

```
1: flex version 2.5.35 usage statistics:
2:  scanner options: -dpsvI8 -Cem -oyylex.cc
3:  106/2000 NFA states
4:  29/1000 DFA states (158 words)
5:  16 rules
6:  Compressed tables always back-up
7:  1/40 start conditions
8:  73 epsilon states, 39 double epsilon states
9:  18/100 character classes needed 197/500 words of storage, 0 reused
10:  92 state/nextstate pairs created
11:  57/35 unique/duplicate transitions
12:  33/1000 base-def entries created
13:  59/2000 (peak 77) nxt-chk entries created
14:  20/2500 (peak 68) template nxt-chk entries created
15:  0 empty table entries
16:  5 protos created
17:  4 templates created, 5 uses
18:  17/256 equivalence classes created
19:  5/256 meta-equivalence classes created
20:  0 (2 saved) hash collisions, 31 DFAs equal
21:  0 sets of reallocations needed
22:  457 total table entries needed
```

```
1: Terminals unused in grammar
2:
3:   ROOT
4:   "u+"
5:   "u-"
6:
7:
8: Grammar
9:
10:   0 $accept: program $end
11:
12:   1 program: stmtseq
13:
14:   2 stmtseq: stmtseq expr ';'
15:   3         | stmtseq error ';'
16:   4         | stmtseq ';'
17:   5         | /* empty */
18:
19:   6 expr: expr '=' expr
20:   7      | expr '+' expr
21:   8      | expr '-' expr
22:   9      | expr '*' expr
23:  10      | expr '/' expr
24:  11      | expr '^' expr
25:  12      | '+' expr
26:  13      | '-' expr
27:  14      | '(' expr ')'
28:  15      | IDENT
29:  16      | NUMBER
30:
31:
32: Terminals, with rules where they appear
33:
34: $end (0) 0
35: '(' (40) 14
36: ')' (41) 14
37: '*' (42) 9
38: '+' (43) 7 12
39: '-' (45) 8 13
40: '/' (47) 10
41: ';' (59) 2 3 4
42: '=' (61) 6
43: '^' (94) 11
44: error (256) 3
45: ROOT (258)
46: IDENT (259) 15
47: NUMBER (260) 16
48: "u+" (261)
49: "u-" (262)
50: NEG (263)
51: POS (264)
52:
53:
54: Nonterminals, with rules where they appear
55:
56: $accept (19)
57:   on left: 0
58: program (20)
```

```
59:      on left: 1, on right: 0
60: stmtseq (21)
61:      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 1.2 2013-09-19 19:55:32-07 - - $
3:   $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
4:   $Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
5:   $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
6:   $Id: astree.cc,v 1.14 2013-10-10 18:48:18-07 - - $
7:   $Compiled: astree.cc Oct 15 2013 16:37:57 $
8:
9: emit.o:
10:  $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
11:  $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
12:  $Id: emit.h,v 1.1 2013-09-19 16:38:25-07 - - $
13:  $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
14:  $Id: emit.cc,v 1.3 2013-09-20 17:52:13-07 - - $
15:  $Compiled: emit.cc Oct 15 2013 16:37:57 $
16:
17: lyutils.o:
18:  $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
19:  $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
20:  $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
21:  $Id: lyutils.cc,v 1.3 2013-10-10 18:17:45-07 - - $
22:  $Compiled: lyutils.cc Oct 15 2013 16:37:57 $
23:
24: auxlib.o:
25:  $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
26:  $Id: auxlib.cc,v 1.3 2013-09-20 17:52:13-07 - - $
27:  $Compiled: auxlib.cc Oct 15 2013 16:37:57 $
28:
29: stringset.o:
30:  $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
31:  $Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
32:  $Id: stringset.cc,v 1.6 2013-10-10 17:44:18-07 - - $
33:  $Compiled: stringset.cc Oct 15 2013 16:37:58 $
34:
35: main.o:
36:  $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
37:  $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
38:  $Id: emit.h,v 1.1 2013-09-19 16:38:25-07 - - $
39:  $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
40:  $Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
41:  $Id: main.cc,v 1.4 2013-09-20 17:52:13-07 - - $
42:  $Compiled: main.cc Oct 15 2013 16:37:58 $
43:
44: yylex.o:
45:  $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
46:  $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
47:  $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
48:  $Id: scanner.l,v 1.2 2013-09-19 19:55:32-07 - - $
49:  $Compiled: scanner.l Oct 15 2013 16:37:58 $
50:
51: yyparse.o:
52:  $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
53:  $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
54:  $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
55:  $Id: parser.y,v 1.5 2013-10-10 18:48:18-07 - - $
56:  $Compiled: parser.y Oct 15 2013 16:37:58 $
57:
58: zexprsm:
```

```
59: $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
60: $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
61: $Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
62: $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
63: $Id: astree.cc,v 1.14 2013-10-10 18:48:18-07 - - $
64: $Compiled: astree.cc Oct 15 2013 16:37:57 $
65: $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
66: $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
67: $Id: emit.h,v 1.1 2013-09-19 16:38:25-07 - - $
68: $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
69: $Id: emit.cc,v 1.3 2013-09-20 17:52:13-07 - - $
70: $Compiled: emit.cc Oct 15 2013 16:37:57 $
71: $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
72: $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
73: $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
74: $Id: lyutils.cc,v 1.3 2013-10-10 18:17:45-07 - - $
75: $Compiled: lyutils.cc Oct 15 2013 16:37:57 $
76: $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
77: $Id: auxlib.cc,v 1.3 2013-09-20 17:52:13-07 - - $
78: $Compiled: auxlib.cc Oct 15 2013 16:37:57 $
79: $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
80: $Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
81: $Id: stringset.cc,v 1.6 2013-10-10 17:44:18-07 - - $
82: $Compiled: stringset.cc Oct 15 2013 16:37:58 $
83: $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
84: $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
85: $Id: emit.h,v 1.1 2013-09-19 16:38:25-07 - - $
86: $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
87: $Id: stringset.h,v 1.5 2013-09-23 14:16:09-07 - - $
88: $Id: main.cc,v 1.4 2013-09-20 17:52:13-07 - - $
89: $Compiled: main.cc Oct 15 2013 16:37:58 $
90: $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
91: $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
92: $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
93: $Id: scanner.l,v 1.2 2013-09-19 19:55:32-07 - - $
94: $Compiled: scanner.l Oct 15 2013 16:37:58 $
95: $Id: auxlib.h,v 1.2 2013-09-19 19:55:32-07 - - $
96: $Id: astree.h,v 1.3 2013-09-20 12:23:31-07 - - $
97: $Id: lyutils.h,v 1.5 2013-10-10 18:17:45-07 - - $
98: $Id: parser.y,v 1.5 2013-10-10 18:48:18-07 - - $
99: $Compiled: parser.y Oct 15 2013 16:37:58 $
```

```
1: // $Id: test1.in,v 1.1 2013-09-19 16:38:25-07 - - $  
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: ==5540== Memcheck, a memory error detector
2: ==5540== Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward et al.
