```
flex version 2.5.35 usage statistics:
  scanner options: -bdpsvI8 -Cem -oyylex.c
  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
=================================================================
Compressed tables always back up.
```

```
Terminals unused in grammar

   ROOT
   "u+"
   "u-"


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
"u+" (261)
"u-" (262)
NEG (263)
POS (264)


Nonterminals, with rules where they appear

$accept (19)
    on left: 0
program (20)
    on left: 1, on right: 0
stmtseq (21)
    on left: 2 3 4 5, on right: 1 2 3 4
```

```
expr (22)
    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 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.c,v 1.12 2013-08-22 13:58:43-07 - - $
     $Compiled: astree.c Sep 11 2013 20:17:49 $

emit.o:
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
     $Id: emit.h,v 1.3 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: emit.c,v 1.6 2013-08-22 13:58:43-07 - - $
     $Compiled: emit.c Sep 11 2013 20:17:49 $

lyutils.o:
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.c,v 1.12 2013-08-22 13:58:43-07 - - $
     $Compiled: lyutils.c Sep 11 2013 20:17:49 $

main.o:
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: emit.h,v 1.3 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: main.c,v 1.17 2013-08-22 13:58:43-07 - - $
     $Compiled: main.c Sep 11 2013 20:17:49 $

auxlib.o:
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: auxlib.c,v 1.16 2013-08-22 13:59:59-07 - - $
     $Compiled: auxlib.c Sep 11 2013 20:17:49 $

yylex.o:
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: yylex.c,v 1.10 2013-09-11 20:17:45-07 - - $
     $Compiled: scanner.l Sep 11 2013 20:17:49 $

yyparse.o:
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
     $Id: yyparse.c,v 1.5 2013-08-22 14:00:02-07 - - $
     $Compiled: parser.y Sep 11 2013 20:17:49 $

zexprsm:
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.c,v 1.12 2013-08-22 13:58:43-07 - - $
     $Compiled: astree.c Sep 11 2013 20:17:49 $
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
     $Id: emit.h,v 1.3 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: emit.c,v 1.6 2013-08-22 13:58:43-07 - - $
     $Compiled: emit.c Sep 11 2013 20:17:49 $
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.c,v 1.12 2013-08-22 13:58:43-07 - - $
     $Compiled: lyutils.c Sep 11 2013 20:17:49 $
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: emit.h,v 1.3 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: main.c,v 1.17 2013-08-22 13:58:43-07 - - $
     $Compiled: main.c Sep 11 2013 20:17:49 $
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: auxlib.c,v 1.16 2013-08-22 13:59:59-07 - - $
     $Compiled: auxlib.c Sep 11 2013 20:17:49 $
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: yylex.c,v 1.10 2013-09-11 20:17:45-07 - - $
     $Compiled: scanner.l Sep 11 2013 20:17:49 $
     $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
     $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
     $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
     $Id: yyparse.c,v 1.5 2013-08-22 14:00:02-07 - - $
     $Compiled: parser.y Sep 11 2013 20:17:49 $
```

```
// $Id: test1.in,v 1.2 2011-08-31 17:54:03-07 - - $
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
```

```
DEBUGF(x): auxlib.c[117] set_debugflags():
Debugflags = "@", all = 1
zexprsm: bad option (d)
DEBUGF(m): main.c[69] scan_opts():
filename = test1.in, yyin = 0x1e22030, fileno (yyin) = 3
Starting parse
Entering state 0
Reducing stack by rule 5 (line 45):
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e221f0-> 0:0.0: ROOT: 0x1e22230->"<<ROOT>>"
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --(end of buffer or a NUL)
--accepting rule at line 36 ("# 1 "test1.in"")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=test1.in, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 1 "<built-in>"")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=<built-in>, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 1 "<command-line>"")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=<command-line>, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 1 "test1.in"")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=test1.in, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 38 ("
")
--accepting rule at line 41 ("a")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e263d0-> 4:2.0: IDENT: 0x1e26410->"a"
Next token is token IDENT (259=IDENT)
         (0x1e263d0-> astree {IDENT(259), 4:2.000, 0x1e26410->"a",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x1e263d0-> astree {IDENT(259), 4:2.000, 0x1e26410->"a",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x1e263d0-> astree {IDENT(259), 4:2.000, 0x1e26410->"a",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 43 ("=")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e26430-> 4:2.1: '=': 0x1e26470->"="
Next token is token '=' (61='=')
