1: # 1 "<built-in>";# 1 "<built-in>"
;    0:
;    1: # 1 "<command-line>";# 1 "<command-line>"
;    0:
;    1: # 1 "test1.in";# 1 "test1.in"
;    0:
;    1:
;    2: a=b*c+d*e;

          pushvar   b                   ; test1.in 2.2
          pushvar   c                   ; test1.in 2.4
          mul                           ; test1.in 2.3
          pushvar   d                   ; test1.in 2.6
          pushvar   e                   ; test1.in 2.8
          mul                           ; test1.in 2.7
          add                           ; test1.in 2.5
          popvar    a                   ; test1.in 2.0
```

```
==8933== Memcheck, a memory error detector
==8933== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==8933== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright info
==8933== Command: zexprsm -ly -@@ test1.in
==8933==
DEBUGF(x): auxlib.cc[112] set_debugflags():
Debugflags = "@", all = 1
DEBUGF(m): main.cc[69] scan_opts():
filename = test1.in, yyin = 0x4c2e1d0, fileno (yyin) = 3
Starting parse
Entering state 0
Reducing stack by rule 5 (line 47):
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c2e3d0->{0:0.0: ROOT: "<<ROOT>>"}
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --(end of buffer or a NUL)
--accepting rule at line 29 ("# 1 "test1.in"")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=test1.in, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 1 "<built-in>"")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=<built-in>, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 1 "<command-line>"")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=<command-line>, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 1 "test1.in"")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=test1.in, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 31 ("
")
--accepting rule at line 34 ("a")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32930->{4:2.0: IDENT: "a"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32930
0x4c32930->{IDENT(259) 4:2.000 "a" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32930
0x4c32930->{IDENT(259) 4:2.000 "a" []})
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32930
0x4c32930->{IDENT(259) 4:2.000 "a" []})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
```

```
Reading a token: --accepting rule at line 36 ("=")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32a70->{4:2.1: '=': "="}
Next token is token '=' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 61, yyvaluep = 0x4c32a70
0x4c32a70->{'='(61) 4:2.001 "=" []})
Shifting token '=' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 61, yyvaluep = 0x4c32a70
0x4c32a70->{'='(61) 4:2.001 "=" []})
Entering state 16
Reading a token: --accepting rule at line 34 ("b")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32bb0->{4:2.2: IDENT: "b"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32bb0
0x4c32bb0->{IDENT(259) 4:2.002 "b" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32bb0
0x4c32bb0->{IDENT(259) 4:2.002 "b" []})
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32bb0
0x4c32bb0->{IDENT(259) 4:2.002 "b" []})
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Reading a token: --accepting rule at line 39 ("*")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32cf0->{4:2.3: '*': "*"}
Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c32cf0
0x4c32cf0->{'*'(42) 4:2.003 "*" []})
Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c32cf0
0x4c32cf0->{'*'(42) 4:2.003 "*" []})
Entering state 19
Reading a token: --accepting rule at line 34 ("c")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32e30->{4:2.4: IDENT: "c"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32e30
0x4c32e30->{IDENT(259) 4:2.004 "c" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32e30
0x4c32e30->{IDENT(259) 4:2.004 "c" []})
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32e30
0x4c32e30->{IDENT(259) 4:2.004 "c" []})
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 19
Entering state 27
Reading a token: --accepting rule at line 37 ("+")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32f70->{4:2.5: '+': "+"}
Next token is token '+' (DEBUGF(f): astree.cc[79] yyprint():
```

```
toknum = 43, yyvaluep = 0x4c32f70
0x4c32f70->{’+’(43) 4:2.005 "+" []})
Reducing stack by rule 9 (line 53):
   $1 = nterm expr ()
   $2 = token ’*’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c32cf0
0x4c32cf0->{’*’(42) 4:2.003 "*" []})
   $3 = nterm expr ()
DEBUGF(a): astree.cc[32] adopt1():
0x4c32cf0 (*) adopting 0x4c32bb0 (b)
DEBUGF(a): astree.cc[32] adopt1():
0x4c32cf0 (*) adopting 0x4c32e30 (c)
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Next token is token ’+’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 43, yyvaluep = 0x4c32f70
0x4c32f70->{’+’(43) 4:2.005 "+" []})
Shifting token ’+’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 43, yyvaluep = 0x4c32f70
0x4c32f70->{’+’(43) 4:2.005 "+" []})
Entering state 17
Reading a token: --accepting rule at line 34 ("d")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33150->{4:2.6: IDENT: "d"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33150
0x4c33150->{IDENT(259) 4:2.006 "d" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33150
0x4c33150->{IDENT(259) 4:2.006 "d" []})
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33150
0x4c33150->{IDENT(259) 4:2.006 "d" []})
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 17
Entering state 25
Reading a token: --accepting rule at line 39 ("*")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33290->{4:2.7: ’*’: "*"}
Next token is token ’*’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c33290
0x4c33290->{’*’(42) 4:2.007 "*" []})
Shifting token ’*’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c33290
0x4c33290->{’*’(42) 4:2.007 "*" []})
Entering state 19
Reading a token: --accepting rule at line 34 ("e")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33370->{4:2.8: IDENT: "e"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33370
0x4c33370->{IDENT(259) 4:2.008 "e" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33370
0x4c33370->{IDENT(259) 4:2.008 "e" []})
```

