```
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
```

```
  62: expr (22)
  63:     on left: 6 7 8 9 10 11 12 13 14 15 16, on right: 2 6 7 8 9 10 11
  64:     12 13 14
  65:
  66:
  67: state 0
  68:
  69:     0 $accept: . program $end
  70:
  71:     $default  reduce using rule 5 (stmtseq)
  72:
  73:     program  go to state 1
  74:     stmtseq  go to state 2
  75:
  76:
  77: state 1
  78:
  79:     0 $accept: program . $end
  80:
  81:     $end  shift, and go to state 3
  82:
  83:
  84: state 2
  85:
  86:     1 program: stmtseq .
  87:     2 stmtseq: stmtseq . expr ';'
  88:     3        | stmtseq . error ';'
  89:     4        | stmtseq . ';'
  90:
  91:     error   shift, and go to state 4
  92:     IDENT   shift, and go to state 5
  93:     NUMBER  shift, and go to state 6
  94:     '+'     shift, and go to state 7
  95:     '-'     shift, and go to state 8
  96:     ';'     shift, and go to state 9
  97:     '('     shift, and go to state 10
  98:
  99:     $end  reduce using rule 1 (program)
 100:
 101:     expr  go to state 11
 102:
 103:
 104: state 3
 105:
 106:     0 $accept: program $end .
 107:
 108:     $default  accept
 109:
 110:
 111: state 4
 112:
 113:     3 stmtseq: stmtseq error . ';'
 114:
 115:     ';'  shift, and go to state 12
 116:
 117:
 118: state 5
 119:
 120:    15 expr: IDENT .
 121:
 122:     $default  reduce using rule 15 (expr)
```

```
123:
124:
125: state 6
126:
127:    16 expr: NUMBER .
128:
129:     $default  reduce using rule 16 (expr)
130:
131:
132: state 7
133:
134:    12 expr: '+' . expr
135:
136:     IDENT   shift, and go to state 5
137:     NUMBER  shift, and go to state 6
138:     '+'     shift, and go to state 7
139:     '-'     shift, and go to state 8
140:     '('     shift, and go to state 10
141:
142:     expr  go to state 13
143:
144:
145: state 8
146:
147:    13 expr: '-' . expr
148:
149:     IDENT   shift, and go to state 5
150:     NUMBER  shift, and go to state 6
151:     '+'     shift, and go to state 7
152:     '-'     shift, and go to state 8
153:     '('     shift, and go to state 10
154:
155:     expr  go to state 14
156:
157:
158: state 9
159:
160:     4 stmtseq: stmtseq ';' .
161:
162:     $default  reduce using rule 4 (stmtseq)
163:
164:
165: state 10
166:
167:    14 expr: '(' . expr ')'
168:
169:     IDENT   shift, and go to state 5
170:     NUMBER  shift, and go to state 6
171:     '+'     shift, and go to state 7
172:     '-'     shift, and go to state 8
173:     '('     shift, and go to state 10
174:
175:     expr  go to state 15
176:
177:
178: state 11
179:
180:     2 stmtseq: stmtseq expr . ';'
181:     6 expr: expr . '=' expr
182:     7     | expr . '+' expr
183:     8     | expr . '-' expr
```

```
184:     9     | expr . '*' expr
185:    10     | expr . '/' expr
186:    11     | expr . '^' expr
187:
188:     '='  shift, and go to state 16
189:     '+'  shift, and go to state 17
190:     '-'  shift, and go to state 18
191:     '*'  shift, and go to state 19
192:     '/'  shift, and go to state 20
193:     '^'  shift, and go to state 21
194:     ';'  shift, and go to state 22
195:
196:
197: state 12
198:
199:     3 stmtseq: stmtseq error ';' .
200:
201:     $default  reduce using rule 3 (stmtseq)
202:
203:
204: state 13
205:
206:     6 expr: expr . '=' expr
207:     7     | expr . '+' expr
208:     8     | expr . '-' expr
209:     9     | expr . '*' expr
210:    10     | expr . '/' expr
211:    11     | expr . '^' expr
212:    12     | '+' expr .
213:
214:     $default  reduce using rule 12 (expr)
215:
216:
217: state 14
218:
219:     6 expr: expr . '=' expr
220:     7     | expr . '+' expr
221:     8     | expr . '-' expr
222:     9     | expr . '*' expr
223:    10     | expr . '/' expr
224:    11     | expr . '^' expr
225:    13     | '-' expr .
226:
227:     $default  reduce using rule 13 (expr)
228:
229:
230: state 15
231:
232:     6 expr: expr . '=' expr
233:     7     | expr . '+' expr
234:     8     | expr . '-' expr
235:     9     | expr . '*' expr
236:    10     | expr . '/' expr
237:    11     | expr . '^' expr
238:    14     | '(' expr . ')'
239:
240:     '='  shift, and go to state 16
241:     '+'  shift, and go to state 17
242:     '-'  shift, and go to state 18
243:     '*'  shift, and go to state 19
244:     '/'  shift, and go to state 20
```

```
245:     '^'  shift, and go to state 21
246:     ')'  shift, and go to state 23
247:
248:
249: state 16
250:
251:     6 expr: expr '=' . expr
252:
253:     IDENT   shift, and go to state 5
254:     NUMBER  shift, and go to state 6
255:     '+'     shift, and go to state 7
256:     '-'     shift, and go to state 8
257:     '('     shift, and go to state 10
258:
259:     expr  go to state 24
260:
261:
262: state 17
263:
264:     7 expr: expr '+' . expr
265:
266:     IDENT   shift, and go to state 5
267:     NUMBER  shift, and go to state 6
268:     '+'     shift, and go to state 7
269:     '-'     shift, and go to state 8
270:     '('     shift, and go to state 10
271:
272:     expr  go to state 25
273:
274:
275: state 18
276:
277:     8 expr: expr '-' . expr
278:
279:     IDENT   shift, and go to state 5
280:     NUMBER  shift, and go to state 6
281:     '+'     shift, and go to state 7
282:     '-'     shift, and go to state 8
283:     '('     shift, and go to state 10
284:
285:     expr  go to state 26
286:
287:
288: state 19
289:
290:     9 expr: expr '*' . expr
291:
292:     IDENT   shift, and go to state 5
293:     NUMBER  shift, and go to state 6
294:     '+'     shift, and go to state 7
295:     '-'     shift, and go to state 8
296:     '('     shift, and go to state 10
297:
298:     expr  go to state 27
299:
300:
301: state 20
302:
303:    10 expr: expr '/' . expr
304:
305:     IDENT   shift, and go to state 5
```

```
306:     NUMBER  shift, and go to state 6
307:     '+'     shift, and go to state 7
308:     '-'     shift, and go to state 8
309:     '('     shift, and go to state 10
310:
311:     expr  go to state 28
312:
313:
314: state 21
315:
316:    11 expr: expr '^' . expr
317:
318:     IDENT   shift, and go to state 5
319:     NUMBER  shift, and go to state 6
320:     '+'     shift, and go to state 7
321:     '-'     shift, and go to state 8
322:     '('     shift, and go to state 10
323:
324:     expr  go to state 29
325:
326:
327: state 22
328:
329:     2 stmtseq: stmtseq expr ';' .
330:
331:     $default  reduce using rule 2 (stmtseq)
332:
333:
334: state 23
335:
336:    14 expr: '(' expr ')' .
337:
338:     $default  reduce using rule 14 (expr)
339:
340:
341: state 24
342:
343:     6 expr: expr . '=' expr
344:     6     | expr '=' expr .
345:     7     | expr . '+' expr
346:     8     | expr . '-' expr
347:     9     | expr . '*' expr
348:    10     | expr . '/' expr
349:    11     | expr . '^' expr
350:
351:     '='  shift, and go to state 16
352:     '+'  shift, and go to state 17
353:     '-'  shift, and go to state 18
354:     '*'  shift, and go to state 19
355:     '/'  shift, and go to state 20
356:     '^'  shift, and go to state 21
357:
358:     $default  reduce using rule 6 (expr)
359:
360:
361: state 25
362:
363:     6 expr: expr . '=' expr
364:     7     | expr . '+' expr
365:     7     | expr '+' expr .
