

```
1: flex version 2.5.35 usage statistics:
2:  scanner options: -dsvI8 -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:
5:
6: Grammar
7:
8:   0 $accept: program $end
9:
10:  1 program: stmtseq
11:
12:  2 stmtseq: stmtseq expr ';'
13:  3         | stmtseq error ';'
14:  4         | stmtseq ';'
15:  5         | /* empty */
16:
17:  6 expr: expr '=' expr
18:  7      | expr '+' expr
19:  8      | expr '-' expr
20:  9      | expr '*' expr
21: 10      | expr '/' expr
22: 11      | expr '^' expr
23: 12      | '+' expr
24: 13      | '-' expr
25: 14      | '(' expr ')'
26: 15      | IDENT
27: 16      | NUMBER
28:
29:
30: Terminals, with rules where they appear
31:
32: $end (0) 0
33: '(' (40) 14
34: ')' (41) 14
35: '*' (42) 9
36: '+' (43) 7 12
37: '-' (45) 8 13
38: '/' (47) 10
39: ';' (59) 2 3 4
40: '=' (61) 6
41: '^' (94) 11
42: error (256) 3
43: ROOT (258)
44: IDENT (259) 15
45: NUMBER (260) 16
46: NEG (261)
47: POS (262)
48:
49:
50: Nonterminals, with rules where they appear
51:
52: $accept (17)
53:   on left: 0
54: program (18)
55:   on left: 1, on right: 0
56: stmtseq (19)
57:   on left: 2 3 4 5, on right: 1 2 3 4
58: expr (20)
```

```
59:      on left: 6 7 8 9 10 11 12 13 14 15 16, on right: 2 6 7 8 9 10 11
60:      12 13 14
61:
62:
63: state 0
64:
65:      0 $accept: . program $end
66:
67:      $default  reduce using rule 5 (stmtseq)
68:
69:      program go to state 1
70:      stmtseq go to state 2
71:
72:
73: state 1
74:
75:      0 $accept: program . $end
76:
77:      $end  shift, and go to state 3
78:
79:
80: state 2
81:
82:      1 program: stmtseq .
83:      2 stmtseq: stmtseq . expr ';'
84:      3          | stmtseq . error ';'
85:      4          | stmtseq . ';'
86:
87:      error  shift, and go to state 4
88:      IDENT  shift, and go to state 5
89:      NUMBER shift, and go to state 6
90:      '+'    shift, and go to state 7
91:      '-'    shift, and go to state 8
92:      ';'    shift, and go to state 9
93:      '('    shift, and go to state 10
94:
95:      $end  reduce using rule 1 (program)
96:
97:      expr go to state 11
98:
99:
100: state 3
101:
102:      0 $accept: program $end .
103:
104:      $default  accept
105:
106:
107: state 4
108:
109:      3 stmtseq: stmtseq error . ';'
110:
111:      ';'  shift, and go to state 12
112:
113:
114: state 5
115:
116:      15 expr: IDENT .
```

```
117:
118:     $default  reduce using rule 15 (expr)
119:
120:
121: state 6
122:
123:     16 expr: NUMBER .
124:
125:     $default  reduce using rule 16 (expr)
126:
127:
128: state 7
129:
130:     12 expr: '+' . expr
131:
132:     IDENT      shift, and go to state 5
133:     NUMBER     shift, and go to state 6
134:     '+'        shift, and go to state 7
135:     '-'        shift, and go to state 8
136:     '('        shift, and go to state 10
137:
138:     expr go to state 13
139:
140:
141: state 8
142:
143:     13 expr: '-' . expr
144:
145:     IDENT      shift, and go to state 5
146:     NUMBER     shift, and go to state 6
147:     '+'        shift, and go to state 7
148:     '-'        shift, and go to state 8
149:     '('        shift, and go to state 10
150:
151:     expr go to state 14
152:
153:
154: state 9
155:
156:     4 stmtseq: stmtseq ';' .
157:
158:     $default  reduce using rule 4 (stmtseq)
159:
160:
161: state 10
162:
163:     14 expr: '(' . expr ')'
164:
165:     IDENT      shift, and go to state 5
166:     NUMBER     shift, and go to state 6
167:     '+'        shift, and go to state 7
168:     '-'        shift, and go to state 8
169:     '('        shift, and go to state 10
170:
171:     expr go to state 15
172:
173:
174: state 11
```

```
175:
176:      2 stmtseq: stmtseq expr . ';'
177:      6 expr: expr . '=' expr
178:      7      | expr . '+' expr
179:      8      | expr . '-' expr
180:      9      | expr . '*' expr
181:     10      | expr . '/' expr
182:     11      | expr . '^' expr
183:
184:      '=' shift, and go to state 16
185:      '+' shift, and go to state 17
186:      '-' shift, and go to state 18
187:      '*' shift, and go to state 19
188:      '/' shift, and go to state 20
189:      '^' shift, and go to state 21
190:      ';' shift, and go to state 22
191:
192:
193: state 12
194:
195:      3 stmtseq: stmtseq error ';' .
196:
197:      $default reduce using rule 3 (stmtseq)
198:
199:
200: state 13
201:
202:      6 expr: expr . '=' expr
203:      7      | expr . '+' expr
204:      8      | expr . '-' expr
205:      9      | expr . '*' expr
206:     10      | expr . '/' expr
207:     11      | expr . '^' expr
208:     12      | '+' expr .