```
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33370
0x4c33370->{IDENT(259) 4:2.008 "e" []})
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 17 25 19
Entering state 27
Reading a token: --accepting rule at line 44 (";")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c334b0->{4:2.9: ';': ";"}
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c334b0
0x4c334b0->{';'(59) 4:2.009 ";" []})
Reducing stack by rule 9 (line 53):
   $1 = nterm expr ()
   $2 = token '*' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c33290
0x4c33290->{'*'(42) 4:2.007 "*" []})
   $3 = nterm expr ()
DEBUGF(a): astree.cc[32] adopt1():
0x4c33290 (*) adopting 0x4c33150 (d)
DEBUGF(a): astree.cc[32] adopt1():
0x4c33290 (*) adopting 0x4c33370 (e)
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 17
Entering state 25
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c334b0
0x4c334b0->{';'(59) 4:2.009 ";" []})
Reducing stack by rule 7 (line 51):
   $1 = nterm expr ()
   $2 = token '+' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 43, yyvaluep = 0x4c32f70
0x4c32f70->{'+'(43) 4:2.005 "+" []})
   $3 = nterm expr ()
DEBUGF(a): astree.cc[32] adopt1():
0x4c32f70 (+) adopting 0x4c32cf0 (*)
DEBUGF(a): astree.cc[32] adopt1():
0x4c32f70 (+) adopting 0x4c33290 (*)
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c334b0
0x4c334b0->{';'(59) 4:2.009 ";" []})
Reducing stack by rule 6 (line 50):
   $1 = nterm expr ()
   $2 = token '=' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 61, yyvaluep = 0x4c32a70
0x4c32a70->{'='(61) 4:2.001 "=" []})
   $3 = nterm expr ()
DEBUGF(a): astree.cc[32] adopt1():
0x4c32a70 (=) adopting 0x4c32930 (a)
DEBUGF(a): astree.cc[32] adopt1():
0x4c32a70 (=) adopting 0x4c32f70 (+)
-> $$ = nterm expr ()
Stack now 0 2
```

```
Entering state 11
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c334b0
0x4c334b0->{';'(59) 4:2.009 ";" []})
Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c334b0
0x4c334b0->{';'(59) 4:2.009 ";" []})
Entering state 22
Reducing stack by rule 2 (line 44):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c334b0
0x4c334b0->{';'(59) 4:2.009 ";" []})
DEBUGF(f): astree.cc[97] free_ast():
free [4C334B0]-> 4:2.9: ';': ";")
DEBUGF(a): astree.cc[32] adopt1():
0x4c2e3d0 (<<ROOT>>) adopting 0x4c32a70 (=)
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 31 ("
")
--(end of buffer or a NUL)
--EOF (start condition 0)
Now at end of input.
Reducing stack by rule 1 (line 41):
   $1 = nterm stmtseq ()
-> $$ = nterm program ()
Stack now 0
Entering state 1
Now at end of input.
Shifting token $end (DEBUGF(f): astree.cc[79] yyprint():
toknum = 0, yyvaluep = 0x4c334b0
$end(0)
)
Entering state 3
Stack now 0 1 3
Cleanup: popping token $end (DEBUGF(f): astree.cc[79] yyprint():
toknum = 0, yyvaluep = 0x4c334b0
$end(0)
)
Cleanup: popping nterm program ()
DEBUGF(a): main.cc[87] main():