3: ==5540== Using Valgrind-3.8.1 and LibVEX; rerun with -h for copyright in
fo
4: ==5540== Command: zexprsm -ly -@@ test1.in
5: ==5540==
6: DEBUGF(x): auxlib.cc[112] set_debugflags():
7: Debugflags = "@", all = 1
8: DEBUGF(m): main.cc[69] scan_opts():
9: filename = test1.in, yyin = 0x4c2ald0, fileno (yyin) = 3
10: Starting parse
11: Entering state 0
12: Reducing stack by rule 5 (line 47):
13: DEBUGF(f): astree.cc[23] new_astree():
14: astree 0x4c2a3d0->{0:0.0: ROOT: "<<ROOT>>"}
15: -> $$ = nterm stmtseq ()
16: Stack now 0
17: Entering state 2
18: Reading a token: --(end of buffer or a NUL)
19: --accepting rule at line 32 ("# 1 "test1.in")
20: DEBUGF(m): lyutils.cc[97] scanner_include():
21: filename=test1.in, scan_linenr=0
22: --accepting rule at line 34 ("
23: ")
24: --accepting rule at line 32 ("# 1 "<built-in>")
25: DEBUGF(m): lyutils.cc[97] scanner_include():
26: filename=<built-in>, scan_linenr=0
27: --accepting rule at line 34 ("
28: ")
29: --accepting rule at line 32 ("# 1 "<command-line>")
30: DEBUGF(m): lyutils.cc[97] scanner_include():
31: filename=<command-line>, scan_linenr=0
32: --accepting rule at line 34 ("
33: ")
34: --accepting rule at line 32 ("# 1 "test1.in")
35: DEBUGF(m): lyutils.cc[97] scanner_include():
36: filename=test1.in, scan_linenr=0
37: --accepting rule at line 34 ("
38: ")
39: --accepting rule at line 34 ("
40: ")
41: --accepting rule at line 37 ("a")
42: DEBUGF(f): astree.cc[23] new_astree():
43: astree 0x4c2e920->{4:2.0: IDENT: "a"}
44: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
45: toknum = 259, yyvaluep = 0x4c2e920
46: 0x4c2e920->{IDENT(259) 4:2.000 "a" []})
47: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
48: toknum = 259, yyvaluep = 0x4c2e920
49: 0x4c2e920->{IDENT(259) 4:2.000 "a" []})
50: Entering state 5
51: Reducing stack by rule 15 (line 59):
52: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
53: toknum = 259, yyvaluep = 0x4c2e920
54: 0x4c2e920->{IDENT(259) 4:2.000 "a" []})
55: -> $$ = nterm expr ()
56: Stack now 0 2
57: Entering state 11
```

```
58: Reading a token: --accepting rule at line 39 ("=")
59: DEBUGF(f): astree.cc[23] new_astree():
60: astree 0x4c2ea50->{4:2.1: '=': "="}
61: Next token is token '=' (DEBUGF(f): astree.cc[79] yyprint():
62: toknum = 61, yyvaluep = 0x4c2ea50
63: 0x4c2ea50->{'='(61) 4:2.001 "=" []})
64: Shifting token '=' (DEBUGF(f): astree.cc[79] yyprint():
65: toknum = 61, yyvaluep = 0x4c2ea50
66: 0x4c2ea50->{'='(61) 4:2.001 "=" []})
67: Entering state 16
68: Reading a token: --accepting rule at line 37 ("b")
69: DEBUGF(f): astree.cc[23] new_astree():
70: astree 0x4c2eb80->{4:2.2: IDENT: "b"}
71: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
72: toknum = 259, yyvaluep = 0x4c2eb80
73: 0x4c2eb80->{IDENT(259) 4:2.002 "b" []})
74: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
75: toknum = 259, yyvaluep = 0x4c2eb80
76: 0x4c2eb80->{IDENT(259) 4:2.002 "b" []})
77: Entering state 5
78: Reducing stack by rule 15 (line 59):
79: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
80: toknum = 259, yyvaluep = 0x4c2eb80
81: 0x4c2eb80->{IDENT(259) 4:2.002 "b" []})
82: -> $$ = nterm expr ()
83: Stack now 0 2 11 16
84: Entering state 24
85: Reading a token: --accepting rule at line 42 ("*")
86: DEBUGF(f): astree.cc[23] new_astree():
87: astree 0x4c2ecb0->{4:2.3: '*': "*"}
88: Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
89: toknum = 42, yyvaluep = 0x4c2ecb0
90: 0x4c2ecb0->{'*'(42) 4:2.003 "*" []})
91: Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
92: toknum = 42, yyvaluep = 0x4c2ecb0
93: 0x4c2ecb0->{'*'(42) 4:2.003 "*" []})
94: Entering state 19
95: Reading a token: --accepting rule at line 37 ("c")
96: DEBUGF(f): astree.cc[23] new_astree():
97: astree 0x4c2ede0->{4:2.4: IDENT: "c"}
98: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
99: toknum = 259, yyvaluep = 0x4c2ede0
100: 0x4c2ede0->{IDENT(259) 4:2.004 "c" []})
101: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
102: toknum = 259, yyvaluep = 0x4c2ede0
103: 0x4c2ede0->{IDENT(259) 4:2.004 "c" []})
104: Entering state 5
105: Reducing stack by rule 15 (line 59):
106: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
107: toknum = 259, yyvaluep = 0x4c2ede0
108: 0x4c2ede0->{IDENT(259) 4:2.004 "c" []})
109: -> $$ = nterm expr ()
110: Stack now 0 2 11 16 24 19
111: Entering state 27
112: Reading a token: --accepting rule at line 40 ("+")
113: DEBUGF(f): astree.cc[23] new_astree():
114: astree 0x4c2ef10->{4:2.5: '+': "+"}
115: Next token is token '+' (DEBUGF(f): astree.cc[79] yyprint():
```

```
116: toknum = 43, yyvaluep = 0x4c2ef10
117: 0x4c2ef10->{'+' (43) 4:2.005 "+" []})
118: Reducing stack by rule 9 (line 53):
119:   $1 = nterm expr ()
120:   $2 = token '*' (DEBUGF(f): astree.cc[79] yyprint():
121: toknum = 42, yyvaluep = 0x4c2ecb0
122: 0x4c2ecb0->{'*' (42) 4:2.003 "*" []})
123:   $3 = nterm expr ()
124: DEBUGF(a): astree.cc[32] adopt1():
125: 0x4c2ecb0 (*) adopting 0x4c2eb80 (b)
126: DEBUGF(a): astree.cc[32] adopt1():
127: 0x4c2ecb0 (*) adopting 0x4c2ede0 (c)
128: -> $$ = nterm expr ()
129: Stack now 0 2 11 16
130: Entering state 24
131: Next token is token '+' (DEBUGF(f): astree.cc[79] yyprint():
132: toknum = 43, yyvaluep = 0x4c2ef10