         (0x1e26430-> astree {'='(61), 4:2.001, 0x1e26470->"=",
                     first=(nil), last=(nil), next=(nil)})
Shifting token '=' (61='=')
         (0x1e26430-> astree {'='(61), 4:2.001, 0x1e26470->"=",
                     first=(nil), last=(nil), next=(nil)})
Entering state 16
Reading a token: --accepting rule at line 41 ("b")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e26490-> 4:2.2: IDENT: 0x1e264d0->"b"
Next token is token IDENT (259=IDENT)
         (0x1e26490-> astree {IDENT(259), 4:2.002, 0x1e264d0->"b",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x1e26490-> astree {IDENT(259), 4:2.002, 0x1e264d0->"b",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x1e26490-> astree {IDENT(259), 4:2.002, 0x1e264d0->"b",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Reading a token: --accepting rule at line 46 ("*")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e264f0-> 4:2.3: '*': 0x1e26530->"*"
Next token is token '*' (42='*')
         (0x1e264f0-> astree {'*'(42), 4:2.003, 0x1e26530->"*",
                     first=(nil), last=(nil), next=(nil)})
Shifting token '*' (42='*')
         (0x1e264f0-> astree {'*'(42), 4:2.003, 0x1e26530->"*",
                     first=(nil), last=(nil), next=(nil)})
Entering state 19
Reading a token: --accepting rule at line 41 ("c")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e26550-> 4:2.4: IDENT: 0x1e26590->"c"
Next token is token IDENT (259=IDENT)
         (0x1e26550-> astree {IDENT(259), 4:2.004, 0x1e26590->"c",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x1e26550-> astree {IDENT(259), 4:2.004, 0x1e26590->"c",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x1e26550-> astree {IDENT(259), 4:2.004, 0x1e26590->"c",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 19
Entering state 27
Reading a token: --accepting rule at line 44 ("+")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e265b0-> 4:2.5: '+': 0x1e265f0->"+"
Next token is token '+' (43='+')
         (0x1e265b0-> astree {'+'(43), 4:2.005, 0x1e265f0->"+",
                     first=(nil), last=(nil), next=(nil)})
Reducing stack by rule 9 (line 51):
   $1 = nterm expr ()
   $2 = token '*' (42='*')
         (0x1e264f0-> astree {'*'(42), 4:2.003, 0x1e26530->"*",
                     first=(nil), last=(nil), next=(nil)})
   $3 = nterm expr ()
DEBUGF(a): astree.c[55] adopt():
0x1e264f0 (*) adopting 0x1e26490 (b)
DEBUGF(a): astree.c[55] adopt():
```

```
0x1e264f0 (*) adopting 0x1e26550 (c)
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Next token is token '+' (43='+')
         (0x1e265b0-> astree {'+'(43), 4:2.005, 0x1e265f0->"+",
                     first=(nil), last=(nil), next=(nil)})
Shifting token '+' (43='+')
         (0x1e265b0-> astree {'+'(43), 4:2.005, 0x1e265f0->"+",
                     first=(nil), last=(nil), next=(nil)})
Entering state 17
Reading a token: --accepting rule at line 41 ("d")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e26610-> 4:2.6: IDENT: 0x1e26650->"d"
Next token is token IDENT (259=IDENT)
         (0x1e26610-> astree {IDENT(259), 4:2.006, 0x1e26650->"d",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x1e26610-> astree {IDENT(259), 4:2.006, 0x1e26650->"d",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x1e26610-> astree {IDENT(259), 4:2.006, 0x1e26650->"d",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 17
Entering state 25
Reading a token: --accepting rule at line 46 ("*")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e26670-> 4:2.7: '*': 0x1e266b0->"*"
Next token is token '*' (42='*')
         (0x1e26670-> astree {'*'(42), 4:2.007, 0x1e266b0->"*",
                     first=(nil), last=(nil), next=(nil)})
Shifting token '*' (42='*')
         (0x1e26670-> astree {'*'(42), 4:2.007, 0x1e266b0->"*",
                     first=(nil), last=(nil), next=(nil)})
Entering state 19
Reading a token: --accepting rule at line 41 ("e")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e266d0-> 4:2.8: IDENT: 0x1e26710->"e"
Next token is token IDENT (259=IDENT)
         (0x1e266d0-> astree {IDENT(259), 4:2.008, 0x1e26710->"e",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x1e266d0-> astree {IDENT(259), 4:2.008, 0x1e26710->"e",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x1e266d0-> astree {IDENT(259), 4:2.008, 0x1e26710->"e",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 17 25 19
Entering state 27
Reading a token: --accepting rule at line 51 (";")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x1e26730-> 4:2.9: ';': 0x1e26770->";"
Next token is token ';' (59=';')
         (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
                     first=(nil), last=(nil), next=(nil)})
```

```
Reducing stack by rule 9 (line 51):
   $1 = nterm expr ()
   $2 = token '*' (42='*')
         (0x1e26670-> astree {'*'(42), 4:2.007, 0x1e266b0->"*",
                     first=(nil), last=(nil), next=(nil)})
   $3 = nterm expr ()
DEBUGF(a): astree.c[55] adopt():