<<ROOT>> 0x4c2e3d0->{ROOT(258) 0:0.000 "<<ROOT>>" [0x4c32a70]}
   = 0x4c32a70->{'='(61) 4:2.001 "=" [0x4c32930 0x4c32f70]}
      a 0x4c32930->{IDENT(259) 4:2.000 "a" []}
      + 0x4c32f70->{'+'(43) 4:2.005 "+" [0x4c32cf0 0x4c33290]}
         * 0x4c32cf0->{'*'(42) 4:2.003 "*" [0x4c32bb0 0x4c32e30]}
            b 0x4c32bb0->{IDENT(259) 4:2.002 "b" []}
            c 0x4c32e30->{IDENT(259) 4:2.004 "c" []}
         * 0x4c33290->{'*'(42) 4:2.007 "*" [0x4c33150 0x4c33370]}
            d 0x4c33150->{IDENT(259) 4:2.006 "d" []}
            e 0x4c33370->{IDENT(259) 4:2.008 "e" []}
DEBUGF(f): astree.cc[97] free_ast():
free [4C33370]-> 4:2.8: IDENT: "e")
DEBUGF(f): astree.cc[97] free_ast():
```

```
free [4C33150]-> 4:2.6: IDENT: "d")
DEBUGF(f): astree.cc[97] free_ast():
free [4C33290]-> 4:2.7: '*': "*")
DEBUGF(f): astree.cc[97] free_ast():
free [4C32E30]-> 4:2.4: IDENT: "c")
DEBUGF(f): astree.cc[97] free_ast():
free [4C32BB0]-> 4:2.2: IDENT: "b")
DEBUGF(f): astree.cc[97] free_ast():
free [4C32CF0]-> 4:2.3: '*': "*")
DEBUGF(f): astree.cc[97] free_ast():
free [4C32F70]-> 4:2.5: '+': "+")
DEBUGF(f): astree.cc[97] free_ast():
free [4C32930]-> 4:2.0: IDENT: "a")
DEBUGF(f): astree.cc[97] free_ast():
free [4C32A70]-> 4:2.1: '=': "=")
DEBUGF(f): astree.cc[97] free_ast():
free [4C2E3D0]-> 0:0.0: ROOT: "<<ROOT>>")
DEBUGF(s): main.cc[92] main():

stringset[   0]:   10959529184379665549 0x4c32f18->"c"
                    3729804957429652673 0x4c2e4c8->"<<ROOT>>"
stringset[   1]:    2540012008095083820 0x4c33458->"e"
                   10838281452030117757 0x4c32c98->"b"
                   11597697714117577063 0x4c32b58->"="
stringset[   3]:    1370730858159036685 0x4c33058->"+"
                    6637313742931709005 0x4c32dd8->"*"
stringset[   4]:    5344662657899890615 0x4c33598->";"
stringset[   5]:   14494284460613645429 0x4c33238->"d"
stringset[   7]:    4993892634952068459 0x4c32a18->"a"
load_factor = 0.909
bucket_count = 11
max_bucket_size = 3
==8933==
==8933== HEAP SUMMARY:
==8933==     in use at exit: 0 bytes in 0 blocks
==8933==   total heap usage: 57 allocs, 57 frees, 18,522 bytes allocated
==8933==
==8933== All heap blocks were freed -- no leaks are possible
==8933==
==8933== For counts of detected and suppressed errors, rerun with: -v
==8933== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
EXIT STATUS 0
```

```
 t// $Id: test4.in,v 1.1 2013-09-19 16:38:25-07 - - $
#include "test4a.inh"
#include "test4b.inh"
3*4;
```

```
;    1: # 1 "test4.in";# 1 "test4.in"
;    0:
;    1: # 1 "<built-in>";# 1 "<built-in>"
;    0:
;    1: # 1 "<command-line>";# 1 "<command-line>"
;    0:
;    1: # 1 "test4.in";# 1 "test4.in"
;    0:
;    1:  t
;    2: # 1 "test4a.inh" 1;# 1 "test4a.inh"
;    0:
;    1:
;    2:
;    3: pi=3.141592653589793238462643383280;
;    4: pi;
;    5: # 3 "test4.in" 2;# 3 "test4.in"
;    2:
;    3: # 1 "test4b.inh" 1;# 1 "test4b.inh"
;    0:
;    1:
;    2: a=pi*r^2;
;    3: 3.141592653589793238462643383280;
;    4: # 4 "test4.in" 2;# 4 "test4.in"
;    3:
;    4: 3*4;

          pushvar   pi                  ; test4a.inh 4.0
          pushvar   pi                  ; test4b.inh 2.2
          pushvar   r                   ; test4b.inh 2.5
          pushnum   2                   ; test4b.inh 2.7
          pow                           ; test4b.inh 2.6
          mul                           ; test4b.inh 2.4
          popvar    a                   ; test4b.inh 2.0
          pushnum   3.141592653589793238462643383280; test4b.inh 3.0
          pushnum   3                   ; test4.in 4.0
          pushnum   4                   ; test4.in 4.2
          mul                           ; test4.in 4.1
```