209:
210:      $default reduce using rule 12 (expr)
211:
212:
213: state 14
214:
215:      6 expr: expr . '=' expr
216:      7      | expr . '+' expr
217:      8      | expr . '-' expr
218:      9      | expr . '*' expr
219:     10      | expr . '/' expr
220:     11      | expr . '^' expr
221:     13      | '-' expr .
222:
223:      $default reduce using rule 13 (expr)
224:
225:
226: state 15
227:
228:      6 expr: expr . '=' expr
229:      7      | expr . '+' expr
230:      8      | expr . '-' expr
231:      9      | expr . '*' expr
232:     10      | expr . '/' expr
```

```
233: 11      | expr . '^' expr
234: 14      | '(' expr . ')'
235:
236:      '=' shift, and go to state 16
237:      '+' shift, and go to state 17
238:      '-' shift, and go to state 18
239:      '*' shift, and go to state 19
240:      '/' shift, and go to state 20
241:      '^' shift, and go to state 21
242:      ')' shift, and go to state 23
243:
244:
245: state 16
246:
247:      6 expr: expr '=' . expr
248:
249:      IDENT shift, and go to state 5
250:      NUMBER shift, and go to state 6
251:      '+' shift, and go to state 7
252:      '-' shift, and go to state 8
253:      '(' shift, and go to state 10
254:
255:      expr go to state 24
256:
257:
258: state 17
259:
260:      7 expr: expr '+' . expr
261:
262:      IDENT shift, and go to state 5
263:      NUMBER shift, and go to state 6
264:      '+' shift, and go to state 7
265:      '-' shift, and go to state 8
266:      '(' shift, and go to state 10
267:
268:      expr go to state 25
269:
270:
271: state 18
272:
273:      8 expr: expr '-' . expr
274:
275:      IDENT shift, and go to state 5
276:      NUMBER shift, and go to state 6
277:      '+' shift, and go to state 7
278:      '-' shift, and go to state 8
279:      '(' shift, and go to state 10
280:
281:      expr go to state 26
282:
283:
284: state 19
285:
286:      9 expr: expr '*' . expr
287:
288:      IDENT shift, and go to state 5
289:      NUMBER shift, and go to state 6
290:      '+' shift, and go to state 7
```

```
291:      '-'      shift, and go to state 8
292:      '('      shift, and go to state 10
293:
294:      expr go to state 27
295:
296:
297: state 20
298:
299:      10 expr: expr '/' . expr
300:
301:      IDENT      shift, and go to state 5
302:      NUMBER      shift, and go to state 6
303:      '+'      shift, and go to state 7
304:      '-'      shift, and go to state 8
305:      '('      shift, and go to state 10
306:
307:      expr go to state 28
308:
309:
310: state 21
311:
312:      11 expr: expr '^' . expr
313:
314:      IDENT      shift, and go to state 5
315:      NUMBER      shift, and go to state 6
316:      '+'      shift, and go to state 7
317:      '-'      shift, and go to state 8
318:      '('      shift, and go to state 10
319:
320:      expr go to state 29
321:
322:
323: state 22
324:
325:      2 stmtseq: stmtseq expr ';' .
326:
327:      $default  reduce using rule 2 (stmtseq)
328:
329:
330: state 23
331:
332:      14 expr: '(' expr ')' .
333:
334:      $default  reduce using rule 14 (expr)
335:
336:
337: state 24
338:
339:      6 expr: expr . '=' expr
340:      6      | expr '=' expr .
341:      7      | expr . '+' expr
342:      8      | expr . '-' expr
343:      9      | expr . '*' expr
344:      10     | expr . '/' expr
345:      11     | expr . '^' expr
346:
347:      '='      shift, and go to state 16
348:      '+'      shift, and go to state 17
```

```
349:      '-'  shift, and go to state 18
350:      '*'  shift, and go to state 19
351:      '/'  shift, and go to state 20
352:      '^'  shift, and go to state 21
353:
354:      $default  reduce using rule 6 (expr)
355:
356:
357: state 25
358:
359:      6 expr: expr . '=' expr
360:      7      | expr . '+' expr
361:      7      | expr '+' expr .
362:      8      | expr . '-' expr
363:      9      | expr . '*' expr
364:     10      | expr . '/' expr
365:     11      | expr . '^' expr
366:
367:      '*'  shift, and go to state 19
368:      '/'  shift, and go to state 20
369:      '^'  shift, and go to state 21
370:
371:      $default  reduce using rule 7 (expr)
372:
373:
374: state 26
375:
376:      6 expr: expr . '=' expr
377:      7      | expr . '+' expr
378:      8      | expr . '-' expr
379:      8      | expr '-' expr .
380:      9      | expr . '*' expr
381:     10      | expr . '/' expr
382:     11      | expr . '^' expr
383:
384:      '*'  shift, and go to state 19
385:      '/'  shift, and go to state 20
386:      '^'  shift, and go to state 21
387:
388:      $default  reduce using rule 8 (expr)
389:
390:
391: state 27
392:
393:      6 expr: expr . '=' expr
394:      7      | expr . '+' expr
395:      8      | expr . '-' expr
396:      9      | expr . '*' expr
397:      9      | expr '*' expr .
398:     10      | expr . '/' expr
399:     11      | expr . '^' expr
400:
401:      '^'  shift, and go to state 21
402:
403:      $default  reduce using rule 9 (expr)
404:
405:
406: state 28
```



```
407:
408:      6 expr: expr . '=' expr
409:      7      | expr . '+' expr
410:      8      | expr . '-' expr
411:      9      | expr . '*' expr
412:     10      | expr . '/' expr
413:     10      | expr '/' expr .