0x1e26670 (*) adopting 0x1e26610 (d)
DEBUGF(a): astree.c[55] adopt():
0x1e26670 (*) adopting 0x1e266d0 (e)
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 17
Entering state 25
Next token is token ';' (59=';')
         (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
                     first=(nil), last=(nil), next=(nil)})
Reducing stack by rule 7 (line 49):
   $1 = nterm expr ()
   $2 = token '+' (43='+')
         (0x1e265b0-> astree {'+'(43), 4:2.005, 0x1e265f0->"+",
                     first=(nil), last=(nil), next=(nil)})
   $3 = nterm expr ()
DEBUGF(a): astree.c[55] adopt():
0x1e265b0 (+) adopting 0x1e264f0 (*)
DEBUGF(a): astree.c[55] adopt():
0x1e265b0 (+) adopting 0x1e26670 (*)
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Next token is token ';' (59=';')
         (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
                     first=(nil), last=(nil), next=(nil)})
Reducing stack by rule 6 (line 48):
   $1 = nterm expr ()
   $2 = token '=' (61='=')
         (0x1e26430-> astree {'='(61), 4:2.001, 0x1e26470->"=",
                     first=(nil), last=(nil), next=(nil)})
   $3 = nterm expr ()
DEBUGF(a): astree.c[55] adopt():
0x1e26430 (=) adopting 0x1e263d0 (a)
DEBUGF(a): astree.c[55] adopt():
0x1e26430 (=) adopting 0x1e265b0 (+)
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Next token is token ';' (59=';')
         (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
                     first=(nil), last=(nil), next=(nil)})
Shifting token ';' (59=';')
         (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
                     first=(nil), last=(nil), next=(nil)})
Entering state 22
Reducing stack by rule 2 (line 42):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ';' (59=';')
         (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
                     first=(nil), last=(nil), next=(nil)})
DEBUGF(f): astree.c[127] freeast():
free [1E26730]-> 4:2.9: ';': 0x1e26770->";")
DEBUGF(a): astree.c[55] adopt():
```

```
0x1e221f0 (<<ROOT>>) adopting 0x1e26430 (=)
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 38 ("
")
--(end of buffer or a NUL)
--EOF (start condition 0)
Now at end of input.
Reducing stack by rule 1 (line 39):
   $1 = nterm stmtseq ()
-> $$ = nterm program ()
Stack now 0
Entering state 1
Now at end of input.
Shifting token $end (0=$end)
         (yyvaluep = 0x1e26730)
Entering state 3
Stack now 0 1 3
Cleanup: popping token $end (0=$end)
         (yyvaluep = 0x1e26730)
Cleanup: popping nterm program ()
DEBUGF(a): main.c[88] main():

<<ROOT>> 0x1e221f0-> astree {ROOT(258), 0:0.000, 0x1e22230->"<<ROOT>>",
            first=0x1e26430, last=0x1e26430, next=(nil)}
   = 0x1e26430-> astree {'='(61), 4:2.001, 0x1e26470->"=",
               first=0x1e263d0, last=0x1e265b0, next=(nil)}
      a 0x1e263d0-> astree {IDENT(259), 4:2.000, 0x1e26410->"a",
                  first=(nil), last=(nil), next=0x1e265b0}
      + 0x1e265b0-> astree {'+'(43), 4:2.005, 0x1e265f0->"+",
                  first=0x1e264f0, last=0x1e26670, next=(nil)}
         * 0x1e264f0-> astree {'*'(42), 4:2.003, 0x1e26530->"*",
                     first=0x1e26490, last=0x1e26550, next=0x1e26670}
            b 0x1e26490-> astree {IDENT(259), 4:2.002, 0x1e264d0->"b",
                        first=(nil), last=(nil), next=0x1e26550}
            c 0x1e26550-> astree {IDENT(259), 4:2.004, 0x1e26590->"c",
                        first=(nil), last=(nil), next=(nil)}
         * 0x1e26670-> astree {'*'(42), 4:2.007, 0x1e266b0->"*",
                     first=0x1e26610, last=0x1e266d0, next=(nil)}
            d 0x1e26610-> astree {IDENT(259), 4:2.006, 0x1e26650->"d",
                        first=(nil), last=(nil), next=0x1e266d0}
            e 0x1e266d0-> astree {IDENT(259), 4:2.008, 0x1e26710->"e",
                        first=(nil), last=(nil), next=(nil)}
DEBUGF(f): astree.c[127] freeast():
free [1E263D0]-> 4:2.0: IDENT: 0x1e26410->"a")
DEBUGF(f): astree.c[127] freeast():
free [1E26490]-> 4:2.2: IDENT: 0x1e264d0->"b")
DEBUGF(f): astree.c[127] freeast():
free [1E26550]-> 4:2.4: IDENT: 0x1e26590->"c")
DEBUGF(f): astree.c[127] freeast():
free [1E264F0]-> 4:2.3: '*': 0x1e26530->"*")
DEBUGF(f): astree.c[127] freeast():
free [1E26610]-> 4:2.6: IDENT: 0x1e26650->"d")
DEBUGF(f): astree.c[127] freeast():
free [1E266D0]-> 4:2.8: IDENT: 0x1e26710->"e")
DEBUGF(f): astree.c[127] freeast():
free [1E26670]-> 4:2.7: '*': 0x1e266b0->"*")
DEBUGF(f): astree.c[127] freeast():
free [1E265B0]-> 4:2.5: '+': 0x1e265f0->"+")
DEBUGF(f): astree.c[127] freeast():
```

```
free [1E26430]-> 4:2.1: ’=’: 0x1e26470->"=")
DEBUGF(f): astree.c[127] freeast():
free [1E221F0]-> 0:0.0: ROOT: 0x1e22230->"<<ROOT>>")
```

```
Script  : /afs/cats.ucsc.edu/courses/cmps012b-wm/bin/runprog
limit c :    0 max core file size (KB)
limit f : 4194303 max output file size (KB)
limit t : 4294967295 max CPU time (sec)
stdin   : test1.in
stdout  : test1.out
stderr  : test1.err
log     : test1.log