133: 0x4c2ef10->{'+' (43) 4:2.005 "+" []})
134: Shifting token '+' (DEBUGF(f): astree.cc[79] yyprint():
135: toknum = 43, yyvaluep = 0x4c2ef10
136: 0x4c2ef10->{'+' (43) 4:2.005 "+" []})
137: Entering state 17
138: Reading a token: --accepting rule at line 37 ("d")
139: DEBUGF(f): astree.cc[23] new_astree():
140: astree 0x4c2f0e0->{4:2.6: IDENT: "d"}
141: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
142: toknum = 259, yyvaluep = 0x4c2f0e0
143: 0x4c2f0e0->{IDENT(259) 4:2.006 "d" []})
144: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
145: toknum = 259, yyvaluep = 0x4c2f0e0
146: 0x4c2f0e0->{IDENT(259) 4:2.006 "d" []})
147: Entering state 5
148: Reducing stack by rule 15 (line 59):
149:   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
150: toknum = 259, yyvaluep = 0x4c2f0e0
151: 0x4c2f0e0->{IDENT(259) 4:2.006 "d" []})
152: -> $$ = nterm expr ()
153: Stack now 0 2 11 16 24 17
154: Entering state 25
155: Reading a token: --accepting rule at line 42 ("*")
156: DEBUGF(f): astree.cc[23] new_astree():
157: astree 0x4c2f210->{4:2.7: '*': "*"}
158: Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
159: toknum = 42, yyvaluep = 0x4c2f210
160: 0x4c2f210->{'*' (42) 4:2.007 "*" []})
161: Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
162: toknum = 42, yyvaluep = 0x4c2f210
163: 0x4c2f210->{'*' (42) 4:2.007 "*" []})
164: Entering state 19
165: Reading a token: --accepting rule at line 37 ("e")
166: DEBUGF(f): astree.cc[23] new_astree():
167: astree 0x4c2f2f0->{4:2.8: IDENT: "e"}
168: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
169: toknum = 259, yyvaluep = 0x4c2f2f0
170: 0x4c2f2f0->{IDENT(259) 4:2.008 "e" []})
171: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
172: toknum = 259, yyvaluep = 0x4c2f2f0
173: 0x4c2f2f0->{IDENT(259) 4:2.008 "e" []})
```



```
174: Entering state 5
175: Reducing stack by rule 15 (line 59):
176:   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
177: toknum = 259, yyvaluep = 0x4c2f2f0
178: 0x4c2f2f0->{IDENT(259) 4:2.008 "e" []})
179: -> $$ = nterm expr ()
180: Stack now 0 2 11 16 24 17 25 19
181: Entering state 27
182: Reading a token: --accepting rule at line 47 (";")
183: DEBUGF(f): astree.cc[23] new_astree():
184: astree 0x4c2f420->{4:2.9: ';' : ";" }
185: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
186: toknum = 59, yyvaluep = 0x4c2f420
187: 0x4c2f420->{';' (59) 4:2.009 ";" []})
188: Reducing stack by rule 9 (line 53):
189:   $1 = nterm expr ()
190:   $2 = token '*' (DEBUGF(f): astree.cc[79] yyprint():
191: toknum = 42, yyvaluep = 0x4c2f210
192: 0x4c2f210->{'*' (42) 4:2.007 "*" []})
193:   $3 = nterm expr ()
194: DEBUGF(a): astree.cc[32] adopt1():
195: 0x4c2f210 (*) adopting 0x4c2f0e0 (d)
196: DEBUGF(a): astree.cc[32] adopt1():
197: 0x4c2f210 (*) adopting 0x4c2f2f0 (e)
198: -> $$ = nterm expr ()
199: Stack now 0 2 11 16 24 17
200: Entering state 25
201: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
202: toknum = 59, yyvaluep = 0x4c2f420
203: 0x4c2f420->{';' (59) 4:2.009 ";" []})
204: Reducing stack by rule 7 (line 51):
205:   $1 = nterm expr ()
206:   $2 = token '+' (DEBUGF(f): astree.cc[79] yyprint():
207: toknum = 43, yyvaluep = 0x4c2ef10
208: 0x4c2ef10->{'+' (43) 4:2.005 "+" []})
209:   $3 = nterm expr ()
210: DEBUGF(a): astree.cc[32] adopt1():
211: 0x4c2ef10 (+) adopting 0x4c2ecb0 (*)
212: DEBUGF(a): astree.cc[32] adopt1():
213: 0x4c2ef10 (+) adopting 0x4c2f210 (*)
214: -> $$ = nterm expr ()
215: Stack now 0 2 11 16
216: Entering state 24
217: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
218: toknum = 59, yyvaluep = 0x4c2f420
219: 0x4c2f420->{';' (59) 4:2.009 ";" []})
220: Reducing stack by rule 6 (line 50):
221:   $1 = nterm expr ()
222:   $2 = token '=' (DEBUGF(f): astree.cc[79] yyprint():
223: toknum = 61, yyvaluep = 0x4c2ea50
224: 0x4c2ea50->{'=' (61) 4:2.001 "=" []})
225:   $3 = nterm expr ()
226: DEBUGF(a): astree.cc[32] adopt1():
227: 0x4c2ea50 (=) adopting 0x4c2e920 (a)
228: DEBUGF(a): astree.cc[32] adopt1():
229: 0x4c2ea50 (=) adopting 0x4c2ef10 (+)
230: -> $$ = nterm expr ()
231: Stack now 0 2
```

```
232: Entering state 11
233: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
234: toknum = 59, yyvaluep = 0x4c2f420
235: 0x4c2f420->{';' (59) 4:2.009 ";" []})
236: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
237: toknum = 59, yyvaluep = 0x4c2f420
238: 0x4c2f420->{';' (59) 4:2.009 ";" []})
239: Entering state 22
240: Reducing stack by rule 2 (line 44):
241:   $1 = nterm stmtseq ()
242:   $2 = nterm expr ()
243:   $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
244: toknum = 59, yyvaluep = 0x4c2f420
245: 0x4c2f420->{';' (59) 4:2.009 ";" []})
246: DEBUGF(f): astree.cc[97] free_ast():
247: free [4C2F420]-> 4:2.9: ';' : ";"
248: DEBUGF(a): astree.cc[32] adopt1():
249: 0x4c2a3d0 (<<ROOT>>) adopting 0x4c2ea50 (=)
250: -> $$ = nterm stmtseq ()
251: Stack now 0
252: Entering state 2
253: Reading a token: --accepting rule at line 34 ("
254: ")
255: --(end of buffer or a NUL)
256: --EOF (start condition 0)
257: Now at end of input.
258: Reducing stack by rule 1 (line 41):
259:   $1 = nterm stmtseq ()
260: -> $$ = nterm program ()
261: Stack now 0
262: Entering state 1
263: Now at end of input.