414:     11      | expr . '^' expr
415:
416:      '^' shift, and go to state 21
417:
418:      $default reduce using rule 10 (expr)
419:
420:
421: state 29
422:
423:      6 expr: expr . '=' expr
424:      7      | expr . '+' expr
425:      8      | expr . '-' expr
426:      9      | expr . '*' expr
427:     10      | expr . '/' expr
428:     11      | expr . '^' expr
429:     11      | expr '^' expr .
430:
431:      '^' shift, and go to state 21
432:
433:      $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 1.6 2015-04-16 17:19:58-07 - - $
6:   $Id: astree.cc,v 1.14 2013-10-10 18:48:18-07 - - $
7:   $Compiled: astree.cc Apr 16 2015 17:21:13 $
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 1.6 2015-04-16 17:19:58-07 - - $
14:  $Id: emit.cc,v 1.3 2013-09-20 17:52:13-07 - - $
15:  $Compiled: emit.cc Apr 16 2015 17:21:13 $
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 1.6 2015-04-16 17:19:58-07 - - $
21:  $Id: lyutils.cc,v 1.3 2013-10-10 18:17:45-07 - - $
22:  $Compiled: lyutils.cc Apr 16 2015 17:21:13 $
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 Apr 16 2015 17:21:14 $
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 1.7 2015-04-16 17:19:22-07 - - $
33:  $Compiled: stringset.cc Apr 16 2015 17:21:14 $
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 1.6 2015-04-16 17:19:58-07 - - $
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 Apr 16 2015 17:21:14 $
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 1.6 2015-04-16 17:19:58-07 - - $
48:  $Id: scanner.l,v 1.5 2015-04-16 17:17:51-07 - - $
49:  $Compiled: scanner.l Apr 16 2015 17:21:14 $
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 1.6 2015-04-16 17:19:58-07 - - $
55:  $Id: parser.y,v 1.6 2015-04-16 17:17:22-07 - - $
56:  $Compiled: parser.y Apr 16 2015 17:21:15 $
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 1.6 2015-04-16 17:19:58-07 - - $
63: $Id: astree.cc,v 1.14 2013-10-10 18:48:18-07 - - $
64: $Compiled: astree.cc Apr 16 2015 17:21:13 $
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 1.6 2015-04-16 17:19:58-07 - - $
69: $Id: emit.cc,v 1.3 2013-09-20 17:52:13-07 - - $
70: $Compiled: emit.cc Apr 16 2015 17:21:13 $
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 1.6 2015-04-16 17:19:58-07 - - $
74: $Id: lyutils.cc,v 1.3 2013-10-10 18:17:45-07 - - $
75: $Compiled: lyutils.cc Apr 16 2015 17:21:13 $
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 Apr 16 2015 17:21:14 $
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 1.7 2015-04-16 17:19:22-07 - - $
82: $Compiled: stringset.cc Apr 16 2015 17:21:14 $
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 1.6 2015-04-16 17:19:58-07 - - $
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 Apr 16 2015 17:21:14 $
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 1.6 2015-04-16 17:19:58-07 - - $
93: $Id: scanner.l,v 1.5 2015-04-16 17:17:51-07 - - $
94: $Compiled: scanner.l Apr 16 2015 17:21:14 $
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 1.6 2015-04-16 17:19:58-07 - - $
98: $Id: parser.y,v 1.6 2015-04-16 17:17:22-07 - - $
99: $Compiled: parser.y Apr 16 2015 17:21:15 $
```

```
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: ==8933== Memcheck, a memory error detector
2: ==8933== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
3: ==8933== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright in
fo
4: ==8933== Command: zexprsm -ly -@@ test1.in
5: ==8933==
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 = 0x4c2e1d0, 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 0x4c2e3d0->{0:0.0: ROOT: "<<ROOT>>"}
15: -> $$ = nterm stmtseq ()
16: Stack now 0
17: Entering state 2
18: Reading a token: --(end of buffer or a NUL)
19: --accepting rule at line 29 ("# 1 "test1.in")
20: DEBUGF(m): lyutils.cc[97] scanner_include():
21: filename=test1.in, scan_linenr=0
22: --accepting rule at line 31 ("
23: ")
24: --accepting rule at line 29 ("# 1 "<built-in>")
25: DEBUGF(m): lyutils.cc[97] scanner_include():
26: filename=<built-in>, scan_linenr=0
27: --accepting rule at line 31 ("
28: ")
29: --accepting rule at line 29 ("# 1 "<command-line>")
30: DEBUGF(m): lyutils.cc[97] scanner_include():
31: filename=<command-line>, scan_linenr=0
32: --accepting rule at line 31 ("
33: ")
34: --accepting rule at line 29 ("# 1 "test1.in")
35: DEBUGF(m): lyutils.cc[97] scanner_include():
36: filename=test1.in, scan_linenr=0
37: --accepting rule at line 31 ("
38: ")
39: --accepting rule at line 31 ("
40: ")