base    : test1
Command : zexprsm -@@ -ly -de test1.in
starting: pid 17436: 20:17:49.00
finished: pid 17436: 20:17:49.00, real 0.00, user 0.00, sys 0.01
pstatus: 0x0100 EXIT STATUS = 1
```

```
 t// $Id: test4.in,v 1.4 2012-10-11 19:07:58-07 - - $
#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=3.14159265358979323846264338328O;
;    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: 3.14159265358979323846264338328O;
;    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   3.14159265358979323846264338328O; test4b.inh 3.0
          pushnum   3                   ; test4.in 4.0
          pushnum   4                   ; test4.in 4.2
          mul                           ; test4.in 4.1
```

```
DEBUGF(x): auxlib.c[117] set_debugflags():
Debugflags = "@", all = 1
zexprsm: bad option (d)
DEBUGF(m): main.c[69] scan_opts():
filename = test4.in, yyin = 0x661030, fileno (yyin) = 3
Starting parse
Entering state 0
Reducing stack by rule 5 (line 45):
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6611f0-> 0:0.0: ROOT: 0x661230->"<<ROOT>>"
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --(end of buffer or a NUL)
--accepting rule at line 36 ("# 1 "test4.in"")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=test4.in, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 1 "<built-in>"")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=<built-in>, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 1 "<command-line>"")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=<command-line>, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 1 "test4.in"")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=test4.in, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 37 (" ")
--accepting rule at line 41 ("t")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6653d0-> 4:1.1: IDENT: 0x665410->"t"
Next token is token IDENT (259=IDENT)
         (0x6653d0-> astree {IDENT(259), 4:1.001, 0x665410->"t",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x6653d0-> astree {IDENT(259), 4:1.001, 0x665410->"t",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x6653d0-> astree {IDENT(259), 4:1.001, 0x665410->"t",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 1 "test4a.inh" 1")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=test4a.inh, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 38 ("
")
```

```
--accepting rule at line 38 ("
")
--accepting rule at line 41 ("pi")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665470-> 5:3.0: IDENT: 0x6654b0->"pi"
Next token is token IDENT (259=IDENT)
         (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
                    first=(nil), last=(nil), next=(nil)})
zexprsm: test4a.inh: 3: syntax error, unexpected IDENT
Error: popping nterm expr ()
Stack now 0 2
Shifting token error (256=error)
         (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
                    first=(nil), last=(nil), next=(nil)})
Entering state 4
Next token is token IDENT (259=IDENT)
         (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
                    first=(nil), last=(nil), next=(nil)})
Error: discarding token IDENT (259=IDENT)
         (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
                    first=(nil), last=(nil), next=(nil)})
Error: popping token error (256=error)
         (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
                    first=(nil), last=(nil), next=(nil)})
Stack now 0 2
Shifting token error (256=error)
         (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
                    first=(nil), last=(nil), next=(nil)})
Entering state 4
Reading a token: --accepting rule at line 43 ("=")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6654d0-> 5:3.2: '=': 0x665510->"="
Next token is token '=' (61='=')
         (0x6654d0-> astree {'='(61), 5:3.002, 0x665510->"=",
                    first=(nil), last=(nil), next=(nil)})
Error: discarding token '=' (61='=')
         (0x6654d0-> astree {'='(61), 5:3.002, 0x665510->"=",
                    first=(nil), last=(nil), next=(nil)})
Error: popping token error (256=error)
         (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
                    first=(nil), last=(nil), next=(nil)})
Stack now 0 2
Shifting token error (256=error)
         (0x6654d0-> astree {'='(61), 5:3.002, 0x665510->"=",
                    first=(nil), last=(nil), next=(nil)})
Entering state 4
Reading a token: --accepting rule at line 40 ("3.141592653589793238462643383280")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665530-> 5:3.3: NUMBER: 0x665570->"3.141592653589793238462643383280"
Next token is token NUMBER (260=NUMBER)
         (0x665530-> astree {NUMBER(260), 5:3.003, 0x665570->"3.141592653589793238462643383280",
                    first=(nil), last=(nil), next=(nil)})
Error: discarding token NUMBER (260=NUMBER)
         (0x665530-> astree {NUMBER(260), 5:3.003, 0x665570->"3.141592653589793238462643383280",