264: Shifting token $end (DEBUGF(f): astree.cc[79] yyprint():
265: toknum = 0, yyvaluep = 0x4c2f420
266: $end(0)
267: )
268: Entering state 3
269: Stack now 0 1 3
270: Cleanup: popping token $end (DEBUGF(f): astree.cc[79] yyprint():
271: toknum = 0, yyvaluep = 0x4c2f420
272: $end(0)
273: )
274: Cleanup: popping nterm program ()
275: DEBUGF(a): main.cc[87] main():
276:
277: <<ROOT>> 0x4c2a3d0->{ROOT(258) 0:0.000 "<<ROOT>>" [0x4c2ea50]}
278:   = 0x4c2ea50->{'=' (61) 4:2.001 "=" [0x4c2e920 0x4c2ef10]}
279:     a 0x4c2e920->{IDENT(259) 4:2.000 "a" []}
280:     + 0x4c2ef10->{'+' (43) 4:2.005 "+" [0x4c2ecb0 0x4c2f210]}
281:       * 0x4c2ecb0->{'*' (42) 4:2.003 "*" [0x4c2eb80 0x4c2ede0]}
282:         b 0x4c2eb80->{IDENT(259) 4:2.002 "b" []}
283:         c 0x4c2ede0->{IDENT(259) 4:2.004 "c" []}
284:       * 0x4c2f210->{'*' (42) 4:2.007 "*" [0x4c2f0e0 0x4c2f2f0]}
285:         d 0x4c2f0e0->{IDENT(259) 4:2.006 "d" []}
286:         e 0x4c2f2f0->{IDENT(259) 4:2.008 "e" []}
287: DEBUGF(f): astree.cc[97] free_ast():
288: free [4C2F2F0]-> 4:2.8: IDENT: "e")
289: DEBUGF(f): astree.cc[97] free_ast():
```

```
290: free [4C2F0E0]-> 4:2.6: IDENT: "d")
291: DEBUGF(f): astree.cc[97] free_ast():
292: free [4C2F210]-> 4:2.7: '*' : "*" )
293: DEBUGF(f): astree.cc[97] free_ast():
294: free [4C2EDE0]-> 4:2.4: IDENT: "c")
295: DEBUGF(f): astree.cc[97] free_ast():
296: free [4C2EB80]-> 4:2.2: IDENT: "b")
297: DEBUGF(f): astree.cc[97] free_ast():
298: free [4C2ECB0]-> 4:2.3: '*' : "*" )
299: DEBUGF(f): astree.cc[97] free_ast():
300: free [4C2EF10]-> 4:2.5: '+' : "+" )
301: DEBUGF(f): astree.cc[97] free_ast():
302: free [4C2E920]-> 4:2.0: IDENT: "a")
303: DEBUGF(f): astree.cc[97] free_ast():
304: free [4C2EA50]-> 4:2.1: '=' : "=" )
305: DEBUGF(f): astree.cc[97] free_ast():
306: free [4C2A3D0]-> 0:0.0: ROOT: "<<ROOT>>")
307: DEBUGF(s): main.cc[92] main():
308:
309: stringset[ 1]: 12638182802509129152 0x4c2f3d0->"e"
310: stringset[ 3]: 12638127826927718602 0x4c2eff0-> "+"
311: stringset[ 5]: 12638145419113769978 0x4c2f500->";"
312: stringset[ 6]: 12638190499090526629 0x4c2ec60->"b"
313: stringset[ 7]: 12638138822044000712 0x4c2eb30-> "="
314: 12638187200555641996 0x4c2ea00->"a"
315: stringset[ 8]: 12638183902020757363 0x4c2f1c0->"d"
316: 15588046758478310861 0x4c2a4c0->"<<ROOT>>"
317: stringset[ 10]: 12638189399578898418 0x4c2eec0->"c"
318: 12638128926439346813 0x4c2ed90->"*"
319: load_factor = 0.909
320: bucket_count = 11
321: max_bucket_size = 2
322: ==5540==
323: ==5540== HEAP SUMMARY:
324: ==5540==      in use at exit: 0 bytes in 0 blocks
325: ==5540==    total heap usage: 57 allocs, 57 frees, 18,450 bytes allocated
326: ==5540==
327: ==5540== All heap blocks were freed -- no leaks are possible
328: ==5540==
329: ==5540== For counts of detected and suppressed errors, rerun with: -v
330: ==5540== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
331: EXIT STATUS 0
```

```
1:  t// $Id: test4.in,v 1.1 2013-09-19 16:38:25-07 - - $  
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=3.141592653589793238462643383280;
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: 3.141592653589793238462643383280;
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    3.141592653589793238462643383280; 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: ==5545== Memcheck, a memory error detector
2: ==5545== Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward et al.
3: ==5545== Using Valgrind-3.8.1 and LibVEX; rerun with -h for copyright in
fo
4: ==5545== Command: zexprsm -ly -@@ test4.in
5: ==5545==
6: DEBUGF(x): auxlib.cc[112] set_debugflags():
7: Debugflags = "@", all = 1
8: DEBUGF(m): main.cc[69] scan_opts():
9: filename = test4.in, yyin = 0x4c2ald0, fileno (yyin) = 3
10: Starting parse
11: Entering state 0
12: Reducing stack by rule 5 (line 47):
13: DEBUGF(f): astree.cc[23] new_astree():
14: astree 0x4c2a3d0->{0:0.0: ROOT: "<<ROOT>>"}
15: -> $$ = nterm stmtseq ()
16: Stack now 0
17: Entering state 2
18: Reading a token: --(end of buffer or a NUL)
19: --accepting rule at line 32 ("# 1 "test4.in")
20: DEBUGF(m): lyutils.cc[97] scanner_include():
21: filename=test4.in, scan_linenr=0
22: --accepting rule at line 34 ("
23: ")
24: --accepting rule at line 32 ("# 1 "<built-in>")
25: DEBUGF(m): lyutils.cc[97] scanner_include():
26: filename=<built-in>, scan_linenr=0
27: --accepting rule at line 34 ("
28: ")
29: --accepting rule at line 32 ("# 1 "<command-line>")
30: DEBUGF(m): lyutils.cc[97] scanner_include():
31: filename=<command-line>, scan_linenr=0
32: --accepting rule at line 34 ("
33: ")
34: --accepting rule at line 32 ("# 1 "test4.in")
35: DEBUGF(m): lyutils.cc[97] scanner_include():
36: filename=test4.in, scan_linenr=0
37: --accepting rule at line 34 ("
38: ")
39: --accepting rule at line 33 (" ")
40: --accepting rule at line 37 ("t")
41: DEBUGF(f): astree.cc[23] new_astree():
42: astree 0x4c2e920->{4:1.1: IDENT: "t"}
43: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint()):
44: toknum = 259, yyvaluep = 0x4c2e920
45: 0x4c2e920->{IDENT(259) 4:1.001 "t" []})
46: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint()):
47: toknum = 259, yyvaluep = 0x4c2e920
48: 0x4c2e920->{IDENT(259) 4:1.001 "t" []})
49: Entering state 5
50: Reducing stack by rule 15 (line 59):
51: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint()):
52: toknum = 259, yyvaluep = 0x4c2e920
53: 0x4c2e920->{IDENT(259) 4:1.001 "t" []})
54: -> $$ = nterm expr ()
55: Stack now 0 2
56: Entering state 11
57: Reading a token: --accepting rule at line 34 ("
```

```
58: ")
59: --accepting rule at line 32 ("# 1 "test4a.inh" 1")
60: DEBUGF(m): lyutils.cc[97] scanner_include():
61: filename=test4a.inh, scan_linenr=0
62: --accepting rule at line 34 ("
63: ")
64: --accepting rule at line 34 ("
65: ")
66: --accepting rule at line 34 ("
67: ")
68: --accepting rule at line 37 ("pi")
69: DEBUGF(f): astree.cc[23] new_astree():
70: astree 0x4c2eac0->{5:3.0: IDENT: "pi"}
71: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
72: toknum = 259, yyvaluep = 0x4c2eac0
73: 0x4c2eac0->{IDENT(259) 5:3.000 "pi" []})
74: zexprsm: test4a.inh: 3: syntax error, unexpected IDENT
75: Error: popping nterm expr ()
76: DEBUGF(a): lyutils.cc[74] error_destructor():
77:
78: t 0x4c2e920->{IDENT(259) 4:1.001 "t" []}
79: DEBUGF(f): astree.cc[97] free_ast():