41: --accepting rule at line 34 ("a")
42: DEBUGF(f): astree.cc[23] new_astree():
43: astree 0x4c32930->{4:2.0: IDENT: "a"}
44: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
45: toknum = 259, yyvaluep = 0x4c32930
46: 0x4c32930->{IDENT(259) 4:2.000 "a" []})
47: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
48: toknum = 259, yyvaluep = 0x4c32930
49: 0x4c32930->{IDENT(259) 4:2.000 "a" []})
50: Entering state 5
51: Reducing stack by rule 15 (line 59):
52: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
53: toknum = 259, yyvaluep = 0x4c32930
54: 0x4c32930->{IDENT(259) 4:2.000 "a" []})
55: -> $$ = nterm expr ()
56: Stack now 0 2
57: Entering state 11
```

```
58: Reading a token: --accepting rule at line 36 ("=")
59: DEBUGF(f): astree.cc[23] new_astree():
60: astree 0x4c32a70->{4:2.1: '=': "="}
61: Next token is token '=' (DEBUGF(f): astree.cc[79] yyprint():
62: toknum = 61, yyvaluep = 0x4c32a70
63: 0x4c32a70->{'=' (61) 4:2.001 "=" []})
64: Shifting token '=' (DEBUGF(f): astree.cc[79] yyprint():
65: toknum = 61, yyvaluep = 0x4c32a70
66: 0x4c32a70->{'=' (61) 4:2.001 "=" []})
67: Entering state 16
68: Reading a token: --accepting rule at line 34 ("b")
69: DEBUGF(f): astree.cc[23] new_astree():
70: astree 0x4c32bb0->{4:2.2: IDENT: "b"}
71: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
72: toknum = 259, yyvaluep = 0x4c32bb0
73: 0x4c32bb0->{IDENT(259) 4:2.002 "b" []})
74: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
75: toknum = 259, yyvaluep = 0x4c32bb0
76: 0x4c32bb0->{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 = 0x4c32bb0
81: 0x4c32bb0->{IDENT(259) 4:2.002 "b" []})
82: -> $$ = nterm expr ()
83: Stack now 0 2 11 16
84: Entering state 24
85: Reading a token: --accepting rule at line 39 ("*")
86: DEBUGF(f): astree.cc[23] new_astree():
87: astree 0x4c32cf0->{4:2.3: '*': "*"}
88: Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
89: toknum = 42, yyvaluep = 0x4c32cf0
90: 0x4c32cf0->{'*' (42) 4:2.003 "*" []})
91: Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
92: toknum = 42, yyvaluep = 0x4c32cf0
93: 0x4c32cf0->{'*' (42) 4:2.003 "*" []})
94: Entering state 19
95: Reading a token: --accepting rule at line 34 ("c")
96: DEBUGF(f): astree.cc[23] new_astree():
97: astree 0x4c32e30->{4:2.4: IDENT: "c"}
98: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
99: toknum = 259, yyvaluep = 0x4c32e30
100: 0x4c32e30->{IDENT(259) 4:2.004 "c" []})
101: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
102: toknum = 259, yyvaluep = 0x4c32e30
103: 0x4c32e30->{IDENT(259) 4:2.004 "c" []})
104: Entering state 5
105: Reducing stack by rule 15 (line 59):
106: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
107: toknum = 259, yyvaluep = 0x4c32e30
108: 0x4c32e30->{IDENT(259) 4:2.004 "c" []})
109: -> $$ = nterm expr ()
110: Stack now 0 2 11 16 24 19
111: Entering state 27
112: Reading a token: --accepting rule at line 37 ("+")
113: DEBUGF(f): astree.cc[23] new_astree():
114: astree 0x4c32f70->{4:2.5: '+': "+"}
115: Next token is token '+' (DEBUGF(f): astree.cc[79] yyprint():
```

```
116: toknum = 43, yyvaluep = 0x4c32f70
117: 0x4c32f70->{'+' (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 = 0x4c32cf0
122: 0x4c32cf0->{'*' (42) 4:2.003 "*" []})
123:   $3 = nterm expr ()
124: DEBUGF(a): astree.cc[32] adopt1():
125: 0x4c32cf0 (*) adopting 0x4c32bb0 (b)
126: DEBUGF(a): astree.cc[32] adopt1():
127: 0x4c32cf0 (*) adopting 0x4c32e30 (c)
128: -> $$ = nterm expr ()
129: Stack now 0 2 11 16
130: Entering state 24
131: Next token is token '+' (DEBUGF(f): astree.cc[79] yyprint():
132: toknum = 43, yyvaluep = 0x4c32f70
133: 0x4c32f70->{'+' (43) 4:2.005 "+" []})
134: Shifting token '+' (DEBUGF(f): astree.cc[79] yyprint():
135: toknum = 43, yyvaluep = 0x4c32f70
136: 0x4c32f70->{'+' (43) 4:2.005 "+" []})
137: Entering state 17
138: Reading a token: --accepting rule at line 34 ("d")
139: DEBUGF(f): astree.cc[23] new_astree():
140: astree 0x4c33150->{4:2.6: IDENT: "d"}
141: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
142: toknum = 259, yyvaluep = 0x4c33150
143: 0x4c33150->{IDENT(259) 4:2.006 "d" []})
144: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
145: toknum = 259, yyvaluep = 0x4c33150
146: 0x4c33150->{IDENT(259) 4:2.006 "d" []})
147: Entering state 5
148: Reducing stack by rule 15 (line 59):
149:   $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
150: toknum = 259, yyvaluep = 0x4c33150
151: 0x4c33150->{IDENT(259) 4:2.006 "d" []})
152: -> $$ = nterm expr ()
153: Stack now 0 2 11 16 24 17