                    first=(nil), last=(nil), next=(nil)})
Error: popping token error (256=error)
         (0x6654d0-> astree {'='(61), 5:3.002, 0x665510->"=",
                     first=(nil), last=(nil), next=(nil)})
Stack now 0 2
Shifting token error (256=error)
         (0x665530-> astree {NUMBER(260), 5:3.003, 0x665570->"3.141592653589793238462643383280",
                     first=(nil), last=(nil), next=(nil)})
Entering state 4
Reading a token: --accepting rule at line 51 (";")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6655a0-> 5:3.35: ';': 0x6655e0->";"
Next token is token ';' (59=';')
         (0x6655a0-> astree {';'(59), 5:3.035, 0x6655e0->";",
                     first=(nil), last=(nil), next=(nil)})
Shifting token ';' (59=';')
         (0x6655a0-> astree {';'(59), 5:3.035, 0x6655e0->";",
                     first=(nil), last=(nil), next=(nil)})
Entering state 12
Reducing stack by rule 3 (line 43):
   $1 = nterm stmtseq ()
   $2 = token error (256=error)
         (0x665530-> astree {NUMBER(260), 5:3.003, 0x665570->"3.141592653589793238462643383280",
                     first=(nil), last=(nil), next=(nil)})
   $3 = token ';' (59=';')
         (0x6655a0-> astree {';'(59), 5:3.035, 0x6655e0->";",
                     first=(nil), last=(nil), next=(nil)})
DEBUGF(f): astree.c[127] freeast():
free [6655A0]-> 5:3.35: ';': 0x6655e0->";")
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 38 ("
")
--accepting rule at line 41 ("pi")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6655a0-> 5:4.0: IDENT: 0x6655e0->"pi"
Next token is token IDENT (259=IDENT)
         (0x6655a0-> astree {IDENT(259), 5:4.000, 0x6655e0->"pi",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x6655a0-> astree {IDENT(259), 5:4.000, 0x6655e0->"pi",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x6655a0-> astree {IDENT(259), 5:4.000, 0x6655e0->"pi",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 51 (";")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665600-> 5:4.2: ';': 0x665640->";"
Next token is token ';' (59=';')
         (0x665600-> astree {';'(59), 5:4.002, 0x665640->";",
                     first=(nil), last=(nil), next=(nil)})
Shifting token ';' (59=';')
         (0x665600-> astree {';'(59), 5:4.002, 0x665640->";",
                     first=(nil), last=(nil), next=(nil)})
Entering state 22
Reducing stack by rule 2 (line 42):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ';' (59=';')
         (0x665600-> astree {';'(59), 5:4.002, 0x665640->";",
                     first=(nil), last=(nil), next=(nil)})
DEBUGF(f): astree.c[127] freeast():
free [665600]-> 5:4.2: ';': 0x665640->";")
DEBUGF(a): astree.c[55] adopt():
0x6611f0 (<<ROOT>>) adopting 0x6655a0 (pi)
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 3 "test4.in" 2")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=test4.in, scan_linenr=2
--accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 1 "test4b.inh" 1")
DEBUGF(m): lyutils.c[118] scanner_include():
filename=test4b.inh, scan_linenr=0
--accepting rule at line 38 ("
")
--accepting rule at line 38 ("
")
--accepting rule at line 41 ("a")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665600-> 7:2.0: IDENT: 0x6656c0->"a"
Next token is token IDENT (259=IDENT)
         (0x665600-> astree {IDENT(259), 7:2.000, 0x6656c0->"a",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x665600-> astree {IDENT(259), 7:2.000, 0x6656c0->"a",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x665600-> astree {IDENT(259), 7:2.000, 0x6656c0->"a",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 43 ("=")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6656e0-> 7:2.1: '=': 0x665720->"="
Next token is token '=' (61='=')
         (0x6656e0-> astree {'='(61), 7:2.001, 0x665720->"=",
                     first=(nil), last=(nil), next=(nil)})
Shifting token '=' (61='=')
         (0x6656e0-> astree {'='(61), 7:2.001, 0x665720->"=",
                     first=(nil), last=(nil), next=(nil)})
Entering state 16
Reading a token: --accepting rule at line 41 ("pi")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665740-> 7:2.2: IDENT: 0x665780->"pi"
Next token is token IDENT (259=IDENT)
         (0x665740-> astree {IDENT(259), 7:2.002, 0x665780->"pi",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x665740-> astree {IDENT(259), 7:2.002, 0x665780->"pi",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x665740-> astree {IDENT(259), 7:2.002, 0x665780->"pi",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Reading a token: --accepting rule at line 46 ("*")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6657a0-> 7:2.4: ’*’: 0x6657e0->"*"
Next token is token ’*’ (42=’*’)
         (0x6657a0-> astree {’*’(42), 7:2.004, 0x6657e0->"*",
                     first=(nil), last=(nil), next=(nil)})
Shifting token ’*’ (42=’*’)
         (0x6657a0-> astree {’*’(42), 7:2.004, 0x6657e0->"*",
                     first=(nil), last=(nil), next=(nil)})