366:     8     | expr . '-' expr
```

```
367:     9     | expr . '*' expr
368:    10     | expr . '/' expr
369:    11     | expr . '^' expr
370:
371:     '*'  shift, and go to state 19
372:     '/'  shift, and go to state 20
373:     '^'  shift, and go to state 21
374:
375:     $default  reduce using rule 7 (expr)
376:
377:
378: state 26
379:
380:     6 expr: expr . '=' expr
381:     7     | expr . '+' expr
382:     8     | expr . '-' expr
383:     8     | expr '-' expr .
384:     9     | expr . '*' expr
385:    10     | expr . '/' expr
386:    11     | expr . '^' expr
387:
388:     '*'  shift, and go to state 19
389:     '/'  shift, and go to state 20
390:     '^'  shift, and go to state 21
391:
392:     $default  reduce using rule 8 (expr)
393:
394:
395: state 27
396:
397:     6 expr: expr . '=' expr
398:     7     | expr . '+' expr
399:     8     | expr . '-' expr
400:     9     | expr . '*' expr
401:     9     | expr '*' expr .
402:    10     | expr . '/' expr
403:    11     | expr . '^' expr
404:
405:     '^'  shift, and go to state 21
406:
407:     $default  reduce using rule 9 (expr)
408:
409:
410: state 28
411:
412:     6 expr: expr . '=' expr
413:     7     | expr . '+' expr
414:     8     | expr . '-' expr
415:     9     | expr . '*' expr
416:    10     | expr . '/' expr
417:    10     | expr '/' expr .
418:    11     | expr . '^' expr
419:
420:     '^'  shift, and go to state 21
421:
422:     $default  reduce using rule 10 (expr)
423:
424:
425: state 29
426:
427:     6 expr: expr . '=' expr
```

```
428:      7     | expr . '+' expr
429:      8     | expr . '-' expr
430:      9     | expr . '*' expr
431:     10     | expr . '/' expr
432:     11     | expr . '^' expr
433:     11     | expr '^' expr .
434:
435:      '^'  shift, and go to state 21
436:
437:      $default  reduce using rule 11 (expr)
```

```
 1: astree.o:
 2:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
 3:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
 4:      $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
 5:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
 6:      $Id: astree.c,v 1.12 2013-08-22 13:58:43-07 - - $
 7:      $Compiled: astree.c Sep 11 2013 20:17:49 $
 8:
 9: emit.o:
10:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
11:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
12:      $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
13:      $Id: emit.h,v 1.3 2013-08-22 13:58:43-07 - - $
14:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
15:      $Id: emit.c,v 1.6 2013-08-22 13:58:43-07 - - $
16:      $Compiled: emit.c Sep 11 2013 20:17:49 $
17:
18: lyutils.o:
19:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
20:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
21:      $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
22:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
23:      $Id: lyutils.c,v 1.12 2013-08-22 13:58:43-07 - - $
24:      $Compiled: lyutils.c Sep 11 2013 20:17:49 $
25:
26: main.o:
27:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
28:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
29:      $Id: emit.h,v 1.3 2013-08-22 13:58:43-07 - - $
30:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
31:      $Id: main.c,v 1.17 2013-08-22 13:58:43-07 - - $
32:      $Compiled: main.c Sep 11 2013 20:17:49 $
33:
34: auxlib.o:
35:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
36:      $Id: auxlib.c,v 1.16 2013-08-22 13:59:59-07 - - $
37:      $Compiled: auxlib.c Sep 11 2013 20:17:49 $
38:
39: yylex.o:
40:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
41:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
42:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
43:      $Id: yylex.c,v 1.10 2013-09-11 20:17:45-07 - - $
44:      $Compiled: scanner.l Sep 11 2013 20:17:49 $
45:
46: yyparse.o:
47:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
48:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
49:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
50:      $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
51:      $Id: yyparse.c,v 1.5 2013-08-22 14:00:02-07 - - $
52:      $Compiled: parser.y Sep 11 2013 20:17:49 $
53:
54: zexprsm:
55:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
56:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
57:      $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
58:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
59:      $Id: astree.c,v 1.12 2013-08-22 13:58:43-07 - - $
60:      $Compiled: astree.c Sep 11 2013 20:17:49 $
61:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
```

```
62:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
63:      $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
64:      $Id: emit.h,v 1.3 2013-08-22 13:58:43-07 - - $
65:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
66:      $Id: emit.c,v 1.6 2013-08-22 13:58:43-07 - - $
67:      $Compiled: emit.c Sep 11 2013 20:17:49 $
68:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
69:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
70:      $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
71:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
72:      $Id: lyutils.c,v 1.12 2013-08-22 13:58:43-07 - - $
73:      $Compiled: lyutils.c Sep 11 2013 20:17:49 $
74:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
75:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
76:      $Id: emit.h,v 1.3 2013-08-22 13:58:43-07 - - $
77:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
78:      $Id: main.c,v 1.17 2013-08-22 13:58:43-07 - - $
79:      $Compiled: main.c Sep 11 2013 20:17:49 $
80:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
81:      $Id: auxlib.c,v 1.16 2013-08-22 13:59:59-07 - - $
82:      $Compiled: auxlib.c Sep 11 2013 20:17:49 $
83:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
84:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
85:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
86:      $Id: yylex.c,v 1.10 2013-09-11 20:17:45-07 - - $
87:      $Compiled: scanner.l Sep 11 2013 20:17:49 $
88:      $Id: auxlib.h,v 1.10 2013-08-22 13:58:43-07 - - $
89:      $Id: astree.h,v 1.8 2013-08-22 13:58:43-07 - - $
90:      $Id: lyutils.h,v 1.10 2013-08-22 13:58:43-07 - - $
91:      $Id: astree.rep.h,v 1.5 2013-08-22 13:58:43-07 - - $
92:      $Id: yyparse.c,v 1.5 2013-08-22 14:00:02-07 - - $
93:      $Compiled: parser.y Sep 11 2013 20:17:49 $
```

```
1: // $Id: test1.in,v 1.2 2011-08-31 17:54:03-07 - - $
2: a=b*c+d*e;
```

```
 1: ;    1: # 1 "test1.in";# 1 "test1.in"
 2: ;    0:
 3: ;    1: # 1 "<built-in>";# 1 "<built-in>"
 4: ;    0:
 5: ;    1: # 1 "<command-line>";# 1 "<command-line>"
 6: ;    0:
 7: ;    1: # 1 "test1.in";# 1 "test1.in"
 8: ;    0:
 9: ;    1:
10: ;    2: a=b*c+d*e;
11:
12:           pushvar   b                   ; test1.in 2.2
13:           pushvar   c                   ; test1.in 2.4
14:           mul                           ; test1.in 2.3
15:           pushvar   d                   ; test1.in 2.6
16:           pushvar   e                   ; test1.in 2.8
17:           mul                           ; test1.in 2.7
18:           add                           ; test1.in 2.5
19:           popvar    a                   ; test1.in 2.0
```

```
 1: DEBUGF(x): auxlib.c[117] set_debugflags():
 2: Debugflags = "@", all = 1
 3: zexprsm: bad option (d)
 4: DEBUGF(m): main.c[69] scan_opts():
 5: filename = test1.in, yyin = 0x1e22030, fileno (yyin) = 3
 6: Starting parse
 7: Entering state 0
 8: Reducing stack by rule 5 (line 45):
 9: DEBUGF(f): astree.c[37] new_astree():
10: malloc (56) = 0x1e221f0-> 0:0.0: ROOT: 0x1e22230->"<<ROOT>>"
11: -> $$ = nterm stmtseq ()
12: Stack now 0
13: Entering state 2
14: Reading a token: --(end of buffer or a NUL)
15: --accepting rule at line 36 ("# 1 "test1.in"")