154: Entering state 25
155: Reading a token: --accepting rule at line 39 ("*")
156: DEBUGF(f): astree.cc[23] new_astree():
157: astree 0x4c33290->{4:2.7: '*': "*" }
158: Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
159: toknum = 42, yyvaluep = 0x4c33290
160: 0x4c33290->{'*' (42) 4:2.007 "*" []})
161: Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
162: toknum = 42, yyvaluep = 0x4c33290
163: 0x4c33290->{'*' (42) 4:2.007 "*" []})
164: Entering state 19
165: Reading a token: --accepting rule at line 34 ("e")
166: DEBUGF(f): astree.cc[23] new_astree():
167: astree 0x4c33370->{4:2.8: IDENT: "e"}
168: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
169: toknum = 259, yyvaluep = 0x4c33370
170: 0x4c33370->{IDENT(259) 4:2.008 "e" []})
171: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
172: toknum = 259, yyvaluep = 0x4c33370
173: 0x4c33370->{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 = 0x4c33370
178: 0x4c33370->{IDENT(259) 4:2.008 "e" []})
179: -> $$ = nterm expr ()
180: Stack now 0 2 11 16 24 17 25 19
181: Entering state 27
182: Reading a token: --accepting rule at line 44 (";")
183: DEBUGF(f): astree.cc[23] new_astree():
184: astree 0x4c334b0->{4:2.9: ';' : ";" }
185: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
186: toknum = 59, yyvaluep = 0x4c334b0
187: 0x4c334b0->{';' (59) 4:2.009 ";" []})
188: Reducing stack by rule 9 (line 53):
189:   $1 = nterm expr ()
190:   $2 = token '*' (DEBUGF(f): astree.cc[79] yyprint():
191: toknum = 42, yyvaluep = 0x4c33290
192: 0x4c33290->{'*' (42) 4:2.007 "*" []})
193:   $3 = nterm expr ()
194: DEBUGF(a): astree.cc[32] adopt1():
195: 0x4c33290 (*) adopting 0x4c33150 (d)
196: DEBUGF(a): astree.cc[32] adopt1():
197: 0x4c33290 (*) adopting 0x4c33370 (e)
198: -> $$ = 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 = 0x4c334b0
203: 0x4c334b0->{';' (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 = 0x4c32f70
208: 0x4c32f70->{'+' (43) 4:2.005 "+" []})
209:   $3 = nterm expr ()
210: DEBUGF(a): astree.cc[32] adopt1():
211: 0x4c32f70 (+) adopting 0x4c32cf0 (*)
212: DEBUGF(a): astree.cc[32] adopt1():
213: 0x4c32f70 (+) adopting 0x4c33290 (*)
214: -> $$ = nterm expr ()
215: Stack now 0 2 11 16
216: Entering state 24
217: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
218: toknum = 59, yyvaluep = 0x4c334b0
219: 0x4c334b0->{';' (59) 4:2.009 ";" []})
220: Reducing stack by rule 6 (line 50):
221:   $1 = nterm expr ()
222:   $2 = token '=' (DEBUGF(f): astree.cc[79] yyprint():
223: toknum = 61, yyvaluep = 0x4c32a70
224: 0x4c32a70->{'=' (61) 4:2.001 "=" []})
225:   $3 = nterm expr ()
226: DEBUGF(a): astree.cc[32] adopt1():
227: 0x4c32a70 (=) adopting 0x4c32930 (a)
228: DEBUGF(a): astree.cc[32] adopt1():
229: 0x4c32a70 (=) adopting 0x4c32f70 (+)
230: -> $$ = 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 = 0x4c334b0
235: 0x4c334b0->{';' (59) 4:2.009 ";" []})
236: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
237: toknum = 59, yyvaluep = 0x4c334b0
238: 0x4c334b0->{';' (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 = 0x4c334b0
245: 0x4c334b0->{';' (59) 4:2.009 ";" []})
246: DEBUGF(f): astree.cc[97] free_ast():
247: free [4C334B0]-> 4:2.9: ';' : ";"
248: DEBUGF(a): astree.cc[32] adopt1():
249: 0x4c2e3d0 (<<ROOT>>) adopting 0x4c32a70 (=)
250: -> $$ = nterm stmtseq ()
251: Stack now 0
252: Entering state 2
253: Reading a token: --accepting rule at line 31 ("
254: ")
255: --(end of buffer or a NUL)
256: --EOF (start condition 0)
257: Now at end of input.
258: Reducing stack by rule 1 (line 41):
259:   $1 = nterm stmtseq ()
260: -> $$ = 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 = 0x4c334b0
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 = 0x4c334b0
272: $end(0)
273: )
274: Cleanup: popping nterm program ()
275: DEBUGF(a): main.cc[87] main():
276:
277: <<ROOT>> 0x4c2e3d0->{ROOT(258) 0:0.000 "<<ROOT>>" [0x4c32a70]}
278:   = 0x4c32a70->{'=' (61) 4:2.001 "=" [0x4c32930 0x4c32f70]}
279:     a 0x4c32930->{IDENT(259) 4:2.000 "a" []}
280:     + 0x4c32f70->{'+' (43) 4:2.005 "+" [0x4c32cf0 0x4c33290]}
281:       * 0x4c32cf0->{'*' (42) 4:2.003 "*" [0x4c32bb0 0x4c32e30]}
282:         b 0x4c32bb0->{IDENT(259) 4:2.002 "b" []}
283:         c 0x4c32e30->{IDENT(259) 4:2.004 "c" []}
284:       * 0x4c33290->{'*' (42) 4:2.007 "*" [0x4c33150 0x4c33370]}
285:         d 0x4c33150->{IDENT(259) 4:2.006 "d" []}
286:         e 0x4c33370->{IDENT(259) 4:2.008 "e" []}
287: DEBUGF(f): astree.cc[97] free_ast():