Entering state 19
Reading a token: --accepting rule at line 41 ("r")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665800-> 7:2.5: IDENT: 0x665840->"r"
Next token is token IDENT (259=IDENT)
         (0x665800-> astree {IDENT(259), 7:2.005, 0x665840->"r",
                     first=(nil), last=(nil), next=(nil)})
Shifting token IDENT (259=IDENT)
         (0x665800-> astree {IDENT(259), 7:2.005, 0x665840->"r",
                     first=(nil), last=(nil), next=(nil)})
Entering state 5
Reducing stack by rule 15 (line 57):
   $1 = token IDENT (259=IDENT)
         (0x665800-> astree {IDENT(259), 7:2.005, 0x665840->"r",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 19
Entering state 27
Reading a token: --accepting rule at line 48 ("^")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665860-> 7:2.6: ’^’: 0x6658a0->"^"
Next token is token ’^’ (94=’^’)
         (0x665860-> astree {’^’(94), 7:2.006, 0x6658a0->"^",
                     first=(nil), last=(nil), next=(nil)})
Shifting token ’^’ (94=’^’)
         (0x665860-> astree {’^’(94), 7:2.006, 0x6658a0->"^",
                     first=(nil), last=(nil), next=(nil)})
Entering state 21
Reading a token: --accepting rule at line 40 ("2")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6658c0-> 7:2.7: NUMBER: 0x665900->"2"
Next token is token NUMBER (260=NUMBER)
         (0x6658c0-> astree {NUMBER(260), 7:2.007, 0x665900->"2",
                     first=(nil), last=(nil), next=(nil)})
Shifting token NUMBER (260=NUMBER)
         (0x6658c0-> astree {NUMBER(260), 7:2.007, 0x665900->"2",
                     first=(nil), last=(nil), next=(nil)})
Entering state 6
Reducing stack by rule 16 (line 58):
   $1 = token NUMBER (260=NUMBER)
         (0x6658c0-> astree {NUMBER(260), 7:2.007, 0x665900->"2",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
```

```
Stack now 0 2 11 16 24 19 27 21
Entering state 29
Reading a token: --accepting rule at line 51 (";")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665920-> 7:2.8: ';': 0x665960->";"
Next token is token ';' (59=';')
         (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
                     first=(nil), last=(nil), next=(nil)})
Reducing stack by rule 11 (line 53):
   $1 = nterm expr ()
   $2 = token '^' (94='^')
         (0x665860-> astree {'^'(94), 7:2.006, 0x6658a0->"^",
                     first=(nil), last=(nil), next=(nil)})
   $3 = nterm expr ()
DEBUGF(a): astree.c[55] adopt():
0x665860 (^) adopting 0x665800 (r)
DEBUGF(a): astree.c[55] adopt():
0x665860 (^) adopting 0x6658c0 (2)
-> $$ = nterm expr ()
Stack now 0 2 11 16 24 19
Entering state 27
Next token is token ';' (59=';')
         (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
                     first=(nil), last=(nil), next=(nil)})
Reducing stack by rule 9 (line 51):
   $1 = nterm expr ()
   $2 = token '*' (42='*')
         (0x6657a0-> astree {'*'(42), 7:2.004, 0x6657e0->"*",
                     first=(nil), last=(nil), next=(nil)})
   $3 = nterm expr ()
DEBUGF(a): astree.c[55] adopt():
0x6657a0 (*) adopting 0x665740 (pi)
DEBUGF(a): astree.c[55] adopt():
0x6657a0 (*) adopting 0x665860 (^)
-> $$ = nterm expr ()
Stack now 0 2 11 16
Entering state 24
Next token is token ';' (59=';')
         (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
                     first=(nil), last=(nil), next=(nil)})
Reducing stack by rule 6 (line 48):
   $1 = nterm expr ()
   $2 = token '=' (61='=')
         (0x6656e0-> astree {'='(61), 7:2.001, 0x665720->"=",
                     first=(nil), last=(nil), next=(nil)})
   $3 = nterm expr ()
DEBUGF(a): astree.c[55] adopt():
0x6656e0 (=) adopting 0x665600 (a)
DEBUGF(a): astree.c[55] adopt():
0x6656e0 (=) adopting 0x6657a0 (*)
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Next token is token ';' (59=';')
         (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
                     first=(nil), last=(nil), next=(nil)})
Shifting token ';' (59=';')
         (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
                     first=(nil), last=(nil), next=(nil)})
Entering state 22
Reducing stack by rule 2 (line 42):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ';' (59=';')
          (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
                      first=(nil), last=(nil), next=(nil)})
DEBUGF(f): astree.c[127] freeast():
free [665920]-> 7:2.8: ';': 0x665960->";")
DEBUGF(a): astree.c[55] adopt():
0x6611f0 (<<ROOT>>) adopting 0x6656e0 (=)
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 38 ("
")
--accepting rule at line 40 ("3.14159265358979323846264383280")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665920-> 7:3.0: NUMBER: 0x665980->"3.14159265358979323846264383280"
Next token is token NUMBER (260=NUMBER)
          (0x665920-> astree {NUMBER(260), 7:3.000, 0x665980->"3.14159265358979323846264383280",
                      first=(nil), last=(nil), next=(nil)})
Shifting token NUMBER (260=NUMBER)