```
==8984== Memcheck, a memory error detector
==8984== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==8984== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright info
==8984== Command: zexprsm -ly -@@ test4.in
==8984==
DEBUGF(x): auxlib.cc[112] set_debugflags():
Debugflags = "@", all = 1
DEBUGF(m): main.cc[69] scan_opts():
filename = test4.in, yyin = 0x4c2e1d0, fileno (yyin) = 3
Starting parse
Entering state 0
Reducing stack by rule 5 (line 47):
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c2e3d0->{0:0.0: ROOT: "<<ROOT>>"}
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --(end of buffer or a NUL)
--accepting rule at line 29 ("# 1 "test4.in"")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=test4.in, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 1 "<built-in>"")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=<built-in>, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 1 "<command-line>"")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=<command-line>, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 1 "test4.in"")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=test4.in, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 30 (" ")
--accepting rule at line 34 ("t")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32930->{4:1.1: IDENT: "t"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32930
0x4c32930->{IDENT(259) 4:1.001 "t" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32930
0x4c32930->{IDENT(259) 4:1.001 "t" []})
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32930
0x4c32930->{IDENT(259) 4:1.001 "t" []})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 1 "test4a.inh" 1")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=test4a.inh, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 31 ("
")
--accepting rule at line 31 ("
")
--accepting rule at line 34 ("pi")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32ae0->{5:3.0: IDENT: "pi"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32ae0
0x4c32ae0->{IDENT(259) 5:3.000 "pi" []})
zexprsm: test4a.inh: 3: syntax error, unexpected IDENT
Error: popping nterm expr ()
DEBUGF(a): lyutils.cc[74] error_destructor():

t 0x4c32930->{IDENT(259) 4:1.001 "t" []}
DEBUGF(f): astree.cc[97] free_ast():
free [4C32930]-> 4:1.1: IDENT: "t")
Stack now 0 2
Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
toknum = 256, yyvaluep = 0x4c32ae0
error(256)
)
Entering state 4
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32ae0
0x4c32ae0->{IDENT(259) 5:3.000 "pi" []})
Error: discarding token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c32ae0
0x4c32ae0->{IDENT(259) 5:3.000 "pi" []})
DEBUGF(a): lyutils.cc[74] error_destructor():

pi 0x4c32ae0->{IDENT(259) 5:3.000 "pi" []}
DEBUGF(f): astree.cc[97] free_ast():
free [4C32AE0]-> 5:3.0: IDENT: "pi")
Error: popping token error (DEBUGF(f): astree.cc[79] yyprint():
toknum = 256, yyvaluep = 0x4c32ae0
error(256)
)
Stack now 0 2
Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
toknum = 256, yyvaluep = 0x4c32ae0
error(256)
)
Entering state 4
Reading a token: --accepting rule at line 36 ("=")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32c20->{5:3.2: '=': "="}
Next token is token '=' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 61, yyvaluep = 0x4c32c20
0x4c32c20->{'='(61) 5:3.002 "=" []})
Error: discarding token '=' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 61, yyvaluep = 0x4c32c20
```

```
0x4c32c20->{'='(61) 5:3.002 "=" []})
DEBUGF(a): lyutils.cc[74] error_destructor():

= 0x4c32c20->{'='(61) 5:3.002 "=" []}
DEBUGF(f): astree.cc[97] free_ast():
free [4C32C20]-> 5:3.2: '=': "=")
Error: popping token error (DEBUGF(f): astree.cc[79] yyprint():
toknum = 256, yyvaluep = 0x4c32ae0
error(256)
)
Stack now 0 2
Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
toknum = 256, yyvaluep = 0x4c32c20
error(256)
)
Entering state 4
Reading a token: --accepting rule at line 33 ("3.141592653589793238462643383280")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32d60->{5:3.3: NUMBER: "3.141592653589793238462643383280"}
Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c32d60
0x4c32d60->{NUMBER(260) 5:3.003 "3.141592653589793238462643383280" []})
Error: discarding token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c32d60
0x4c32d60->{NUMBER(260) 5:3.003 "3.141592653589793238462643383280" []})
DEBUGF(a): lyutils.cc[74] error_destructor():