80: free [4C2E920]-> 4:1.1: IDENT: "t")
81: Stack now 0 2
82: Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
83: toknum = 256, yyvaluep = 0x4c2eac0
84: error(256)
85: )
86: Entering state 4
87: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
88: toknum = 259, yyvaluep = 0x4c2eac0
89: 0x4c2eac0->{IDENT(259) 5:3.000 "pi" []})
90: Error: discarding token IDENT (DEBUGF(f): astree.cc[79] yyprint():
91: toknum = 259, yyvaluep = 0x4c2eac0
92: 0x4c2eac0->{IDENT(259) 5:3.000 "pi" []})
93: DEBUGF(a): lyutils.cc[74] error_destructor():
94:
95: pi 0x4c2eac0->{IDENT(259) 5:3.000 "pi" []}
96: DEBUGF(f): astree.cc[97] free_ast():
97: free [4C2EAC0]-> 5:3.0: IDENT: "pi")
98: Error: popping token error (DEBUGF(f): astree.cc[79] yyprint():
99: toknum = 256, yyvaluep = 0x4c2eac0
100: error(256)
101: )
102: Stack now 0 2
103: Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
104: toknum = 256, yyvaluep = 0x4c2eac0
105: error(256)
106: )
107: Entering state 4
108: Reading a token: --accepting rule at line 39 ("=")
109: DEBUGF(f): astree.cc[23] new_astree():
110: astree 0x4c2ebf0->{5:3.2: '=': "="}
111: Next token is token '=' (DEBUGF(f): astree.cc[79] yyprint():
112: toknum = 61, yyvaluep = 0x4c2ebf0
113: 0x4c2ebf0->{'=' (61) 5:3.002 "=" []})
114: Error: discarding token '=' (DEBUGF(f): astree.cc[79] yyprint():
115: toknum = 61, yyvaluep = 0x4c2ebf0
```

```
116: 0x4c2ebf0->{'=' (61) 5:3.002 "=" []})
117: DEBUGF(a): lyutils.cc[74] error_destructor():
118:
119: = 0x4c2ebf0->{'=' (61) 5:3.002 "=" []}
120: DEBUGF(f): astree.cc[97] free_ast():
121: free [4C2EBF0]-> 5:3.2: '=': "=")
122: Error: popping token error (DEBUGF(f): astree.cc[79] yyprint():
123: toknum = 256, yyvaluep = 0x4c2eac0
124: error(256)
125: )
126: Stack now 0 2
127: Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
128: toknum = 256, yyvaluep = 0x4c2ebf0
129: error(256)
130: )
131: Entering state 4
132: Reading a token: --accepting rule at line 36 ("3.14159265358979323846264
3383280")
133: DEBUGF(f): astree.cc[23] new_astree():
134: astree 0x4c2ed20->{5:3.3: NUMBER: "3.141592653589793238462643383280"}
135: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
136: toknum = 260, yyvaluep = 0x4c2ed20
137: 0x4c2ed20->{NUMBER(260) 5:3.003 "3.141592653589793238462643383280" []})
138: Error: discarding token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
139: toknum = 260, yyvaluep = 0x4c2ed20
140: 0x4c2ed20->{NUMBER(260) 5:3.003 "3.141592653589793238462643383280" []})
141: DEBUGF(a): lyutils.cc[74] error_destructor():
142:
143: 3.141592653589793238462643383280 0x4c2ed20->{NUMBER(260) 5:3.003 "3.1415
92653589793238462643383280" []}
144: DEBUGF(f): astree.cc[97] free_ast():
145: free [4C2ED20]-> 5:3.3: NUMBER: "3.141592653589793238462643383280")
146: Error: popping token error (DEBUGF(f): astree.cc[79] yyprint():
147: toknum = 256, yyvaluep = 0x4c2ebf0
148: error(256)
149: )
150: Stack now 0 2
151: Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
152: toknum = 256, yyvaluep = 0x4c2ed20
153: error(256)
154: )
155: Entering state 4
156: Reading a token: --accepting rule at line 47 (";")
157: DEBUGF(f): astree.cc[23] new_astree():
158: astree 0x4c2ee70->{5:3.35: ';' : ";" }
159: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
160: toknum = 59, yyvaluep = 0x4c2ee70
161: 0x4c2ee70->{';' (59) 5:3.035 ";" []})
162: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
163: toknum = 59, yyvaluep = 0x4c2ee70
164: 0x4c2ee70->{';' (59) 5:3.035 ";" []})
165: Entering state 12
166: Reducing stack by rule 3 (line 45):
167: $1 = nterm stmtseq ()
168: $2 = token error (DEBUGF(f): astree.cc[79] yyprint():
169: toknum = 256, yyvaluep = 0x4c2ed20
170: error(256)
171: )
```



```
172:    $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
173: toknum = 59, yyvaluep = 0x4c2ee70
174: 0x4c2ee70->{';' (59) 5:3.035 ";" []})
175: DEBUGF(f): astree.cc[97] free_ast():
176: free [4C2EE70]-> 5:3.35: ';' : ";"
177: -> $$ = nterm stmtseq ()
178: Stack now 0
179: Entering state 2
180: Reading a token: --accepting rule at line 34 ("
181: ")
182: --accepting rule at line 37 ("pi")
183: DEBUGF(f): astree.cc[23] new_astree():
184: astree 0x4c2efa0->{5:4.0: IDENT: "pi"}
185: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
186: toknum = 259, yyvaluep = 0x4c2efa0
187: 0x4c2efa0->{IDENT(259) 5:4.000 "pi" []})
188: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
189: toknum = 259, yyvaluep = 0x4c2efa0
190: 0x4c2efa0->{IDENT(259) 5:4.000 "pi" []})
191: Entering state 5
192: Reducing stack by rule 15 (line 59):
193:    $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
194: toknum = 259, yyvaluep = 0x4c2efa0
195: 0x4c2efa0->{IDENT(259) 5:4.000 "pi" []})
196: -> $$ = nterm expr ()
197: Stack now 0 2
198: Entering state 11
199: Reading a token: --accepting rule at line 47 (";"")
200: DEBUGF(f): astree.cc[23] new_astree():
201: astree 0x4c2f080->{5:4.2: ';' : ";" }
202: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
203: toknum = 59, yyvaluep = 0x4c2f080
204: 0x4c2f080->{';' (59) 5:4.002 ";" []})
205: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
206: toknum = 59, yyvaluep = 0x4c2f080
207: 0x4c2f080->{';' (59) 5:4.002 ";" []})
208: Entering state 22
209: Reducing stack by rule 2 (line 44):
210:    $1 = nterm stmtseq ()
211:    $2 = nterm expr ()
212:    $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
213: toknum = 59, yyvaluep = 0x4c2f080
214: 0x4c2f080->{';' (59) 5:4.002 ";" []})
215: DEBUGF(f): astree.cc[97] free_ast():
216: free [4C2F080]-> 5:4.2: ';' : ";"
217: DEBUGF(a): astree.cc[32] adopt1():
218: 0x4c2a3d0 (<<ROOT>>) adopting 0x4c2efa0 (pi)
219: -> $$ = nterm stmtseq ()
220: Stack now 0
221: Entering state 2
222: Reading a token: --accepting rule at line 34 ("
223: ")
224: --accepting rule at line 32 ("# 3 "test4.in" 2")
225: DEBUGF(m): lyutils.cc[97] scanner_include():
226: filename=test4.in, scan_linenr=2
227: --accepting rule at line 34 ("
228: ")
229: --accepting rule at line 32 ("# 1 "test4b.inh" 1")
```

```
230: DEBUGF(m): lyutils.cc[97] scanner_include():
231: filename=test4b.inh, scan_linenr=0
232: --accepting rule at line 34 ("
233: ")
234: --accepting rule at line 34 ("
235: ")
236: --accepting rule at line 37 ("a")
237: DEBUGF(f): astree.cc[23] new_astree():