          (0x665920-> astree {NUMBER(260), 7:3.000, 0x665980->"3.14159265358979323846264383280",
                      first=(nil), last=(nil), next=(nil)})
Entering state 6
Reducing stack by rule 16 (line 58):
   $1 = token NUMBER (260=NUMBER)
          (0x665920-> astree {NUMBER(260), 7:3.000, 0x665980->"3.14159265358979323846264383280",
                      first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 51 (";")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6659b0-> 7:3.32: ';': 0x665960->";"
Next token is token ';' (59=';')
          (0x6659b0-> astree {';'(59), 7:3.032, 0x665960->";",
                      first=(nil), last=(nil), next=(nil)})
Shifting token ';' (59=';')
          (0x6659b0-> astree {';'(59), 7:3.032, 0x665960->";",
                      first=(nil), last=(nil), next=(nil)})
Entering state 22
Reducing stack by rule 2 (line 42):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ';' (59=';')
          (0x6659b0-> astree {';'(59), 7:3.032, 0x665960->";",
                      first=(nil), last=(nil), next=(nil)})
DEBUGF(f): astree.c[127] freeast():
free [6659B0]-> 7:3.32: ';': 0x665960->";")
DEBUGF(a): astree.c[55] adopt():
0x6611f0 (<<ROOT>>) adopting 0x665920 (3.14159265358979323846264383280)
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 38 ("
")
--accepting rule at line 36 ("# 4 "test4.in" 2")
```

```
DEBUGF(m): lyutils.c[118] scanner_include():
filename=test4.in, scan_linenr=3
--accepting rule at line 38 ("
")
--accepting rule at line 40 ("3")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x6659b0-> 8:4.0: NUMBER: 0x665a10->"3"
Next token is token NUMBER (260=NUMBER)
         (0x6659b0-> astree {NUMBER(260), 8:4.000, 0x665a10->"3",
                     first=(nil), last=(nil), next=(nil)})
Shifting token NUMBER (260=NUMBER)
         (0x6659b0-> astree {NUMBER(260), 8:4.000, 0x665a10->"3",
                     first=(nil), last=(nil), next=(nil)})
Entering state 6
Reducing stack by rule 16 (line 58):
   $1 = token NUMBER (260=NUMBER)
         (0x6659b0-> astree {NUMBER(260), 8:4.000, 0x665a10->"3",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Reading a token: --accepting rule at line 46 ("*")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665a30-> 8:4.1: '*': 0x665a70->"*"
Next token is token '*' (42='*')
         (0x665a30-> astree {'*'(42), 8:4.001, 0x665a70->"*",
                     first=(nil), last=(nil), next=(nil)})
Shifting token '*' (42='*')
         (0x665a30-> astree {'*'(42), 8:4.001, 0x665a70->"*",
                     first=(nil), last=(nil), next=(nil)})
Entering state 19
Reading a token: --accepting rule at line 40 ("4")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665a90-> 8:4.2: NUMBER: 0x665ad0->"4"
Next token is token NUMBER (260=NUMBER)
         (0x665a90-> astree {NUMBER(260), 8:4.002, 0x665ad0->"4",
                     first=(nil), last=(nil), next=(nil)})
Shifting token NUMBER (260=NUMBER)
         (0x665a90-> astree {NUMBER(260), 8:4.002, 0x665ad0->"4",
                     first=(nil), last=(nil), next=(nil)})
Entering state 6
Reducing stack by rule 16 (line 58):
   $1 = token NUMBER (260=NUMBER)
         (0x665a90-> astree {NUMBER(260), 8:4.002, 0x665ad0->"4",
                     first=(nil), last=(nil), next=(nil)})
-> $$ = nterm expr ()
Stack now 0 2 11 19
Entering state 27
Reading a token: --accepting rule at line 51 (";")
DEBUGF(f): astree.c[37] new_astree():
malloc (56) = 0x665af0-> 8:4.3: ';': 0x665b30->";"
Next token is token ';' (59=';')
         (0x665af0-> astree {';'(59), 8:4.003, 0x665b30->";",
                     first=(nil), last=(nil), next=(nil)})
Reducing stack by rule 9 (line 51):
   $1 = nterm expr ()
   $2 = token '*' (42='*')
         (0x665a30-> astree {'*'(42), 8:4.001, 0x665a70->"*",
                     first=(nil), last=(nil), next=(nil)})
   $3 = nterm expr ()
DEBUGF(a): astree.c[55] adopt():
```

```
0x665a30 (*) adopting 0x6659b0 (3)
DEBUGF(a): astree.c[55] adopt():
0x665a30 (*) adopting 0x665a90 (4)
-> $$ = nterm expr ()
Stack now 0 2
Entering state 11
Next token is token ';' (59=';')
         (0x665af0-> astree {';'(59), 8:4.003, 0x665b30->";",
                     first=(nil), last=(nil), next=(nil)})
Shifting token ';' (59=';')
         (0x665af0-> astree {';'(59), 8:4.003, 0x665b30->";",
                     first=(nil), last=(nil), next=(nil)})
Entering state 22
Reducing stack by rule 2 (line 42):
   $1 = nterm stmtseq ()
   $2 = nterm expr ()
   $3 = token ';' (59=';')
         (0x665af0-> astree {';'(59), 8:4.003, 0x665b30->";",
                     first=(nil), last=(nil), next=(nil)})
DEBUGF(f): astree.c[127] freeast():
free [665AF0]-> 8:4.3: ';': 0x665b30->";")
DEBUGF(a): astree.c[55] adopt():
0x6611f0 (<<ROOT>>) adopting 0x665a30 (*)
-> $$ = nterm stmtseq ()
Stack now 0
Entering state 2
Reading a token: --accepting rule at line 38 ("
")
--(end of buffer or a NUL)
--EOF (start condition 0)
Now at end of input.