3.141592653589793238462643383280 0x4c32d60->{NUMBER(260) 5:3.003 "3.141592653589793238462643383280" []}
DEBUGF(f): astree.cc[97] free_ast():
free [4C32D60]-> 5:3.3: NUMBER: "3.141592653589793238462643383280")
Error: popping token error (DEBUGF(f): astree.cc[79] yyprint():
toknum = 256, yyvaluep = 0x4c32c20
error(256)
)
Stack now 0 2
Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
toknum = 256, yyvaluep = 0x4c32d60
error(256)
)
Entering state 4
Reading a token: --accepting rule at line 44 (";")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c32ec0->{5:3.35: ';': ";"}
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c32ec0
0x4c32ec0->{';'(59) 5:3.035 ";" []})
Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c32ec0
0x4c32ec0->{';'(59) 5:3.035 ";" []})
Entering state 12
Reducing stack by rule 3 (line 45):
   $1 = nterm stmtseq ()
   $2 = token error (DEBUGF(f): astree.cc[79] yyprint():
toknum = 256, yyvaluep = 0x4c32d60
error(256)
)
```

```
   $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c32ec0
0x4c32ec0->{';'(59) 5:3.035 ";" []})
DEBUGF(f): astree.cc[97] free_ast():
free [4C32EC0]-> 5:3.35: ';': ";")
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 31 ("
")
--accepting rule at line 34 ("pi")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33000->{5:4.0: IDENT: "pi"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33000
0x4c33000->{IDENT(259) 5:4.000 "pi" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33000
0x4c33000->{IDENT(259) 5:4.000 "pi" []})
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33000
0x4c33000->{IDENT(259) 5:4.000 "pi" []})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 44 (";")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c330e0->{5:4.2: ';': ";"}
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c330e0
0x4c330e0->{';'(59) 5:4.002 ";" []})
Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c330e0
0x4c330e0->{';'(59) 5:4.002 ";" []})
Entering state 22
Reducing stack by rule 2 (line 44):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c330e0
0x4c330e0->{';'(59) 5:4.002 ";" []})
DEBUGF(f): astree.cc[97] free_ast():
free [4C330E0]-> 5:4.2: ';': ";")
DEBUGF(a): astree.cc[32] adopt1():
0x4c2e3d0 (<<ROOT>>) adopting 0x4c33000 (pi)
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 3 "test4.in" 2")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=test4.in, scan_linenr=2
--accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 1 "test4b.inh" 1")
```

```
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=test4b.inh, scan_linenr=0
--accepting rule at line 31 ("
")
--accepting rule at line 31 ("
")
--accepting rule at line 34 ("a")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c332f0->{7:2.0: IDENT: "a"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c332f0
0x4c332f0->{IDENT(259) 7:2.000 "a" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c332f0
0x4c332f0->{IDENT(259) 7:2.000 "a" []})
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c332f0
0x4c332f0->{IDENT(259) 7:2.000 "a" []})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 36 ("=")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33430->{7:2.1: '=': "="}
Next token is token '=' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 61, yyvaluep = 0x4c33430
0x4c33430->{'='(61) 7:2.001 "=" []})
Shifting token '=' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 61, yyvaluep = 0x4c33430
0x4c33430->{'='(61) 7:2.001 "=" []})
Entering state 16
Reading a token: --accepting rule at line 34 ("pi")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33510->{7:2.2: IDENT: "pi"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33510
0x4c33510->{IDENT(259) 7:2.002 "pi" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33510
0x4c33510->{IDENT(259) 7:2.002 "pi" []})
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33510
0x4c33510->{IDENT(259) 7:2.002 "pi" []})
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Reading a token: --accepting rule at line 39 ("*")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c335f0->{7:2.4: '*': "*"}
Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c335f0
0x4c335f0->{'*'(42) 7:2.004 "*" []})
Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c335f0
0x4c335f0->{’*’(42) 7:2.004 "*" []})
Entering state 19
Reading a token: --accepting rule at line 34 ("r")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33730->{7:2.5: IDENT: "r"}
Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33730
0x4c33730->{IDENT(259) 7:2.005 "r" []})
Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33730
0x4c33730->{IDENT(259) 7:2.005 "r" []})
Entering state 5
Reducing stack by rule 15 (line 59):
   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
toknum = 259, yyvaluep = 0x4c33730
0x4c33730->{IDENT(259) 7:2.005 "r" []})
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 19
Entering state 27
Reading a token: --accepting rule at line 41 ("^")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33870->{7:2.6: ’^’: "^"}
Next token is token ’^’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 94, yyvaluep = 0x4c33870
0x4c33870->{’^’(94) 7:2.006 "^" []})
Shifting token ’^’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 94, yyvaluep = 0x4c33870
0x4c33870->{’^’(94) 7:2.006 "^" []})
Entering state 21
Reading a token: --accepting rule at line 33 ("2")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c339b0->{7:2.7: NUMBER: "2"}
Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c339b0
0x4c339b0->{NUMBER(260) 7:2.007 "2" []})
Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c339b0
0x4c339b0->{NUMBER(260) 7:2.007 "2" []})
Entering state 6
Reducing stack by rule 16 (line 60):
   $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c339b0
0x4c339b0->{NUMBER(260) 7:2.007 "2" []})
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 19 27 21
Entering state 29
Reading a token: --accepting rule at line 44 (";")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33bf0->{7:2.8: ’;’: ";"}
Next token is token ’;’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c33bf0
0x4c33bf0->{’;’(59) 7:2.008 ";" []})
Reducing stack by rule 11 (line 55):
   $1 = nterm expr ()
   $2 = token ’^’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 94, yyvaluep = 0x4c33870
0x4c33870->{’^’(94) 7:2.006 "^" []})
   $3 = nterm expr ()
```