238: astree 0x4c2f290->{7:2.0: IDENT: "a"}
239: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
240: toknum = 259, yyvaluep = 0x4c2f290
241: 0x4c2f290->{IDENT(259) 7:2.000 "a" []})
242: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
243: toknum = 259, yyvaluep = 0x4c2f290
244: 0x4c2f290->{IDENT(259) 7:2.000 "a" []})
245: Entering state 5
246: Reducing stack by rule 15 (line 59):
247: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
248: toknum = 259, yyvaluep = 0x4c2f290
249: 0x4c2f290->{IDENT(259) 7:2.000 "a" []})
250: -> $$ = nterm expr ()
251: Stack now 0 2
252: Entering state 11
253: Reading a token: --accepting rule at line 39 ("=")
254: DEBUGF(f): astree.cc[23] new_astree():
255: astree 0x4c2f3c0->{7:2.1: '=': "="}
256: Next token is token '=' (DEBUGF(f): astree.cc[79] yyprint():
257: toknum = 61, yyvaluep = 0x4c2f3c0
258: 0x4c2f3c0->{'='(61) 7:2.001 "=" []})
259: Shifting token '=' (DEBUGF(f): astree.cc[79] yyprint():
260: toknum = 61, yyvaluep = 0x4c2f3c0
261: 0x4c2f3c0->{'='(61) 7:2.001 "=" []})
262: Entering state 16
263: Reading a token: --accepting rule at line 37 ("pi")
264: DEBUGF(f): astree.cc[23] new_astree():
265: astree 0x4c2f4a0->{7:2.2: IDENT: "pi"}
266: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
267: toknum = 259, yyvaluep = 0x4c2f4a0
268: 0x4c2f4a0->{IDENT(259) 7:2.002 "pi" []})
269: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
270: toknum = 259, yyvaluep = 0x4c2f4a0
271: 0x4c2f4a0->{IDENT(259) 7:2.002 "pi" []})
272: Entering state 5
273: Reducing stack by rule 15 (line 59):
274: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
275: toknum = 259, yyvaluep = 0x4c2f4a0
276: 0x4c2f4a0->{IDENT(259) 7:2.002 "pi" []})
277: -> $$ = nterm expr ()
278: Stack now 0 2 11 16
279: Entering state 24
280: Reading a token: --accepting rule at line 42 ("*")
281: DEBUGF(f): astree.cc[23] new_astree():
282: astree 0x4c2f580->{7:2.4: '*' : "*"}
283: Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
284: toknum = 42, yyvaluep = 0x4c2f580
285: 0x4c2f580->{'*' (42) 7:2.004 "*" []})
286: Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
287: toknum = 42, yyvaluep = 0x4c2f580
```

```
288: 0x4c2f580->{'*' (42) 7:2.004 "*" []})
289: Entering state 19
290: Reading a token: --accepting rule at line 37 ("r")
291: DEBUGF(f): astree.cc[23] new_astree():
292: astree 0x4c2f6b0->{7:2.5: IDENT: "r"}
293: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
294: toknum = 259, yyvaluep = 0x4c2f6b0
295: 0x4c2f6b0->{IDENT(259) 7:2.005 "r" []})
296: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
297: toknum = 259, yyvaluep = 0x4c2f6b0
298: 0x4c2f6b0->{IDENT(259) 7:2.005 "r" []})
299: Entering state 5
300: Reducing stack by rule 15 (line 59):
301: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
302: toknum = 259, yyvaluep = 0x4c2f6b0
303: 0x4c2f6b0->{IDENT(259) 7:2.005 "r" []})
304: -> $$ = nterm expr ()
305: Stack now 0 2 11 16 24 19
306: Entering state 27
307: Reading a token: --accepting rule at line 44 ("^")
308: DEBUGF(f): astree.cc[23] new_astree():
309: astree 0x4c2f7e0->{7:2.6: '^': "^"}
310: Next token is token '^' (DEBUGF(f): astree.cc[79] yyprint():
311: toknum = 94, yyvaluep = 0x4c2f7e0
312: 0x4c2f7e0->{'^' (94) 7:2.006 "^" []})
313: Shifting token '^' (DEBUGF(f): astree.cc[79] yyprint():
314: toknum = 94, yyvaluep = 0x4c2f7e0
315: 0x4c2f7e0->{'^' (94) 7:2.006 "^" []})
316: Entering state 21
317: Reading a token: --accepting rule at line 36 ("2")
318: DEBUGF(f): astree.cc[23] new_astree():
319: astree 0x4c2f910->{7:2.7: NUMBER: "2"}
320: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
321: toknum = 260, yyvaluep = 0x4c2f910
322: 0x4c2f910->{NUMBER(260) 7:2.007 "2" []})
323: Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
324: toknum = 260, yyvaluep = 0x4c2f910
325: 0x4c2f910->{NUMBER(260) 7:2.007 "2" []})
326: Entering state 6
327: Reducing stack by rule 16 (line 60):
328: $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
329: toknum = 260, yyvaluep = 0x4c2f910
330: 0x4c2f910->{NUMBER(260) 7:2.007 "2" []})
331: -> $$ = nterm expr ()
332: Stack now 0 2 11 16 24 19 27 21
333: Entering state 29
334: Reading a token: --accepting rule at line 47 (";")
335: DEBUGF(f): astree.cc[23] new_astree():
336: astree 0x4c2fa40->{7:2.8: ';'': ";"}
337: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
338: toknum = 59, yyvaluep = 0x4c2fa40
339: 0x4c2fa40->{';' (59) 7:2.008 ";" []})
340: Reducing stack by rule 11 (line 55):
341: $1 = nterm expr ()
342: $2 = token '^' (DEBUGF(f): astree.cc[79] yyprint():
343: toknum = 94, yyvaluep = 0x4c2f7e0
344: 0x4c2f7e0->{'^' (94) 7:2.006 "^" []})
345: $3 = nterm expr ()
```

```
346: DEBUGF(a): astree.cc[32] adopt1():
347: 0x4c2f7e0 (^) adopting 0x4c2f6b0 (r)
348: DEBUGF(a): astree.cc[32] adopt1():
349: 0x4c2f7e0 (^) adopting 0x4c2f910 (2)
350: -> $$ = nterm expr ()
351: Stack now 0 2 11 16 24 19
352: Entering state 27
353: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
354: toknum = 59, yyvaluep = 0x4c2fa40
355: 0x4c2fa40->{';' (59) 7:2.008 ";" []})
356: Reducing stack by rule 9 (line 53):
357:   $1 = nterm expr ()
358:   $2 = token '*' (DEBUGF(f): astree.cc[79] yyprint():
359: toknum = 42, yyvaluep = 0x4c2f580
360: 0x4c2f580->{'*' (42) 7:2.004 "*" []})
361:   $3 = nterm expr ()
362: DEBUGF(a): astree.cc[32] adopt1():
363: 0x4c2f580 (*) adopting 0x4c2f4a0 (pi)
364: DEBUGF(a): astree.cc[32] adopt1():
365: 0x4c2f580 (*) adopting 0x4c2f7e0 (^)
366: -> $$ = nterm expr ()
367: Stack now 0 2 11 16
368: Entering state 24
369: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
370: toknum = 59, yyvaluep = 0x4c2fa40
371: 0x4c2fa40->{';' (59) 7:2.008 ";" []})
372: Reducing stack by rule 6 (line 50):
373:   $1 = nterm expr ()
374:   $2 = token '=' (DEBUGF(f): astree.cc[79] yyprint():
375: toknum = 61, yyvaluep = 0x4c2f3c0
376: 0x4c2f3c0->{'=' (61) 7:2.001 "=" []})
377:   $3 = nterm expr ()
378: DEBUGF(a): astree.cc[32] adopt1():
379: 0x4c2f3c0 (=) adopting 0x4c2f290 (a)
380: DEBUGF(a): astree.cc[32] adopt1():
381: 0x4c2f3c0 (=) adopting 0x4c2f580 (*)
382: -> $$ = nterm expr ()
383: Stack now 0 2
384: Entering state 11
385: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
386: toknum = 59, yyvaluep = 0x4c2fa40
387: 0x4c2fa40->{';' (59) 7:2.008 ";" []})
388: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