16: DEBUGF(m): lyutils.c[118] scanner_include():
17: filename=test1.in, scan_linenr=0
18: --accepting rule at line 38 ("
19: ")
20: --accepting rule at line 36 ("# 1 "<built-in>"")
21: DEBUGF(m): lyutils.c[118] scanner_include():
22: filename=<built-in>, scan_linenr=0
23: --accepting rule at line 38 ("
24: ")
25: --accepting rule at line 36 ("# 1 "<command-line>"")
26: DEBUGF(m): lyutils.c[118] scanner_include():
27: filename=<command-line>, scan_linenr=0
28: --accepting rule at line 38 ("
29: ")
30: --accepting rule at line 36 ("# 1 "test1.in"")
31: DEBUGF(m): lyutils.c[118] scanner_include():
32: filename=test1.in, scan_linenr=0
33: --accepting rule at line 38 ("
34: ")
35: --accepting rule at line 38 ("
36: ")
37: --accepting rule at line 41 ("a")
38: DEBUGF(f): astree.c[37] new_astree():
39: malloc (56) = 0x1e263d0-> 4:2.0: IDENT: 0x1e26410->"a"
40: Next token is token IDENT (259=IDENT)
41:          (0x1e263d0-> astree {IDENT(259), 4:2.000, 0x1e26410->"a",
42:                       first=(nil), last=(nil), next=(nil)})
43: Shifting token IDENT (259=IDENT)
44:          (0x1e263d0-> astree {IDENT(259), 4:2.000, 0x1e26410->"a",
45:                       first=(nil), last=(nil), next=(nil)})
46: Entering state 5
47: Reducing stack by rule 15 (line 57):
48:    $1 = token IDENT (259=IDENT)
49:          (0x1e263d0-> astree {IDENT(259), 4:2.000, 0x1e26410->"a",
50:                       first=(nil), last=(nil), next=(nil)})
51: -> $$ = nterm expr ()
52: Stack now 0 2
53: Entering state 11
54: Reading a token: --accepting rule at line 43 ("=")
55: DEBUGF(f): astree.c[37] new_astree():
56: malloc (56) = 0x1e26430-> 4:2.1: '=': 0x1e26470->"="
57: Next token is token '=' (61='=')
58:          (0x1e26430-> astree {'='(61), 4:2.001, 0x1e26470->"=",
59:                       first=(nil), last=(nil), next=(nil)})
60: Shifting token '=' (61='=')
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():
```

```
123: 0x1e264f0 (*) adopting 0x1e26550 (c)
124: -> $$ = nterm expr ()
125: Stack now 0 2 11 16
126: Entering state 24
127: Next token is token '+' (43='+')
128:          (0x1e265b0-> astree {'+'(43), 4:2.005, 0x1e265f0->"+",
129:                       first=(nil), last=(nil), next=(nil)})
130: Shifting token '+' (43='+')
131:          (0x1e265b0-> astree {'+'(43), 4:2.005, 0x1e265f0->"+",
132:                       first=(nil), last=(nil), next=(nil)})
133: Entering state 17
134: Reading a token: --accepting rule at line 41 ("d")
135: DEBUGF(f): astree.c[37] new_astree():
136: malloc (56) = 0x1e26610-> 4:2.6: IDENT: 0x1e26650->"d"
137: Next token is token IDENT (259=IDENT)
138:          (0x1e26610-> astree {IDENT(259), 4:2.006, 0x1e26650->"d",
139:                       first=(nil), last=(nil), next=(nil)})
140: Shifting token IDENT (259=IDENT)
141:          (0x1e26610-> astree {IDENT(259), 4:2.006, 0x1e26650->"d",
142:                       first=(nil), last=(nil), next=(nil)})
143: Entering state 5
144: Reducing stack by rule 15 (line 57):
145:    $1 = token IDENT (259=IDENT)
146:          (0x1e26610-> astree {IDENT(259), 4:2.006, 0x1e26650->"d",
147:                       first=(nil), last=(nil), next=(nil)})
148: -> $$ = nterm expr ()
149: Stack now 0 2 11 16 24 17
150: Entering state 25
151: Reading a token: --accepting rule at line 46 ("*")
152: DEBUGF(f): astree.c[37] new_astree():
153: malloc (56) = 0x1e26670-> 4:2.7: '*': 0x1e266b0->"*"
154: Next token is token '*' (42='*')
155:          (0x1e26670-> astree {'*'(42), 4:2.007, 0x1e266b0->"*",
156:                       first=(nil), last=(nil), next=(nil)})
157: Shifting token '*' (42='*')
158:          (0x1e26670-> astree {'*'(42), 4:2.007, 0x1e266b0->"*",
159:                       first=(nil), last=(nil), next=(nil)})
160: Entering state 19
161: Reading a token: --accepting rule at line 41 ("e")
162: DEBUGF(f): astree.c[37] new_astree():
163: malloc (56) = 0x1e266d0-> 4:2.8: IDENT: 0x1e26710->"e"
164: Next token is token IDENT (259=IDENT)
165:          (0x1e266d0-> astree {IDENT(259), 4:2.008, 0x1e26710->"e",
166:                       first=(nil), last=(nil), next=(nil)})
167: Shifting token IDENT (259=IDENT)
168:          (0x1e266d0-> astree {IDENT(259), 4:2.008, 0x1e26710->"e",
169:                       first=(nil), last=(nil), next=(nil)})
170: Entering state 5
171: Reducing stack by rule 15 (line 57):
172:    $1 = token IDENT (259=IDENT)
173:          (0x1e266d0-> astree {IDENT(259), 4:2.008, 0x1e26710->"e",
174:                       first=(nil), last=(nil), next=(nil)})
175: -> $$ = nterm expr ()
176: Stack now 0 2 11 16 24 17 25 19
177: Entering state 27
178: Reading a token: --accepting rule at line 51 (";")
179: DEBUGF(f): astree.c[37] new_astree():
180: malloc (56) = 0x1e26730-> 4:2.9: ';': 0x1e26770->";"
181: Next token is token ';' (59=';')
182:          (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
183:                       first=(nil), last=(nil), next=(nil)})
```

```
184: Reducing stack by rule 9 (line 51):
185:    $1 = nterm expr ()
186:    $2 = token '*' (42='*')
187:          (0x1e26670-> astree {'*'(42), 4:2.007, 0x1e266b0->"*",
188:                       first=(nil), last=(nil), next=(nil)})
189:    $3 = nterm expr ()
190: DEBUGF(a): astree.c[55] adopt():
191: 0x1e26670 (*) adopting 0x1e26610 (d)
192: DEBUGF(a): astree.c[55] adopt():
193: 0x1e26670 (*) adopting 0x1e266d0 (e)
194: -> $$ = nterm expr ()
195: Stack now 0 2 11 16 24 17
196: Entering state 25
197: Next token is token ';' (59=';')
198:          (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
199:                       first=(nil), last=(nil), next=(nil)})
200: Reducing stack by rule 7 (line 49):
201:    $1 = nterm expr ()
202:    $2 = token '+' (43='+')
203:          (0x1e265b0-> astree {'+'(43), 4:2.005, 0x1e265f0->"+",
204:                       first=(nil), last=(nil), next=(nil)})
205:    $3 = nterm expr ()
206: DEBUGF(a): astree.c[55] adopt():
207: 0x1e265b0 (+) adopting 0x1e264f0 (*)
208: DEBUGF(a): astree.c[55] adopt():
209: 0x1e265b0 (+) adopting 0x1e26670 (*)
210: -> $$ = nterm expr ()
211: Stack now 0 2 11 16
212: Entering state 24
213: Next token is token ';' (59=';')
214:          (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
215:                       first=(nil), last=(nil), next=(nil)})
216: Reducing stack by rule 6 (line 48):
217:    $1 = nterm expr ()
218:    $2 = token '=' (61='=')
219:          (0x1e26430-> astree {'='(61), 4:2.001, 0x1e26470->"=",
220:                       first=(nil), last=(nil), next=(nil)})
221:    $3 = nterm expr ()
222: DEBUGF(a): astree.c[55] adopt():
223: 0x1e26430 (=) adopting 0x1e263d0 (a)
224: DEBUGF(a): astree.c[55] adopt():
225: 0x1e26430 (=) adopting 0x1e265b0 (+)
226: -> $$ = nterm expr ()
227: Stack now 0 2
228: Entering state 11
229: Next token is token ';' (59=';')
230:          (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
231:                       first=(nil), last=(nil), next=(nil)})
232: Shifting token ';' (59=';')
233:          (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
234:                       first=(nil), last=(nil), next=(nil)})
235: Entering state 22
236: Reducing stack by rule 2 (line 42):
237:    $1 = nterm stmtseq ()
238:    $2 = nterm expr ()
239:    $3 = token ';' (59=';')
240:          (0x1e26730-> astree {';'(59), 4:2.009, 0x1e26770->";",
241:                       first=(nil), last=(nil), next=(nil)})
242: DEBUGF(f): astree.c[127] freeast():
243: free [1E26730]-> 4:2.9: ';': 0x1e26770->";")
244: DEBUGF(a): astree.c[55] adopt():
```

```
245: 0x1e221f0 (<<ROOT>>) adopting 0x1e26430 (=)
246: -> $$ = nterm stmtseq ()
247: Stack now 0
248: Entering state 2
249: Reading a token: --accepting rule at line 38 ("
250: ")
251: --(end of buffer or a NUL)
252: --EOF (start condition 0)
253: Now at end of input.