288: free [4C33370]-> 4:2.8: IDENT: "e")
289: DEBUGF(f): astree.cc[97] free_ast():
```

```
290: free [4C33150]-> 4:2.6: IDENT: "d")
291: DEBUGF(f): astree.cc[97] free_ast():
292: free [4C33290]-> 4:2.7: '*' : "*" )
293: DEBUGF(f): astree.cc[97] free_ast():
294: free [4C32E30]-> 4:2.4: IDENT: "c")
295: DEBUGF(f): astree.cc[97] free_ast():
296: free [4C32BB0]-> 4:2.2: IDENT: "b")
297: DEBUGF(f): astree.cc[97] free_ast():
298: free [4C32CF0]-> 4:2.3: '*' : "*" )
299: DEBUGF(f): astree.cc[97] free_ast():
300: free [4C32F70]-> 4:2.5: '+' : "+" )
301: DEBUGF(f): astree.cc[97] free_ast():
302: free [4C32930]-> 4:2.0: IDENT: "a")
303: DEBUGF(f): astree.cc[97] free_ast():
304: free [4C32A70]-> 4:2.1: '=' : "=" )
305: DEBUGF(f): astree.cc[97] free_ast():
306: free [4C2E3D0]-> 0:0.0: ROOT: "<<ROOT>>")
307: DEBUGF(s): main.cc[92] main():
308:
309: stringset[ 0]: 10959529184379665549 0x4c32f18->"c"
310:                3729804957429652673 0x4c2e4c8->"<<ROOT>>"
311: stringset[ 1]: 2540012008095083820 0x4c33458->"e"
312:                10838281452030117757 0x4c32c98->"b"
313:                11597697714117577063 0x4c32b58-> "="
314: stringset[ 3]: 1370730858159036685 0x4c33058-> "+"
315:                6637313742931709005 0x4c32dd8-> "*"
316: stringset[ 4]: 5344662657899890615 0x4c33598-> "; "
317: stringset[ 5]: 14494284460613645429 0x4c33238-> "d"
318: stringset[ 7]: 4993892634952068459 0x4c32a18-> "a"
319: load_factor = 0.909
320: bucket_count = 11
321: max_bucket_size = 3
322: ==8933==
323: ==8933== HEAP SUMMARY:
324: ==8933==      in use at exit: 0 bytes in 0 blocks
325: ==8933== total heap usage: 57 allocs, 57 frees, 18,522 bytes allocated
326: ==8933==
327: ==8933== All heap blocks were freed -- no leaks are possible
328: ==8933==
329: ==8933== For counts of detected and suppressed errors, rerun with: -v
330: ==8933== 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: ==8984== Memcheck, a memory error detector
2: ==8984== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
3: ==8984== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright in
fo
4: ==8984== Command: zexprsm -ly -@@ test4.in
5: ==8984==
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 = 0x4c2e1d0, 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 0x4c2e3d0->{0:0.0: ROOT: "<<ROOT>>"}
15: -> $$ = nterm stmtseq ()
16: Stack now 0
17: Entering state 2
18: Reading a token: --(end of buffer or a NUL)
19: --accepting rule at line 29 ("# 1 "test4.in")
20: DEBUGF(m): lyutils.cc[97] scanner_include():
21: filename=test4.in, scan_linenr=0
22: --accepting rule at line 31 ("
23: ")
24: --accepting rule at line 29 ("# 1 "<built-in>")
25: DEBUGF(m): lyutils.cc[97] scanner_include():
26: filename=<built-in>, scan_linenr=0
27: --accepting rule at line 31 ("
28: ")
29: --accepting rule at line 29 ("# 1 "<command-line>")
30: DEBUGF(m): lyutils.cc[97] scanner_include():
31: filename=<command-line>, scan_linenr=0
32: --accepting rule at line 31 ("
33: ")
34: --accepting rule at line 29 ("# 1 "test4.in")
35: DEBUGF(m): lyutils.cc[97] scanner_include():
36: filename=test4.in, scan_linenr=0
37: --accepting rule at line 31 ("
38: ")
39: --accepting rule at line 30 (" ")
40: --accepting rule at line 34 ("t")
41: DEBUGF(f): astree.cc[23] new_astree():
42: astree 0x4c32930->{4:1.1: IDENT: "t"}
43: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
44: toknum = 259, yyvaluep = 0x4c32930
45: 0x4c32930->{IDENT(259) 4:1.001 "t" []})
46: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
47: toknum = 259, yyvaluep = 0x4c32930
48: 0x4c32930->{IDENT(259) 4:1.001 "t" []})
49: Entering state 5
50: Reducing stack by rule 15 (line 59):
51: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
52: toknum = 259, yyvaluep = 0x4c32930
53: 0x4c32930->{IDENT(259) 4:1.001 "t" []})
54: -> $$ = nterm expr ()
55: Stack now 0 2
56: Entering state 11
57: Reading a token: --accepting rule at line 31 ("
```

```
58: ")
59: --accepting rule at line 29 ("# 1 "test4a.inh" 1")
60: DEBUGF(m): lyutils.cc[97] scanner_include():
61: filename=test4a.inh, scan_linenr=0
62: --accepting rule at line 31 ("
63: ")
64: --accepting rule at line 31 ("
65: ")
66: --accepting rule at line 31 ("
67: ")
68: --accepting rule at line 34 ("pi")
69: DEBUGF(f): astree.cc[23] new_astree():
70: astree 0x4c32ae0->{5:3.0: IDENT: "pi"}
71: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
72: toknum = 259, yyvaluep = 0x4c32ae0
73: 0x4c32ae0->{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 0x4c32930->{IDENT(259) 4:1.001 "t" []}