389: toknum = 59, yyvaluep = 0x4c2fa40
390: 0x4c2fa40->{';' (59) 7:2.008 ";" []})
391: Entering state 22
392: Reducing stack by rule 2 (line 44):
393:   $1 = nterm stmtseq ()
394:   $2 = nterm expr ()
395:   $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
396: toknum = 59, yyvaluep = 0x4c2fa40
397: 0x4c2fa40->{';' (59) 7:2.008 ";" []})
398: DEBUGF(f): astree.cc[97] free_ast():
399: free [4C2FA40]-> 7:2.8: ';' : ";"
400: DEBUGF(a): astree.cc[32] adopt1():
401: 0x4c2a3d0 (<<ROOT>>) adopting 0x4c2f3c0 (=)
402: -> $$ = nterm stmtseq ()
403: Stack now 0
```

```
404: Entering state 2
405: Reading a token: --accepting rule at line 34 ("
406: ")
407: --accepting rule at line 36 ("3.141592653589793238462643383280")
408: DEBUGF(f): astree.cc[23] new_astree():
409: astree 0x4c2fd50->{7:3.0: NUMBER: "3.141592653589793238462643383280"}
410: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
411: toknum = 260, yyvaluep = 0x4c2fd50
412: 0x4c2fd50->{NUMBER(260) 7:3.000 "3.141592653589793238462643383280" []})
413: Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
414: toknum = 260, yyvaluep = 0x4c2fd50
415: 0x4c2fd50->{NUMBER(260) 7:3.000 "3.141592653589793238462643383280" []})
416: Entering state 6
417: Reducing stack by rule 16 (line 60):
418: $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
419: toknum = 260, yyvaluep = 0x4c2fd50
420: 0x4c2fd50->{NUMBER(260) 7:3.000 "3.141592653589793238462643383280" []})
421: -> $$ = nterm expr ()
422: Stack now 0 2
423: Entering state 11
424: Reading a token: --accepting rule at line 47 (";")
425: DEBUGF(f): astree.cc[23] new_astree():
426: astree 0x4c2fe50->{7:3.32: ';' : ";" }
427: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
428: toknum = 59, yyvaluep = 0x4c2fe50
429: 0x4c2fe50->{';' (59) 7:3.032 ";" []})
430: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
431: toknum = 59, yyvaluep = 0x4c2fe50
432: 0x4c2fe50->{';' (59) 7:3.032 ";" []})
433: Entering state 22
434: Reducing stack by rule 2 (line 44):
435: $1 = nterm stmtseq ()
436: $2 = nterm expr ()
437: $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
438: toknum = 59, yyvaluep = 0x4c2fe50
439: 0x4c2fe50->{';' (59) 7:3.032 ";" []})
440: DEBUGF(f): astree.cc[97] free_ast():
441: free [4C2FE50]-> 7:3.32: ';' : ";" )
442: DEBUGF(a): astree.cc[32] adopt1():
443: 0x4c2a3d0 (<<ROOT>>) adopting 0x4c2fd50 (3.14159265358979323846264338328
0)
444: -> $$ = nterm stmtseq ()
445: Stack now 0
446: Entering state 2
447: Reading a token: --accepting rule at line 34 ("
448: ")
449: --accepting rule at line 32 ("# 4 "test4.in" 2")
450: DEBUGF(m): lyutils.cc[97] scanner_include():
451: filename=test4.in, scan_linenr=3
452: --accepting rule at line 34 ("
453: ")
454: --accepting rule at line 36 ("3")
455: DEBUGF(f): astree.cc[23] new_astree():
456: astree 0x4c300c0->{8:4.0: NUMBER: "3"}
457: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
458: toknum = 260, yyvaluep = 0x4c300c0
459: 0x4c300c0->{NUMBER(260) 8:4.000 "3" []})
460: Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
```

```
461: toknum = 260, yyvaluep = 0x4c300c0
462: 0x4c300c0->{NUMBER(260) 8:4.000 "3" []})
463: Entering state 6
464: Reducing stack by rule 16 (line 60):
465:     $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
466: toknum = 260, yyvaluep = 0x4c300c0
467: 0x4c300c0->{NUMBER(260) 8:4.000 "3" []})
468: -> $$ = nterm expr ()
469: Stack now 0 2
470: Entering state 11
471: Reading a token: --accepting rule at line 42 ("*")
472: DEBUGF(f): astree.cc[23] new_astree():
473: astree 0x4c302f0->{8:4.1: '*' : ""}
474: Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
475: toknum = 42, yyvaluep = 0x4c302f0
476: 0x4c302f0->{'*' (42) 8:4.001 "" []})
477: Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
478: toknum = 42, yyvaluep = 0x4c302f0
479: 0x4c302f0->{'*' (42) 8:4.001 "" []})
480: Entering state 19
481: Reading a token: --accepting rule at line 36 ("4")
482: DEBUGF(f): astree.cc[23] new_astree():
483: astree 0x4c303d0->{8:4.2: NUMBER: "4"}
484: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
485: toknum = 260, yyvaluep = 0x4c303d0
486: 0x4c303d0->{NUMBER(260) 8:4.002 "4" []})
487: Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
488: toknum = 260, yyvaluep = 0x4c303d0
489: 0x4c303d0->{NUMBER(260) 8:4.002 "4" []})
490: Entering state 6
491: Reducing stack by rule 16 (line 60):
492:     $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
493: toknum = 260, yyvaluep = 0x4c303d0
494: 0x4c303d0->{NUMBER(260) 8:4.002 "4" []})
495: -> $$ = nterm expr ()
496: Stack now 0 2 11 19
497: Entering state 27
498: Reading a token: --accepting rule at line 47 (";")
499: DEBUGF(f): astree.cc[23] new_astree():
500: astree 0x4c30500->{8:4.3: ';' : ";" }
501: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
502: toknum = 59, yyvaluep = 0x4c30500
503: 0x4c30500->{';' (59) 8:4.003 ";" []})
504: Reducing stack by rule 9 (line 53):
505:     $1 = nterm expr ()
506:     $2 = token '*' (DEBUGF(f): astree.cc[79] yyprint():
507: toknum = 42, yyvaluep = 0x4c302f0
508: 0x4c302f0->{'*' (42) 8:4.001 "" []})
509:     $3 = nterm expr ()
510: DEBUGF(a): astree.cc[32] adopt1():
511: 0x4c302f0 (*) adopting 0x4c300c0 (3)
512: DEBUGF(a): astree.cc[32] adopt1():
513: 0x4c302f0 (*) adopting 0x4c303d0 (4)
514: -> $$ = nterm expr ()
515: Stack now 0 2
516: Entering state 11
517: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
518: toknum = 59, yyvaluep = 0x4c30500
```

```
519: 0x4c30500->{';' (59) 8:4.003 ";" []})
520: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
521: toknum = 59, yyvaluep = 0x4c30500
522: 0x4c30500->{';' (59) 8:4.003 ";" []})
523: Entering state 22
524: Reducing stack by rule 2 (line 44):
525:   $1 = nterm stmtseq ()
526:   $2 = nterm expr ()
527:   $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
528: toknum = 59, yyvaluep = 0x4c30500
529: 0x4c30500->{';' (59) 8:4.003 ";" []})
530: DEBUGF(f): astree.cc[97] free_ast():
531: free [4C30500]-> 8:4.3: ';' : ";"
532: DEBUGF(a): astree.cc[32] adopt1():
533: 0x4c2a3d0 (<<ROOT>>) adopting 0x4c302f0 (*)
534: -> $$ = nterm stmtseq ()
535: Stack now 0
536: Entering state 2
537: Reading a token: --accepting rule at line 34 ("
538: ")
539: --(end of buffer or a NUL)
540: --EOF (start condition 0)
541: Now at end of input.