254: Reducing stack by rule 1 (line 39):
255:    $1 = nterm stmtseq ()
256: -> $$ = nterm program ()
257: Stack now 0
258: Entering state 1
259: Now at end of input.
260: Shifting token $end (0=$end)
261:          (yyvaluep = 0x1e26730)
262: Entering state 3
263: Stack now 0 1 3
264: Cleanup: popping token $end (0=$end)
265:          (yyvaluep = 0x1e26730)
266: Cleanup: popping nterm program ()
267: DEBUGF(a): main.c[88] main():
268:
269: <<ROOT>> 0x1e221f0-> astree {ROOT(258), 0:0.000, 0x1e22230->"<<ROOT>>",
270:              first=0x1e26430, last=0x1e26430, next=(nil)}
271:    = 0x1e26430-> astree {'='(61), 4:2.001, 0x1e26470->"=",
272:                 first=0x1e263d0, last=0x1e265b0, next=(nil)}
273:       a 0x1e263d0-> astree {IDENT(259), 4:2.000, 0x1e26410->"a",
274:                    first=(nil), last=(nil), next=0x1e265b0}
275:       + 0x1e265b0-> astree {'+'(43), 4:2.005, 0x1e265f0->"+",
276:                    first=0x1e264f0, last=0x1e26670, next=(nil)}
277:          * 0x1e264f0-> astree {'*'(42), 4:2.003, 0x1e26530->"*",
278:                       first=0x1e26490, last=0x1e26550, next=0x1e26670}
279:             b 0x1e26490-> astree {IDENT(259), 4:2.002, 0x1e264d0->"b",
280:                          first=(nil), last=(nil), next=0x1e26550}
281:             c 0x1e26550-> astree {IDENT(259), 4:2.004, 0x1e26590->"c",
282:                          first=(nil), last=(nil), next=(nil)}
283:          * 0x1e26670-> astree {'*'(42), 4:2.007, 0x1e266b0->"*",
284:                       first=0x1e26610, last=0x1e266d0, next=(nil)}
285:             d 0x1e26610-> astree {IDENT(259), 4:2.006, 0x1e26650->"d",
286:                          first=(nil), last=(nil), next=0x1e266d0}
287:             e 0x1e266d0-> astree {IDENT(259), 4:2.008, 0x1e26710->"e",
288:                          first=(nil), last=(nil), next=(nil)}
289: DEBUGF(f): astree.c[127] freeast():
290: free [1E263D0]-> 4:2.0: IDENT: 0x1e26410->"a")
291: DEBUGF(f): astree.c[127] freeast():
292: free [1E26490]-> 4:2.2: IDENT: 0x1e264d0->"b")
293: DEBUGF(f): astree.c[127] freeast():
294: free [1E26550]-> 4:2.4: IDENT: 0x1e26590->"c")
295: DEBUGF(f): astree.c[127] freeast():
296: free [1E264F0]-> 4:2.3: '*': 0x1e26530->"*")
297: DEBUGF(f): astree.c[127] freeast():
298: free [1E26610]-> 4:2.6: IDENT: 0x1e26650->"d")
299: DEBUGF(f): astree.c[127] freeast():
300: free [1E266D0]-> 4:2.8: IDENT: 0x1e26710->"e")
301: DEBUGF(f): astree.c[127] freeast():
302: free [1E26670]-> 4:2.7: '*': 0x1e266b0->"*")
303: DEBUGF(f): astree.c[127] freeast():
304: free [1E265B0]-> 4:2.5: '+': 0x1e265f0->"+")
305: DEBUGF(f): astree.c[127] freeast():
```

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

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

```
1:  t// $Id: test4.in,v 1.4 2012-10-11 19:07:58-07 - - $
2: #include "test4a.inh"
3: #include "test4b.inh"
4: 3*4;
```

```
 1: ;    1: # 1 "test4.in";# 1 "test4.in"
 2: ;    0:
 3: ;    1: # 1 "<built-in>";# 1 "<built-in>"
 4: ;    0:
 5: ;    1: # 1 "<command-line>";# 1 "<command-line>"
 6: ;    0:
 7: ;    1: # 1 "test4.in";# 1 "test4.in"
 8: ;    0:
 9: ;    1:  t
10: ;    2: # 1 "test4a.inh" 1;# 1 "test4a.inh"
11: ;    0:
12: ;    1:
13: ;    2:
14: ;    3: pi=3.14159265358979323846264338328O;
15: ;    4: pi;
16: ;    5: # 3 "test4.in" 2;# 3 "test4.in"
17: ;    2:
18: ;    3: # 1 "test4b.inh" 1;# 1 "test4b.inh"
19: ;    0:
20: ;    1:
21: ;    2: a=pi*r^2;
22: ;    3: 3.14159265358979323846264338328O;
23: ;    4: # 4 "test4.in" 2;# 4 "test4.in"
24: ;    3:
25: ;    4: 3*4;
26:
27:           pushvar   pi                  ; test4a.inh 4.0
28:           pushvar   pi                  ; test4b.inh 2.2
29:           pushvar   r                   ; test4b.inh 2.5
30:           pushnum   2                   ; test4b.inh 2.7
31:           pow                           ; test4b.inh 2.6
32:           mul                           ; test4b.inh 2.4
33:           popvar    a                   ; test4b.inh 2.0
34:           pushnum   3.14159265358979323846264338328O; test4b.inh 3.0
35:           pushnum   3                   ; test4.in 4.0
36:           pushnum   4                   ; test4.in 4.2
37:           mul                           ; test4.in 4.1
```

```
 1: DEBUGF(x): auxlib.c[117] set_debugflags():
 2: Debugflags = "@", all = 1
 3: zexprsm: bad option (d)
 4: DEBUGF(m): main.c[69] scan_opts():
 5: filename = test4.in, yyin = 0x661030, fileno (yyin) = 3
 6: Starting parse
 7: Entering state 0
 8: Reducing stack by rule 5 (line 45):
 9: DEBUGF(f): astree.c[37] new_astree():
10: malloc (56) = 0x6611f0-> 0:0.0: ROOT: 0x661230->"<<ROOT>>"
11: -> $$ = nterm stmtseq ()
12: Stack now 0
13: Entering state 2
14: Reading a token: --(end of buffer or a NUL)
15: --accepting rule at line 36 ("# 1 "test4.in"")
16: DEBUGF(m): lyutils.c[118] scanner_include():
17: filename=test4.in, scan_linenr=0
18: --accepting rule at line 38 ("
19: ")
20: --accepting rule at line 36 ("# 1 "<built-in>"")
21: DEBUGF(m): lyutils.c[118] scanner_include():
22: filename=<built-in>, scan_linenr=0
23: --accepting rule at line 38 ("
24: ")
25: --accepting rule at line 36 ("# 1 "<command-line>"")
26: DEBUGF(m): lyutils.c[118] scanner_include():
27: filename=<command-line>, scan_linenr=0
28: --accepting rule at line 38 ("
29: ")
30: --accepting rule at line 36 ("# 1 "test4.in"")
31: DEBUGF(m): lyutils.c[118] scanner_include():
32: filename=test4.in, scan_linenr=0
33: --accepting rule at line 38 ("
34: ")
35: --accepting rule at line 37 (" ")
36: --accepting rule at line 41 ("t")
37: DEBUGF(f): astree.c[37] new_astree():
38: malloc (56) = 0x6653d0-> 4:1.1: IDENT: 0x665410->"t"
39: Next token is token IDENT (259=IDENT)
40:          (0x6653d0-> astree {IDENT(259), 4:1.001, 0x665410->"t",
41:                      first=(nil), last=(nil), next=(nil)})
42: Shifting token IDENT (259=IDENT)
43:          (0x6653d0-> astree {IDENT(259), 4:1.001, 0x665410->"t",
44:                      first=(nil), last=(nil), next=(nil)})
45: Entering state 5
46: Reducing stack by rule 15 (line 57):
47:    $1 = token IDENT (259=IDENT)
48:          (0x6653d0-> astree {IDENT(259), 4:1.001, 0x665410->"t",
49:                      first=(nil), last=(nil), next=(nil)})
50: -> $$ = nterm expr ()
51: Stack now 0 2
52: Entering state 11