```
DEBUGF(a): astree.cc[32] adopt1():
0x4c33870 (^) adopting 0x4c33730 (r)
DEBUGF(a): astree.cc[32] adopt1():
0x4c33870 (^) adopting 0x4c339b0 (2)
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 19
Entering state 27
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c33bf0
0x4c33bf0->{';'(59) 7:2.008 ";" []})
Reducing stack by rule 9 (line 53):
   $1 = nterm expr ()
   $2 = token '*' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c335f0
0x4c335f0->{'*'(42) 7:2.004 "*" []})
   $3 = nterm expr ()
DEBUGF(a): astree.cc[32] adopt1():
0x4c335f0 (*) adopting 0x4c33510 (pi)
DEBUGF(a): astree.cc[32] adopt1():
0x4c335f0 (*) adopting 0x4c33870 (^)
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c33bf0
0x4c33bf0->{';'(59) 7:2.008 ";" []})
Reducing stack by rule 6 (line 50):
   $1 = nterm expr ()
   $2 = token '=' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 61, yyvaluep = 0x4c33430
0x4c33430->{'='(61) 7:2.001 "=" []})
   $3 = nterm expr ()
DEBUGF(a): astree.cc[32] adopt1():
0x4c33430 (=) adopting 0x4c332f0 (a)
DEBUGF(a): astree.cc[32] adopt1():
0x4c33430 (=) adopting 0x4c335f0 (*)
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c33bf0
0x4c33bf0->{';'(59) 7:2.008 ";" []})
Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c33bf0
0x4c33bf0->{';'(59) 7:2.008 ";" []})
Entering state 22
Reducing stack by rule 2 (line 44):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c33bf0
0x4c33bf0->{';'(59) 7:2.008 ";" []})
DEBUGF(f): astree.cc[97] free_ast():
free [4C33BF0]-> 7:2.8: ';': ";")
DEBUGF(a): astree.cc[32] adopt1():
0x4c2e3d0 (<<ROOT>>) adopting 0x4c33430 (=)
-> $$ = nterm stmtseq ()
Stack now 0
```