542: Reducing stack by rule 1 (line 41):
543:   $1 = nterm stmtseq ()
544: -> $$ = nterm program ()
545: Stack now 0
546: Entering state 1
547: Now at end of input.
548: Shifting token $end (DEBUGF(f): astree.cc[79] yyprint():
549: toknum = 0, yyvaluep = 0x4c30500
550: $end(0)
551: )
552: Entering state 3
553: Stack now 0 1 3
554: Cleanup: popping token $end (DEBUGF(f): astree.cc[79] yyprint():
555: toknum = 0, yyvaluep = 0x4c30500
556: $end(0)
557: )
558: Cleanup: popping nterm program ()
559: DEBUGF(a): main.cc[87] main():
560:
561: <<ROOT>> 0x4c2a3d0->{ROOT(258) 0:0.000 "<<ROOT>>" [0x4c2efa0 0x4c2f3c0 0
x4c2fd50 0x4c302f0]}
562:   pi 0x4c2efa0->{IDENT(259) 5:4.000 "pi" []}
563:   = 0x4c2f3c0->{'=' (61) 7:2.001 "=" [0x4c2f290 0x4c2f580]}
564:   a 0x4c2f290->{IDENT(259) 7:2.000 "a" []}
565:   * 0x4c2f580->{'*' (42) 7:2.004 "*" [0x4c2f4a0 0x4c2f7e0]}
566:     pi 0x4c2f4a0->{IDENT(259) 7:2.002 "pi" []}
567:     ^ 0x4c2f7e0->{'^' (94) 7:2.006 "^" [0x4c2f6b0 0x4c2f910]}
568:     r 0x4c2f6b0->{IDENT(259) 7:2.005 "r" []}
569:     2 0x4c2f910->{NUMBER(260) 7:2.007 "2" []}
570:     3.141592653589793238462643383280 0x4c2fd50->{NUMBER(260) 7:3.000 "3.1
41592653589793238462643383280" []}
571:     * 0x4c302f0->{'*' (42) 8:4.001 "*" [0x4c300c0 0x4c303d0]}
572:     3 0x4c300c0->{NUMBER(260) 8:4.000 "3" []}
573:     4 0x4c303d0->{NUMBER(260) 8:4.002 "4" []}
574: DEBUGF(f): astree.cc[97] free_ast():
```

```
575: free [4C303D0]-> 8:4.2: NUMBER: "4")
576: DEBUGF(f): astree.cc[97] free_ast():
577: free [4C300C0]-> 8:4.0: NUMBER: "3")
578: DEBUGF(f): astree.cc[97] free_ast():
579: free [4C302F0]-> 8:4.1: '*' : "*"")
580: DEBUGF(f): astree.cc[97] free_ast():
581: free [4C2FD50]-> 7:3.0: NUMBER: "3.141592653589793238462643383280")
582: DEBUGF(f): astree.cc[97] free_ast():
583: free [4C2F910]-> 7:2.7: NUMBER: "2")
584: DEBUGF(f): astree.cc[97] free_ast():
585: free [4C2F6B0]-> 7:2.5: IDENT: "r")
586: DEBUGF(f): astree.cc[97] free_ast():
587: free [4C2F7E0]-> 7:2.6: '^' : "^")
588: DEBUGF(f): astree.cc[97] free_ast():
589: free [4C2F4A0]-> 7:2.2: IDENT: "pi")
590: DEBUGF(f): astree.cc[97] free_ast():
591: free [4C2F580]-> 7:2.4: '*' : "*"")
592: DEBUGF(f): astree.cc[97] free_ast():
593: free [4C2F290]-> 7:2.0: IDENT: "a")
594: DEBUGF(f): astree.cc[97] free_ast():
595: free [4C2F3C0]-> 7:2.1: '=' : "=")
596: DEBUGF(f): astree.cc[97] free_ast():
597: free [4C2EFA0]-> 5:4.0: IDENT: "pi")
598: DEBUGF(f): astree.cc[97] free_ast():
599: free [4C2A3D0]-> 0:0.0: ROOT: "<<ROOT>>")
600: DEBUGF(s): main.cc[92] main():
601:
602: stringset[ 3]: 12638137722532372501 0x4c2f9f0->"2"
603: stringset[ 5]: 15588046758478310861 0x4c2a4c0->"<<ROOT>>"
604: stringset[ 6]: 12638131125462603235 0x4c304b0->"4"
605: stringset[ 7]: 12638128926439346813 0x4c2f660->"*"
606: stringset[11]: 12638145419113769978 0x4c2ef50->" ; "
607: stringset[14]: 12638138822044000712 0x4c2ecd0->"="
608: stringset[15]: 12638136623020744290 0x4c301a0->"3"
609: 9410160442050401285 0x4c2ee20->"3.141592653589793238
462643383280"
610: 6364844224684479794 0x4c2eba0->"pi"
611: 12638187200555641996 0x4c2f370->"a"
612: stringset[17]: 12638208091276578005 0x4c2f790->"r"
613: stringset[20]: 12638201494206808739 0x4c2ea00->"t"
614: stringset[22]: 12638247673695193601 0x4c2f8c0->"^"
615: load_factor = 0.565
616: bucket_count = 23
617: max_bucket_size = 4
618: ==5545==
619: ==5545== HEAP SUMMARY:
620: ==5545== in use at exit: 0 bytes in 0 blocks
621: ==5545== total heap usage: 90 allocs, 90 frees, 20,057 bytes allocated
622: ==5545==
623: ==5545== All heap blocks were freed -- no leaks are possible
624: ==5545==
625: ==5545== For counts of detected and suppressed errors, rerun with: -v
626: ==5545== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
627: EXIT STATUS 1
```