53: Reading a token: --accepting rule at line 38 ("
54: ")
55: --accepting rule at line 36 ("# 1 "test4a.inh" 1")
56: DEBUGF(m): lyutils.c[118] scanner_include():
57: filename=test4a.inh, scan_linenr=0
58: --accepting rule at line 38 ("
59: ")
60: --accepting rule at line 38 ("
61: ")
```

```
   62: --accepting rule at line 38 ("
   63: ")
   64: --accepting rule at line 41 ("pi")
   65: DEBUGF(f): astree.c[37] new_astree():
   66: malloc (56) = 0x665470-> 5:3.0: IDENT: 0x6654b0->"pi"
   67: Next token is token IDENT (259=IDENT)
   68:          (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
   69:                      first=(nil), last=(nil), next=(nil)})
   70: zexprsm: test4a.inh: 3: syntax error, unexpected IDENT
   71: Error: popping nterm expr ()
   72: Stack now 0 2
   73: Shifting token error (256=error)
   74:          (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
   75:                      first=(nil), last=(nil), next=(nil)})
   76: Entering state 4
   77: Next token is token IDENT (259=IDENT)
   78:          (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
   79:                      first=(nil), last=(nil), next=(nil)})
   80: Error: discarding token IDENT (259=IDENT)
   81:          (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
   82:                      first=(nil), last=(nil), next=(nil)})
   83: Error: popping token error (256=error)
   84:          (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
   85:                      first=(nil), last=(nil), next=(nil)})
   86: Stack now 0 2
   87: Shifting token error (256=error)
   88:          (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
   89:                      first=(nil), last=(nil), next=(nil)})
   90: Entering state 4
   91: Reading a token: --accepting rule at line 43 ("=")
   92: DEBUGF(f): astree.c[37] new_astree():
   93: malloc (56) = 0x6654d0-> 5:3.2: '=': 0x665510->"="
   94: Next token is token '=' (61='=')
   95:          (0x6654d0-> astree {'='(61), 5:3.002, 0x665510->"=",
   96:                      first=(nil), last=(nil), next=(nil)})
   97: Error: discarding token '=' (61='=')
   98:          (0x6654d0-> astree {'='(61), 5:3.002, 0x665510->"=",
   99:                      first=(nil), last=(nil), next=(nil)})
  100: Error: popping token error (256=error)
  101:          (0x665470-> astree {IDENT(259), 5:3.000, 0x6654b0->"pi",
  102:                      first=(nil), last=(nil), next=(nil)})
  103: Stack now 0 2
  104: Shifting token error (256=error)
  105:          (0x6654d0-> astree {'='(61), 5:3.002, 0x665510->"=",
  106:                      first=(nil), last=(nil), next=(nil)})
  107: Entering state 4
  108: Reading a token: --accepting rule at line 40 ("3.14159265358979323846264338
280")
  109: DEBUGF(f): astree.c[37] new_astree():
  110: malloc (56) = 0x665530-> 5:3.3: NUMBER: 0x665570->"3.14159265358979323846264
3383280"
  111: Next token is token NUMBER (260=NUMBER)
  112:          (0x665530-> astree {NUMBER(260), 5:3.003, 0x665570->"3.141592653589
793238462643383280",
  113:                      first=(nil), last=(nil), next=(nil)})
  114: Error: discarding token NUMBER (260=NUMBER)
  115:          (0x665530-> astree {NUMBER(260), 5:3.003, 0x665570->"3.141592653589
793238462643383280",
  116:                      first=(nil), last=(nil), next=(nil)})
  117: Error: popping token error (256=error)
  118:          (0x6654d0-> astree {'='(61), 5:3.002, 0x665510->"=",
```

```
  119:                        first=(nil), last=(nil), next=(nil)})
  120: Stack now 0 2
  121: Shifting token error (256=error)
  122:          (0x665530-> astree {NUMBER(260), 5:3.003, 0x665570->"3.141592653589
793238462643383280",
  123:                        first=(nil), last=(nil), next=(nil)})
  124: Entering state 4
  125: Reading a token: --accepting rule at line 51 (";")
  126: DEBUGF(f): astree.c[37] new_astree():
  127: malloc (56) = 0x6655a0-> 5:3.35: ';': 0x6655e0->";"
  128: Next token is token ';' (59=';')
  129:          (0x6655a0-> astree {';'(59), 5:3.035, 0x6655e0->";",
  130:                        first=(nil), last=(nil), next=(nil)})
  131: Shifting token ';' (59=';')
  132:          (0x6655a0-> astree {';'(59), 5:3.035, 0x6655e0->";",
  133:                        first=(nil), last=(nil), next=(nil)})
  134: Entering state 12
  135: Reducing stack by rule 3 (line 43):
  136:    $1 = nterm stmtseq ()
  137:    $2 = token error (256=error)
  138:          (0x665530-> astree {NUMBER(260), 5:3.003, 0x665570->"3.141592653589
793238462643383280",
  139:                        first=(nil), last=(nil), next=(nil)})
  140:    $3 = token ';' (59=';')
  141:          (0x6655a0-> astree {';'(59), 5:3.035, 0x6655e0->";",
  142:                        first=(nil), last=(nil), next=(nil)})
  143: DEBUGF(f): astree.c[127] freeast():
  144: free [6655A0]-> 5:3.35: ';': 0x6655e0->";")
  145: -> $$ = nterm stmtseq ()
  146: Stack now 0
  147: Entering state 2
  148: Reading a token: --accepting rule at line 38 ("
  149: ")
  150: --accepting rule at line 41 ("pi")
  151: DEBUGF(f): astree.c[37] new_astree():
  152: malloc (56) = 0x6655a0-> 5:4.0: IDENT: 0x6655e0->"pi"
  153: Next token is token IDENT (259=IDENT)
  154:          (0x6655a0-> astree {IDENT(259), 5:4.000, 0x6655e0->"pi",
  155:                        first=(nil), last=(nil), next=(nil)})
  156: Shifting token IDENT (259=IDENT)
  157:          (0x6655a0-> astree {IDENT(259), 5:4.000, 0x6655e0->"pi",
  158:                        first=(nil), last=(nil), next=(nil)})
  159: Entering state 5
  160: Reducing stack by rule 15 (line 57):
  161:    $1 = token IDENT (259=IDENT)
  162:          (0x6655a0-> astree {IDENT(259), 5:4.000, 0x6655e0->"pi",
  163:                        first=(nil), last=(nil), next=(nil)})
  164: -> $$ = nterm expr ()
  165: Stack now 0 2
  166: Entering state 11
  167: Reading a token: --accepting rule at line 51 (";")
  168: DEBUGF(f): astree.c[37] new_astree():
  169: malloc (56) = 0x665600-> 5:4.2: ';': 0x665640->";"
  170: Next token is token ';' (59=';')
  171:          (0x665600-> astree {';'(59), 5:4.002, 0x665640->";",
  172:                        first=(nil), last=(nil), next=(nil)})
  173: Shifting token ';' (59=';')
  174:          (0x665600-> astree {';'(59), 5:4.002, 0x665640->";",