79: DEBUGF(f): astree.cc[97] free_ast():
80: free [4C32930]-> 4:1.1: IDENT: "t")
81: Stack now 0 2
82: Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
83: toknum = 256, yyvaluep = 0x4c32ae0
84: error(256)
85: )
86: Entering state 4
87: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
88: toknum = 259, yyvaluep = 0x4c32ae0
89: 0x4c32ae0->{IDENT(259) 5:3.000 "pi" []})
90: Error: discarding token IDENT (DEBUGF(f): astree.cc[79] yyprint():
91: toknum = 259, yyvaluep = 0x4c32ae0
92: 0x4c32ae0->{IDENT(259) 5:3.000 "pi" []})
93: DEBUGF(a): lyutils.cc[74] error_destructor():
94:
95: pi 0x4c32ae0->{IDENT(259) 5:3.000 "pi" []}
96: DEBUGF(f): astree.cc[97] free_ast():
97: free [4C32AE0]-> 5:3.0: IDENT: "pi")
98: Error: popping token error (DEBUGF(f): astree.cc[79] yyprint():
99: toknum = 256, yyvaluep = 0x4c32ae0
100: error(256)
101: )
102: Stack now 0 2
103: Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
104: toknum = 256, yyvaluep = 0x4c32ae0
105: error(256)
106: )
107: Entering state 4
108: Reading a token: --accepting rule at line 36 ("=")
109: DEBUGF(f): astree.cc[23] new_astree():
110: astree 0x4c32c20->{5:3.2: '=': "="}
111: Next token is token '=' (DEBUGF(f): astree.cc[79] yyprint():
112: toknum = 61, yyvaluep = 0x4c32c20
113: 0x4c32c20->{'=' (61) 5:3.002 "=" []})
114: Error: discarding token '=' (DEBUGF(f): astree.cc[79] yyprint():
115: toknum = 61, yyvaluep = 0x4c32c20
```

```
116: 0x4c32c20->{'=' (61) 5:3.002 "=" []})
117: DEBUGF(a): lyutils.cc[74] error_destructor():
118:
119: = 0x4c32c20->{'=' (61) 5:3.002 "=" []}
120: DEBUGF(f): astree.cc[97] free_ast():
121: free [4C32C20]-> 5:3.2: '=': "=")
122: Error: popping token error (DEBUGF(f): astree.cc[79] yyprint():
123: toknum = 256, yyvaluep = 0x4c32ae0
124: error(256)
125: )
126: Stack now 0 2
127: Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
128: toknum = 256, yyvaluep = 0x4c32c20
129: error(256)
130: )
131: Entering state 4
132: Reading a token: --accepting rule at line 33 ("3.14159265358979323846264
3383280")
133: DEBUGF(f): astree.cc[23] new_astree():
134: astree 0x4c32d60->{5:3.3: NUMBER: "3.141592653589793238462643383280"}
135: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
136: toknum = 260, yyvaluep = 0x4c32d60
137: 0x4c32d60->{NUMBER(260) 5:3.003 "3.141592653589793238462643383280" []})
138: Error: discarding token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
139: toknum = 260, yyvaluep = 0x4c32d60
140: 0x4c32d60->{NUMBER(260) 5:3.003 "3.141592653589793238462643383280" []})
141: DEBUGF(a): lyutils.cc[74] error_destructor():
142:
143: 3.141592653589793238462643383280 0x4c32d60->{NUMBER(260) 5:3.003 "3.1415
92653589793238462643383280" []}
144: DEBUGF(f): astree.cc[97] free_ast():
145: free [4C32D60]-> 5:3.3: NUMBER: "3.141592653589793238462643383280")
146: Error: popping token error (DEBUGF(f): astree.cc[79] yyprint():
147: toknum = 256, yyvaluep = 0x4c32c20
148: error(256)
149: )
150: Stack now 0 2
151: Shifting token error (DEBUGF(f): astree.cc[79] yyprint():
152: toknum = 256, yyvaluep = 0x4c32d60
153: error(256)
154: )
155: Entering state 4
156: Reading a token: --accepting rule at line 44 (";")
157: DEBUGF(f): astree.cc[23] new_astree():
158: astree 0x4c32ec0->{5:3.35: ';' : ";" }
159: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
160: toknum = 59, yyvaluep = 0x4c32ec0
161: 0x4c32ec0->{';' (59) 5:3.035 ";" []})
162: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
163: toknum = 59, yyvaluep = 0x4c32ec0
164: 0x4c32ec0->{';' (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 = 0x4c32d60
170: error(256)
171: )
```



```
172:    $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
173: toknum = 59, yyvaluep = 0x4c32ec0
174: 0x4c32ec0->{';' (59) 5:3.035 ";" []})
175: DEBUGF(f): astree.cc[97] free_ast():
176: free [4C32EC0]-> 5:3.35: ';' : ";"
177: -> $$ = nterm stmtseq ()
178: Stack now 0
179: Entering state 2
180: Reading a token: --accepting rule at line 31 ("
181: ")
182: --accepting rule at line 34 ("pi")
183: DEBUGF(f): astree.cc[23] new_astree():
184: astree 0x4c33000->{5:4.0: IDENT: "pi"}
185: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
186: toknum = 259, yyvaluep = 0x4c33000
187: 0x4c33000->{IDENT(259) 5:4.000 "pi" []})
188: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
189: toknum = 259, yyvaluep = 0x4c33000
190: 0x4c33000->{IDENT(259) 5:4.000 "pi" []})
191: Entering state 5
192: Reducing stack by rule 15 (line 59):