```
Entering state 2
Reading a token: --accepting rule at line 31 ("
")
--accepting rule at line 33 ("3.141592653589793238462643383280")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c33f00->{7:3.0: NUMBER: "3.141592653589793238462643383280"}
Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c33f00
0x4c33f00->{NUMBER(260) 7:3.000 "3.141592653589793238462643383280" []})
Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c33f00
0x4c33f00->{NUMBER(260) 7:3.000 "3.141592653589793238462643383280" []})
Entering state 6
Reducing stack by rule 16 (line 60):
   $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c33f00
0x4c33f00->{NUMBER(260) 7:3.000 "3.141592653589793238462643383280" []})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 44 (";")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c34000->{7:3.32: ';': ";"}
Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c34000
0x4c34000->{';'(59) 7:3.032 ";" []})
Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c34000
0x4c34000->{';'(59) 7:3.032 ";" []})
Entering state 22
Reducing stack by rule 2 (line 44):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c34000
0x4c34000->{';'(59) 7:3.032 ";" []})
DEBUGF(f): astree.cc[97] free_ast():
free [4C34000]-> 7:3.32: ';': ";")
DEBUGF(a): astree.cc[32] adopt1():
0x4c2e3d0 (<<ROOT>>) adopting 0x4c33f00 (3.141592653589793238462643383280)
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 31 ("
")
--accepting rule at line 29 ("# 4 "test4.in" 2")
DEBUGF(m): lyutils.cc[97] scanner_include():
filename=test4.in, scan_linenr=3
--accepting rule at line 31 ("
")
--accepting rule at line 33 ("3")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c34270->{8:4.0: NUMBER: "3"}
Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c34270
0x4c34270->{NUMBER(260) 8:4.000 "3" []})
Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
```

```
toknum = 260, yyvaluep = 0x4c34270
0x4c34270->{NUMBER(260) 8:4.000 "3" []})
Entering state 6
Reducing stack by rule 16 (line 60):
   $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c34270
0x4c34270->{NUMBER(260) 8:4.000 "3" []})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 39 ("*")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c343b0->{8:4.1: ’*’: "*"}
Next token is token ’*’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c343b0
0x4c343b0->{’*’(42) 8:4.001 "*" []})
Shifting token ’*’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c343b0
0x4c343b0->{’*’(42) 8:4.001 "*" []})
Entering state 19
Reading a token: --accepting rule at line 33 ("4")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c34490->{8:4.2: NUMBER: "4"}
Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c34490
0x4c34490->{NUMBER(260) 8:4.002 "4" []})
Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c34490
0x4c34490->{NUMBER(260) 8:4.002 "4" []})
Entering state 6
Reducing stack by rule 16 (line 60):
   $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
toknum = 260, yyvaluep = 0x4c34490
0x4c34490->{NUMBER(260) 8:4.002 "4" []})
-> $$ = nterm expr ()
Stack now 0 2 11 19
Entering state 27
Reading a token: --accepting rule at line 44 (";")
DEBUGF(f): astree.cc[23] new_astree():
astree 0x4c345d0->{8:4.3: ’;’: ";"}
Next token is token ’;’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c345d0
0x4c345d0->{’;’(59) 8:4.003 ";" []})
Reducing stack by rule 9 (line 53):
   $1 = nterm expr ()
   $2 = token ’*’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 42, yyvaluep = 0x4c343b0
0x4c343b0->{’*’(42) 8:4.001 "*" []})
   $3 = nterm expr ()
DEBUGF(a): astree.cc[32] adopt1():
0x4c343b0 (*) adopting 0x4c34270 (3)
DEBUGF(a): astree.cc[32] adopt1():
0x4c343b0 (*) adopting 0x4c34490 (4)
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Next token is token ’;’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c345d0
```

```
0x4c345d0->{’;’(59) 8:4.003 ";" []})
Shifting token ’;’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c345d0
0x4c345d0->{’;’(59) 8:4.003 ";" []})
Entering state 22
Reducing stack by rule 2 (line 44):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ’;’ (DEBUGF(f): astree.cc[79] yyprint():
toknum = 59, yyvaluep = 0x4c345d0
0x4c345d0->{’;’(59) 8:4.003 ";" []})
DEBUGF(f): astree.cc[97] free_ast():
free [4C345D0]-> 8:4.3: ’;’: ";")
DEBUGF(a): astree.cc[32] adopt1():
0x4c2e3d0 (<<ROOT>>) adopting 0x4c343b0 (*)
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 31 ("
")
--(end of buffer or a NUL)
--EOF (start condition 0)
Now at end of input.
Reducing stack by rule 1 (line 41):
   $1 = nterm stmtseq ()
-> $$ = nterm program ()
Stack now 0
Entering state 1
Now at end of input.
Shifting token $end (DEBUGF(f): astree.cc[79] yyprint():
toknum = 0, yyvaluep = 0x4c345d0
$end(0)
)
Entering state 3
Stack now 0 1 3
Cleanup: popping token $end (DEBUGF(f): astree.cc[79] yyprint():
toknum = 0, yyvaluep = 0x4c345d0
$end(0)
)
Cleanup: popping nterm program ()
DEBUGF(a): main.cc[87] main():