  175:                        first=(nil), last=(nil), next=(nil)})
  176: Entering state 22
  177: Reducing stack by rule 2 (line 42):
```

```
178:    $1 = nterm stmtseq ()
179:    $2 = nterm expr ()
180:    $3 = token ';' (59=';')
181:          (0x665600-> astree {';'(59), 5:4.002, 0x665640->";",
182:                      first=(nil), last=(nil), next=(nil)})
183: DEBUGF(f): astree.c[127] freeast():
184: free [665600]-> 5:4.2: ';': 0x665640->";")
185: DEBUGF(a): astree.c[55] adopt():
186: 0x6611f0 (<<ROOT>>) adopting 0x6655a0 (pi)
187: -> $$ = nterm stmtseq ()
188: Stack now 0
189: Entering state 2
190: Reading a token: --accepting rule at line 38 ("
191: ")
192: --accepting rule at line 36 ("# 3 "test4.in" 2")
193: DEBUGF(m): lyutils.c[118] scanner_include():
194: filename=test4.in, scan_linenr=2
195: --accepting rule at line 38 ("
196: ")
197: --accepting rule at line 36 ("# 1 "test4b.inh" 1")
198: DEBUGF(m): lyutils.c[118] scanner_include():
199: filename=test4b.inh, scan_linenr=0
200: --accepting rule at line 38 ("
201: ")
202: --accepting rule at line 38 ("
203: ")
204: --accepting rule at line 41 ("a")
205: DEBUGF(f): astree.c[37] new_astree():
206: malloc (56) = 0x665600-> 7:2.0: IDENT: 0x6656c0->"a"
207: Next token is token IDENT (259=IDENT)
208:          (0x665600-> astree {IDENT(259), 7:2.000, 0x6656c0->"a",
209:                      first=(nil), last=(nil), next=(nil)})
210: Shifting token IDENT (259=IDENT)
211:          (0x665600-> astree {IDENT(259), 7:2.000, 0x6656c0->"a",
212:                      first=(nil), last=(nil), next=(nil)})
213: Entering state 5
214: Reducing stack by rule 15 (line 57):
215:    $1 = token IDENT (259=IDENT)
216:          (0x665600-> astree {IDENT(259), 7:2.000, 0x6656c0->"a",
217:                      first=(nil), last=(nil), next=(nil)})
218: -> $$ = nterm expr ()
219: Stack now 0 2
220: Entering state 11
221: Reading a token: --accepting rule at line 43 ("=")
222: DEBUGF(f): astree.c[37] new_astree():
223: malloc (56) = 0x6656e0-> 7:2.1: '=': 0x665720->"="
224: Next token is token '=' (61='=')
225:          (0x6656e0-> astree {'='(61), 7:2.001, 0x665720->"=",
226:                      first=(nil), last=(nil), next=(nil)})
227: Shifting token '=' (61='=')
228:          (0x6656e0-> astree {'='(61), 7:2.001, 0x665720->"=",
229:                      first=(nil), last=(nil), next=(nil)})
230: Entering state 16
231: Reading a token: --accepting rule at line 41 ("pi")
232: DEBUGF(f): astree.c[37] new_astree():
233: malloc (56) = 0x665740-> 7:2.2: IDENT: 0x665780->"pi"
234: Next token is token IDENT (259=IDENT)
235:          (0x665740-> astree {IDENT(259), 7:2.002, 0x665780->"pi",
236:                      first=(nil), last=(nil), next=(nil)})
237: Shifting token IDENT (259=IDENT)
238:          (0x665740-> astree {IDENT(259), 7:2.002, 0x665780->"pi",
```

```
239:                        first=(nil), last=(nil), next=(nil)})
240: Entering state 5
241: Reducing stack by rule 15 (line 57):
242:    $1 = token IDENT (259=IDENT)
243:          (0x665740-> astree {IDENT(259), 7:2.002, 0x665780->"pi",
244:                      first=(nil), last=(nil), next=(nil)})
245: -> $$ = nterm expr ()
246: Stack now 0 2 11 16
247: Entering state 24
248: Reading a token: --accepting rule at line 46 ("*")
249: DEBUGF(f): astree.c[37] new_astree():
250: malloc (56) = 0x6657a0-> 7:2.4: '*': 0x6657e0->"*"
251: Next token is token '*' (42='*')
252:          (0x6657a0-> astree {'*'(42), 7:2.004, 0x6657e0->"*",
253:                      first=(nil), last=(nil), next=(nil)})
254: Shifting token '*' (42='*')
255:          (0x6657a0-> astree {'*'(42), 7:2.004, 0x6657e0->"*",
256:                      first=(nil), last=(nil), next=(nil)})
257: Entering state 19
258: Reading a token: --accepting rule at line 41 ("r")
259: DEBUGF(f): astree.c[37] new_astree():
260: malloc (56) = 0x665800-> 7:2.5: IDENT: 0x665840->"r"
261: Next token is token IDENT (259=IDENT)
262:          (0x665800-> astree {IDENT(259), 7:2.005, 0x665840->"r",
263:                      first=(nil), last=(nil), next=(nil)})
264: Shifting token IDENT (259=IDENT)
265:          (0x665800-> astree {IDENT(259), 7:2.005, 0x665840->"r",
266:                      first=(nil), last=(nil), next=(nil)})
267: Entering state 5
268: Reducing stack by rule 15 (line 57):
269:    $1 = token IDENT (259=IDENT)
270:          (0x665800-> astree {IDENT(259), 7:2.005, 0x665840->"r",
271:                      first=(nil), last=(nil), next=(nil)})
272: -> $$ = nterm expr ()
273: Stack now 0 2 11 16 24 19
274: Entering state 27
275: Reading a token: --accepting rule at line 48 ("^")
276: DEBUGF(f): astree.c[37] new_astree():
277: malloc (56) = 0x665860-> 7:2.6: '^': 0x6658a0->"^"
278: Next token is token '^' (94='^')
279:          (0x665860-> astree {'^'(94), 7:2.006, 0x6658a0->"^",
280:                      first=(nil), last=(nil), next=(nil)})
281: Shifting token '^' (94='^')
282:          (0x665860-> astree {'^'(94), 7:2.006, 0x6658a0->"^",
283:                      first=(nil), last=(nil), next=(nil)})
284: Entering state 21
285: Reading a token: --accepting rule at line 40 ("2")
286: DEBUGF(f): astree.c[37] new_astree():
287: malloc (56) = 0x6658c0-> 7:2.7: NUMBER: 0x665900->"2"
288: Next token is token NUMBER (260=NUMBER)
289:          (0x6658c0-> astree {NUMBER(260), 7:2.007, 0x665900->"2",
290:                      first=(nil), last=(nil), next=(nil)})
291: Shifting token NUMBER (260=NUMBER)
292:          (0x6658c0-> astree {NUMBER(260), 7:2.007, 0x665900->"2",
293:                      first=(nil), last=(nil), next=(nil)})
294: Entering state 6
295: Reducing stack by rule 16 (line 58):
296:    $1 = token NUMBER (260=NUMBER)
297:          (0x6658c0-> astree {NUMBER(260), 7:2.007, 0x665900->"2",
298:                      first=(nil), last=(nil), next=(nil)})
299: -> $$ = nterm expr ()
```

```
300: Stack now 0 2 11 16 24 19 27 21
301: Entering state 29
302: Reading a token: --accepting rule at line 51 (";")
303: DEBUGF(f): astree.c[37] new_astree():
304: malloc (56) = 0x665920-> 7:2.8: ';': 0x665960->";"
305: Next token is token ';' (59=';')
306:          (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
307:                      first=(nil), last=(nil), next=(nil)})
308: Reducing stack by rule 11 (line 53):