Reducing stack by rule 1 (line 39):
   $1 = nterm stmtseq ()
-> $$ = nterm program ()
Stack now 0
Entering state 1
Now at end of input.
Shifting token $end (0=$end)
         (yyvaluep = 0x665af0)
Entering state 3
Stack now 0 1 3
Cleanup: popping token $end (0=$end)
         (yyvaluep = 0x665af0)
Cleanup: popping nterm program ()
DEBUGF(a): main.c[88] main():

<<ROOT>> 0x6611f0-> astree {ROOT(258), 0:0.000, 0x661230->"<<ROOT>>",
            first=0x6655a0, last=0x665a30, next=(nil)}
   pi 0x6655a0-> astree {IDENT(259), 5:4.000, 0x6655e0->"pi",
               first=(nil), last=(nil), next=0x6656e0}
   = 0x6656e0-> astree {'='(61), 7:2.001, 0x665720->"=",
               first=0x665600, last=0x6657a0, next=0x665920}
      a 0x665600-> astree {IDENT(259), 7:2.000, 0x6656c0->"a",
                  first=(nil), last=(nil), next=0x6657a0}
      * 0x6657a0-> astree {'*'(42), 7:2.004, 0x6657e0->"*",
                  first=0x665740, last=0x665860, next=(nil)}
         pi 0x665740-> astree {IDENT(259), 7:2.002, 0x665780->"pi",
                     first=(nil), last=(nil), next=0x665860}
         ^ 0x665860-> astree {'^'(94), 7:2.006, 0x6658a0->"^",
                     first=0x665800, last=0x6658c0, next=(nil)}
            r 0x665800-> astree {IDENT(259), 7:2.005, 0x665840->"r",
                        first=(nil), last=(nil), next=0x6658c0}
            2 0x6658c0-> astree {NUMBER(260), 7:2.007, 0x665900->"2",
                        first=(nil), last=(nil), next=(nil)}
   3.141592653589793238462643383280 0x665920-> astree {NUMBER(260), 7:3.000, 0x665980->"3.141592653589793238462643383280",
               first=(nil), last=(nil), next=0x665a30}
   * 0x665a30-> astree {'*'(42), 8:4.001, 0x665a70->"*",
               first=0x6659b0, last=0x665a90, next=(nil)}
      3 0x6659b0-> astree {NUMBER(260), 8:4.000, 0x665a10->"3",
                  first=(nil), last=(nil), next=0x665a90}
      4 0x665a90-> astree {NUMBER(260), 8:4.002, 0x665ad0->"4",
                  first=(nil), last=(nil), next=(nil)}
DEBUGF(f): astree.c[127] freeast():
free [6655A0]-> 5:4.0: IDENT: 0x6655e0->"pi")
DEBUGF(f): astree.c[127] freeast():
free [665600]-> 7:2.0: IDENT: 0x6656c0->"a")
DEBUGF(f): astree.c[127] freeast():
free [665740]-> 7:2.2: IDENT: 0x665780->"pi")
DEBUGF(f): astree.c[127] freeast():
free [665800]-> 7:2.5: IDENT: 0x665840->"r")
DEBUGF(f): astree.c[127] freeast():
free [6658C0]-> 7:2.7: NUMBER: 0x665900->"2")
DEBUGF(f): astree.c[127] freeast():
free [665860]-> 7:2.6: '^': 0x6658a0->"^")
DEBUGF(f): astree.c[127] freeast():
free [6657A0]-> 7:2.4: '*': 0x6657e0->"*")
DEBUGF(f): astree.c[127] freeast():
free [6656E0]-> 7:2.1: '=': 0x665720->"=")
DEBUGF(f): astree.c[127] freeast():
free [665920]-> 7:3.0: NUMBER: 0x665980->"3.141592653589793238462643383280")
DEBUGF(f): astree.c[127] freeast():
free [6659B0]-> 8:4.0: NUMBER: 0x665a10->"3")
DEBUGF(f): astree.c[127] freeast():
free [665A90]-> 8:4.2: NUMBER: 0x665ad0->"4")
DEBUGF(f): astree.c[127] freeast():
free [665A30]-> 8:4.1: '*': 0x665a70->"*")
DEBUGF(f): astree.c[127] freeast():
free [6611F0]-> 0:0.0: ROOT: 0x661230->"<<ROOT>>")
```

```
Script  : /afs/cats.ucsc.edu/courses/cmps012b-wm/bin/runprog
limit c :    0 max core file size (KB)
limit f : 4194303 max output file size (KB)
limit t : 4294967295 max CPU time (sec)
stdin   : test4.in
stdout  : test4.out
stderr  : test4.err
log     : test4.log
base    : test4
Command : zexprsm -@@ -ly -de test4.in
starting: pid 17441: 20:17:49.00
finished: pid 17441: 20:17:49.00, real 0.00, user 0.00, sys 0.01
pstatus: 0x0100 EXIT STATUS = 1
```