<<ROOT>> 0x4c2e3d0->{ROOT(258) 0:0.000 "<<ROOT>>" [0x4c33000 0x4c33430 0x4c33f00 0x4c343b0]}
   pi 0x4c33000->{IDENT(259) 5:4.000 "pi" []}
   = 0x4c33430->{’=’(61) 7:2.001 "=" [0x4c332f0 0x4c335f0]}
      a 0x4c332f0->{IDENT(259) 7:2.000 "a" []}
      * 0x4c335f0->{’*’(42) 7:2.004 "*" [0x4c33510 0x4c33870]}
         pi 0x4c33510->{IDENT(259) 7:2.002 "pi" []}
         ^ 0x4c33870->{’^’(94) 7:2.006 "^" [0x4c33730 0x4c339b0]}
            r 0x4c33730->{IDENT(259) 7:2.005 "r" []}
            2 0x4c339b0->{NUMBER(260) 7:2.007 "2" []}
   3.141592653589793238462643383280 0x4c33f00->{NUMBER(260) 7:3.000 "3.141592653589793238462643383280" []}
   * 0x4c343b0->{’*’(42) 8:4.001 "*" [0x4c34270 0x4c34490]}
      3 0x4c34270->{NUMBER(260) 8:4.000 "3" []}
      4 0x4c34490->{NUMBER(260) 8:4.002 "4" []}
DEBUGF(f): astree.cc[97] free_ast():
```

```
free [4C34490]-> 8:4.2: NUMBER: "4")
DEBUGF(f): astree.cc[97] free_ast():
free [4C34270]-> 8:4.0: NUMBER: "3")
DEBUGF(f): astree.cc[97] free_ast():
free [4C343B0]-> 8:4.1: '*': "*")
DEBUGF(f): astree.cc[97] free_ast():
free [4C33F00]-> 7:3.0: NUMBER: "3.141592653589793238462643383280")
DEBUGF(f): astree.cc[97] free_ast():
free [4C339B0]-> 7:2.7: NUMBER: "2")
DEBUGF(f): astree.cc[97] free_ast():
free [4C33730]-> 7:2.5: IDENT: "r")
DEBUGF(f): astree.cc[97] free_ast():
free [4C33870]-> 7:2.6: '^': "^")
DEBUGF(f): astree.cc[97] free_ast():
free [4C33510]-> 7:2.2: IDENT: "pi")
DEBUGF(f): astree.cc[97] free_ast():
free [4C335F0]-> 7:2.4: '*': "*")
DEBUGF(f): astree.cc[97] free_ast():
free [4C332F0]-> 7:2.0: IDENT: "a")
DEBUGF(f): astree.cc[97] free_ast():
free [4C33430]-> 7:2.1: '=': "=")
DEBUGF(f): astree.cc[97] free_ast():
free [4C33000]-> 5:4.0: IDENT: "pi")
DEBUGF(f): astree.cc[97] free_ast():
free [4C2E3D0]-> 0:0.0: ROOT: "<<ROOT>>")
DEBUGF(s): main.cc[92] main():

stringset[   1]:     831242270113464010 0x4c32a18->"t"
stringset[   3]:    8248777770799913213 0x4c34358->"3"
                    6637313742931709005 0x4c336d8->"*"
                   11597697714117577063 0x4c32d08->"="
                    5344662657899890615 0x4c32fa8->";"
stringset[   6]:    3729804957429652673 0x4c2e4c8->"<<ROOT>>"
stringset[   9]:    4551451650890805270 0x4c33a98->"2"
stringset[  11]:    1998636459596678225 0x4c33958->"^"
stringset[  12]:   14518329498511883088 0x4c33818->"r"
                   15530967369654070964 0x4c32bc8->"pi"
stringset[  17]:    4993892634952068459 0x4c333d8->"a"
stringset[  20]:   17027087115628340017 0x4c32e68->"3.141592653589793238462643383280"
stringset[  21]:   16215888864653804456 0x4c34578->"4"
load_factor = 0.565
bucket_count = 23
max_bucket_size = 4
==8984==
==8984== HEAP SUMMARY:
==8984==     in use at exit: 0 bytes in 0 blocks
==8984==   total heap usage: 90 allocs, 90 frees, 20,145 bytes allocated
==8984==
==8984== All heap blocks were freed -- no leaks are possible
==8984==
==8984== For counts of detected and suppressed errors, rerun with: -v
==8984== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
EXIT STATUS 1
```