309:    $1 = nterm expr ()
310:    $2 = token '^' (94='^')
311:          (0x665860-> astree {'^'(94), 7:2.006, 0x6658a0->"^",
312:                      first=(nil), last=(nil), next=(nil)})
313:    $3 = nterm expr ()
314: DEBUGF(a): astree.c[55] adopt():
315: 0x665860 (^) adopting 0x665800 (r)
316: DEBUGF(a): astree.c[55] adopt():
317: 0x665860 (^) adopting 0x6658c0 (2)
318: -> $$ = nterm expr ()
319: Stack now 0 2 11 16 24 19
320: Entering state 27
321: Next token is token ';' (59=';')
322:          (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
323:                      first=(nil), last=(nil), next=(nil)})
324: Reducing stack by rule 9 (line 51):
325:    $1 = nterm expr ()
326:    $2 = token '*' (42='*')
327:          (0x6657a0-> astree {'*'(42), 7:2.004, 0x6657e0->"*",
328:                      first=(nil), last=(nil), next=(nil)})
329:    $3 = nterm expr ()
330: DEBUGF(a): astree.c[55] adopt():
331: 0x6657a0 (*) adopting 0x665740 (pi)
332: DEBUGF(a): astree.c[55] adopt():
333: 0x6657a0 (*) adopting 0x665860 (^)
334: -> $$ = nterm expr ()
335: Stack now 0 2 11 16
336: Entering state 24
337: Next token is token ';' (59=';')
338:          (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
339:                      first=(nil), last=(nil), next=(nil)})
340: Reducing stack by rule 6 (line 48):
341:    $1 = nterm expr ()
342:    $2 = token '=' (61='=')
343:          (0x6656e0-> astree {'='(61), 7:2.001, 0x665720->"=",
344:                      first=(nil), last=(nil), next=(nil)})
345:    $3 = nterm expr ()
346: DEBUGF(a): astree.c[55] adopt():
347: 0x6656e0 (=) adopting 0x665600 (a)
348: DEBUGF(a): astree.c[55] adopt():
349: 0x6656e0 (=) adopting 0x6657a0 (*)
350: -> $$ = nterm expr ()
351: Stack now 0 2
352: Entering state 11
353: Next token is token ';' (59=';')
354:          (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
355:                      first=(nil), last=(nil), next=(nil)})
356: Shifting token ';' (59=';')
357:          (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
358:                      first=(nil), last=(nil), next=(nil)})
359: Entering state 22
360: Reducing stack by rule 2 (line 42):
```

```
  361:    $1 = nterm stmtseq ()
  362:    $2 = nterm expr ()
  363:    $3 = token ';' (59=';')
  364:          (0x665920-> astree {';'(59), 7:2.008, 0x665960->";",
  365:                      first=(nil), last=(nil), next=(nil)})
  366: DEBUGF(f): astree.c[127] freeast():
  367: free [665920]-> 7:2.8: ';': 0x665960->";")
  368: DEBUGF(a): astree.c[55] adopt():
  369: 0x6611f0 (<<ROOT>>) adopting 0x6656e0 (=)
  370: -> $$ = nterm stmtseq ()
  371: Stack now 0
  372: Entering state 2
  373: Reading a token: --accepting rule at line 38 ("
  374: ")
  375: --accepting rule at line 40 ("3.14159265358979323846264338328")
  376: DEBUGF(f): astree.c[37] new_astree():
  377: malloc (56) = 0x665920-> 7:3.0: NUMBER: 0x665980->"3.14159265358979323846264
3383280"
  378: Next token is token NUMBER (260=NUMBER)
  379:          (0x665920-> astree {NUMBER(260), 7:3.000, 0x665980->"3.141592653589
793238462643383280",
  380:                      first=(nil), last=(nil), next=(nil)})
  381: Shifting token NUMBER (260=NUMBER)
  382:          (0x665920-> astree {NUMBER(260), 7:3.000, 0x665980->"3.141592653589
793238462643383280",
  383:                      first=(nil), last=(nil), next=(nil)})
  384: Entering state 6
  385: Reducing stack by rule 16 (line 58):
  386:    $1 = token NUMBER (260=NUMBER)
  387:          (0x665920-> astree {NUMBER(260), 7:3.000, 0x665980->"3.141592653589
793238462643383280",
  388:                      first=(nil), last=(nil), next=(nil)})
  389: -> $$ = nterm expr ()
  390: Stack now 0 2
  391: Entering state 11
  392: Reading a token: --accepting rule at line 51 (";")
  393: DEBUGF(f): astree.c[37] new_astree():
  394: malloc (56) = 0x6659b0-> 7:3.32: ';': 0x665960->";"
  395: Next token is token ';' (59=';')
  396:          (0x6659b0-> astree {';'(59), 7:3.032, 0x665960->";",
  397:                      first=(nil), last=(nil), next=(nil)})
  398: Shifting token ';' (59=';')
  399:          (0x6659b0-> astree {';'(59), 7:3.032, 0x665960->";",
  400:                      first=(nil), last=(nil), next=(nil)})
  401: Entering state 22
  402: Reducing stack by rule 2 (line 42):
  403:    $1 = nterm stmtseq ()
  404:    $2 = nterm expr ()
  405:    $3 = token ';' (59=';')
  406:          (0x6659b0-> astree {';'(59), 7:3.032, 0x665960->";",
  407:                      first=(nil), last=(nil), next=(nil)})
  408: DEBUGF(f): astree.c[127] freeast():
  409: free [6659B0]-> 7:3.32: ';': 0x665960->";")
  410: DEBUGF(a): astree.c[55] adopt():
  411: 0x6611f0 (<<ROOT>>) adopting 0x665920 (3.14159265358979323846264338328)
  412: -> $$ = nterm stmtseq ()
  413: Stack now 0
  414: Entering state 2
  415: Reading a token: --accepting rule at line 38 ("
  416: ")
  417: --accepting rule at line 36 ("# 4 "test4.in" 2")
```

```
418: DEBUGF(m): lyutils.c[118] scanner_include():
419: filename=test4.in, scan_linenr=3
420: --accepting rule at line 38 ("
421: ")
422: --accepting rule at line 40 ("3")
423: DEBUGF(f): astree.c[37] new_astree():
424: malloc (56) = 0x6659b0-> 8:4.0: NUMBER: 0x665a10->"3"
425: Next token is token NUMBER (260=NUMBER)
426:          (0x6659b0-> astree {NUMBER(260), 8:4.000, 0x665a10->"3",
427:                      first=(nil), last=(nil), next=(nil)})
428: Shifting token NUMBER (260=NUMBER)
429:          (0x6659b0-> astree {NUMBER(260), 8:4.000, 0x665a10->"3",
430:                      first=(nil), last=(nil), next=(nil)})
431: Entering state 6
432: Reducing stack by rule 16 (line 58):
433:    $1 = token NUMBER (260=NUMBER)
434:          (0x6659b0-> astree {NUMBER(260), 8:4.000, 0x665a10->"3",
435:                      first=(nil), last=(nil), next=(nil)})
436: -> $$ = nterm expr ()
437: Stack now 0 2
438: Entering state 11
439: Reading a token: --accepting rule at line 46 ("*")
440: DEBUGF(f): astree.c[37] new_astree():
441: malloc (56) = 0x665a30-> 8:4.1: '*': 0x665a70->"*"
442: Next token is token '*' (42='*')
443:          (0x665a30-> astree {'*'(42), 8:4.001, 0x665a70->"*",
444:                      first=(nil), last=(nil), next=(nil)})
445: Shifting token '*' (42='*')
446:          (0x665a30-> astree {'*'(42), 8:4.001, 0x665a70->"*",
447:                      first=(nil), last=(nil), next=(nil)})
448: Entering state 19
449: Reading a token: --accepting rule at line 40 ("4")
450: DEBUGF(f): astree.c[37] new_astree():
451: malloc (56) = 0x665a90-> 8:4.2: NUMBER: 0x665ad0->"4"
452: Next token is token NUMBER (260=NUMBER)
453:          (0x665a90-> astree {NUMBER(260), 8:4.002, 0x665ad0->"4",
454:                      first=(nil), last=(nil), next=(nil)})
455: Shifting token NUMBER (260=NUMBER)
456:          (0x665a90-> astree {NUMBER(260), 8:4.002, 0x665ad0->"4",
457:                      first=(nil), last=(nil), next=(nil)})
458: Entering state 6
459: Reducing stack by rule 16 (line 58):
460:    $1 = token NUMBER (260=NUMBER)
461:          (0x665a90-> astree {NUMBER(260), 8:4.002, 0x665ad0->"4",
462:                      first=(nil), last=(nil), next=(nil)})
463: -> $$ = nterm expr ()
464: Stack now 0 2 11 19
465: Entering state 27
466: Reading a token: --accepting rule at line 51 (";")
467: DEBUGF(f): astree.c[37] new_astree():
468: malloc (56) = 0x665af0-> 8:4.3: ';': 0x665b30->";"
469: Next token is token ';' (59=';')
470:          (0x665af0-> astree {';'(59), 8:4.003, 0x665b30->";",
471:                      first=(nil), last=(nil), next=(nil)})
472: Reducing stack by rule 9 (line 51):
473:    $1 = nterm expr ()
474:    $2 = token '*' (42='*')
475:          (0x665a30-> astree {'*'(42), 8:4.001, 0x665a70->"*",
476:                      first=(nil), last=(nil), next=(nil)})
477:    $3 = nterm expr ()
478: DEBUGF(a): astree.c[55] adopt():
```

```
479: 0x665a30 (*) adopting 0x6659b0 (3)
480: DEBUGF(a): astree.c[55] adopt():
481: 0x665a30 (*) adopting 0x665a90 (4)
482: -> $$ = nterm expr ()
483: Stack now 0 2
484: Entering state 11
485: Next token is token ';' (59=';')
486:          (0x665af0-> astree {';'(59), 8:4.003, 0x665b30->";",
487:                      first=(nil), last=(nil), next=(nil)})
488: Shifting token ';' (59=';')
489:          (0x665af0-> astree {';'(59), 8:4.003, 0x665b30->";",
490:                      first=(nil), last=(nil), next=(nil)})
491: Entering state 22
492: Reducing stack by rule 2 (line 42):
493:    $1 = nterm stmtseq ()
494:    $2 = nterm expr ()
495:    $3 = token ';' (59=';')
496:          (0x665af0-> astree {';'(59), 8:4.003, 0x665b30->";",
497:                      first=(nil), last=(nil), next=(nil)})
498: DEBUGF(f): astree.c[127] freeast():
499: free [665AF0]-> 8:4.3: ';': 0x665b30->";")
500: DEBUGF(a): astree.c[55] adopt():
501: 0x6611f0 (<<ROOT>>) adopting 0x665a30 (*)
502: -> $$ = nterm stmtseq ()
503: Stack now 0
504: Entering state 2
505: Reading a token: --accepting rule at line 38 ("
506: ")
507: --(end of buffer or a NUL)
508: --EOF (start condition 0)
509: Now at end of input.
510: Reducing stack by rule 1 (line 39):
511:    $1 = nterm stmtseq ()
512: -> $$ = nterm program ()
513: Stack now 0
514: Entering state 1
515: Now at end of input.
516: Shifting token $end (0=$end)
517:          (yyvaluep = 0x665af0)
518: Entering state 3
519: Stack now 0 1 3
520: Cleanup: popping token $end (0=$end)
521:          (yyvaluep = 0x665af0)
522: Cleanup: popping nterm program ()
523: DEBUGF(a): main.c[88] main():
524:
525: <<ROOT>> 0x6611f0-> astree {ROOT(258), 0:0.000, 0x661230->"<<ROOT>>",
526:             first=0x6655a0, last=0x665a30, next=(nil)}
527:    pi 0x6655a0-> astree {IDENT(259), 5:4.000, 0x6655e0->"pi",
528:                 first=(nil), last=(nil), next=0x6656e0}
529:    = 0x6656e0-> astree {'='(61), 7:2.001, 0x665720->"=",
530:                 first=0x665600, last=0x6657a0, next=0x665920}
531:       a 0x665600-> astree {IDENT(259), 7:2.000, 0x6656c0->"a",
532:                    first=(nil), last=(nil), next=0x6657a0}
533:       * 0x6657a0-> astree {'*'(42), 7:2.004, 0x6657e0->"*",
534:                    first=0x665740, last=0x665860, next=(nil)}
535:          pi 0x665740-> astree {IDENT(259), 7:2.002, 0x665780->"pi",
536:                       first=(nil), last=(nil), next=0x665860}
537:          ^ 0x665860-> astree {'^'(94), 7:2.006, 0x6658a0->"^",
538:                       first=0x665800, last=0x6658c0, next=(nil)}
539:             r 0x665800-> astree {IDENT(259), 7:2.005, 0x665840->"r",
```

```
 540:                          first=(nil), last=(nil), next=0x6658c0}
 541:              2 0x6658c0-> astree {NUMBER(260), 7:2.007, 0x665900->"2",
 542:                          first=(nil), last=(nil), next=(nil)}
 543:    3.14159265358979323846264338328 0x665920-> astree {NUMBER(260), 7:3.000,
0x665980->"3.14159265358979323846264338328",
 544:                 first=(nil), last=(nil), next=0x665a30}
 545:    * 0x665a30-> astree {'*'(42), 8:4.001, 0x665a70->"*",
 546:                 first=0x6659b0, last=0x665a90, next=(nil)}
 547:       3 0x6659b0-> astree {NUMBER(260), 8:4.000, 0x665a10->"3",
 548:                    first=(nil), last=(nil), next=0x665a90}
 549:       4 0x665a90-> astree {NUMBER(260), 8:4.002, 0x665ad0->"4",
 550:                    first=(nil), last=(nil), next=(nil)}
 551: DEBUGF(f): astree.c[127] freeast():
 552: free [6655A0]-> 5:4.0: IDENT: 0x6655e0->"pi")
 553: DEBUGF(f): astree.c[127] freeast():
 554: free [665600]-> 7:2.0: IDENT: 0x6656c0->"a")
 555: DEBUGF(f): astree.c[127] freeast():
 556: free [665740]-> 7:2.2: IDENT: 0x665780->"pi")
 557: DEBUGF(f): astree.c[127] freeast():
 558: free [665800]-> 7:2.5: IDENT: 0x665840->"r")
 559: DEBUGF(f): astree.c[127] freeast():
 560: free [6658C0]-> 7:2.7: NUMBER: 0x665900->"2")
 561: DEBUGF(f): astree.c[127] freeast():
 562: free [665860]-> 7:2.6: '^': 0x6658a0->"^")
 563: DEBUGF(f): astree.c[127] freeast():
 564: free [6657A0]-> 7:2.4: '*': 0x6657e0->"*")
 565: DEBUGF(f): astree.c[127] freeast():
 566: free [6656E0]-> 7:2.1: '=': 0x665720->"=")
 567: DEBUGF(f): astree.c[127] freeast():
 568: free [665920]-> 7:3.0: NUMBER: 0x665980->"3.14159265358979323846264338328")
 569: DEBUGF(f): astree.c[127] freeast():
 570: free [6659B0]-> 8:4.0: NUMBER: 0x665a10->"3")
 571: DEBUGF(f): astree.c[127] freeast():
 572: free [665A90]-> 8:4.2: NUMBER: 0x665ad0->"4")
 573: DEBUGF(f): astree.c[127] freeast():
 574: free [665A30]-> 8:4.1: '*': 0x665a70->"*")
 575: DEBUGF(f): astree.c[127] freeast():
 576: free [6611F0]-> 0:0.0: ROOT: 0x661230->"<<ROOT>>")
```

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