193:    $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
194: toknum = 259, yyvaluep = 0x4c33000
195: 0x4c33000->{IDENT(259) 5:4.000 "pi" []})
196: -> $$ = nterm expr ()
197: Stack now 0 2
198: Entering state 11
199: Reading a token: --accepting rule at line 44 (";"")
200: DEBUGF(f): astree.cc[23] new_astree():
201: astree 0x4c330e0->{5:4.2: ';' : ";" }
202: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
203: toknum = 59, yyvaluep = 0x4c330e0
204: 0x4c330e0->{';' (59) 5:4.002 ";" []})
205: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
206: toknum = 59, yyvaluep = 0x4c330e0
207: 0x4c330e0->{';' (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 = 0x4c330e0
214: 0x4c330e0->{';' (59) 5:4.002 ";" []})
215: DEBUGF(f): astree.cc[97] free_ast():
216: free [4C330E0]-> 5:4.2: ';' : ";"
217: DEBUGF(a): astree.cc[32] adopt1():
218: 0x4c2e3d0 (<<ROOT>>) adopting 0x4c33000 (pi)
219: -> $$ = nterm stmtseq ()
220: Stack now 0
221: Entering state 2
222: Reading a token: --accepting rule at line 31 ("
223: ")
224: --accepting rule at line 29 ("# 3 "test4.in" 2")
225: DEBUGF(m): lyutils.cc[97] scanner_include():
226: filename=test4.in, scan_linenr=2
227: --accepting rule at line 31 ("
228: ")
229: --accepting rule at line 29 ("# 1 "test4b.inh" 1")
```

```
230: DEBUGF(m): lyutils.cc[97] scanner_include():
231: filename=test4b.inh, scan_linenr=0
232: --accepting rule at line 31 ("
233: ")
234: --accepting rule at line 31 ("
235: ")
236: --accepting rule at line 34 ("a")
237: DEBUGF(f): astree.cc[23] new_astree():
238: astree 0x4c332f0->{7:2.0: IDENT: "a"}
239: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
240: toknum = 259, yyvaluep = 0x4c332f0
241: 0x4c332f0->{IDENT(259) 7:2.000 "a" []})
242: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
243: toknum = 259, yyvaluep = 0x4c332f0
244: 0x4c332f0->{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 = 0x4c332f0
249: 0x4c332f0->{IDENT(259) 7:2.000 "a" []})
250: -> $$ = nterm expr ()
251: Stack now 0 2
252: Entering state 11
253: Reading a token: --accepting rule at line 36 ("=")
254: DEBUGF(f): astree.cc[23] new_astree():
255: astree 0x4c33430->{7:2.1: '=': "="}
256: Next token is token '=' (DEBUGF(f): astree.cc[79] yyprint():
257: toknum = 61, yyvaluep = 0x4c33430
258: 0x4c33430->{'='(61) 7:2.001 "=" []})
259: Shifting token '=' (DEBUGF(f): astree.cc[79] yyprint():
260: toknum = 61, yyvaluep = 0x4c33430
261: 0x4c33430->{'='(61) 7:2.001 "=" []})
262: Entering state 16
263: Reading a token: --accepting rule at line 34 ("pi")
264: DEBUGF(f): astree.cc[23] new_astree():
265: astree 0x4c33510->{7:2.2: IDENT: "pi"}
266: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
267: toknum = 259, yyvaluep = 0x4c33510
268: 0x4c33510->{IDENT(259) 7:2.002 "pi" []})
269: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
270: toknum = 259, yyvaluep = 0x4c33510
271: 0x4c33510->{IDENT(259) 7:2.002 "pi" []})
272: Entering state 5
273: Reducing stack by rule 15 (line 59):
274: $1 = token IDENT (DEBUGF(f): astree.cc[79] yyprint():
275: toknum = 259, yyvaluep = 0x4c33510
276: 0x4c33510->{IDENT(259) 7:2.002 "pi" []})
277: -> $$ = nterm expr ()
278: Stack now 0 2 11 16
279: Entering state 24
280: Reading a token: --accepting rule at line 39 ("*")
281: DEBUGF(f): astree.cc[23] new_astree():
282: astree 0x4c335f0->{7:2.4: '*' : "*"}
283: Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
284: toknum = 42, yyvaluep = 0x4c335f0
285: 0x4c335f0->{'*'(42) 7:2.004 "*" []})
286: Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
287: toknum = 42, yyvaluep = 0x4c335f0
```

```
288: 0x4c335f0->{'*' (42) 7:2.004 "*" []})
289: Entering state 19
290: Reading a token: --accepting rule at line 34 ("r")
291: DEBUGF(f): astree.cc[23] new_astree():
292: astree 0x4c33730->{7:2.5: IDENT: "r"}
293: Next token is token IDENT (DEBUGF(f): astree.cc[79] yyprint():
294: toknum = 259, yyvaluep = 0x4c33730
295: 0x4c33730->{IDENT(259) 7:2.005 "r" []})
296: Shifting token IDENT (DEBUGF(f): astree.cc[79] yyprint():
297: toknum = 259, yyvaluep = 0x4c33730
298: 0x4c33730->{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 = 0x4c33730
303: 0x4c33730->{IDENT(259) 7:2.005 "r" []})
304: -> $$ = nterm expr ()
305: Stack now 0 2 11 16 24 19
306: Entering state 27
307: Reading a token: --accepting rule at line 41 ("^")
308: DEBUGF(f): astree.cc[23] new_astree():
309: astree 0x4c33870->{7:2.6: '^': "^"}
310: Next token is token '^' (DEBUGF(f): astree.cc[79] yyprint():
311: toknum = 94, yyvaluep = 0x4c33870
312: 0x4c33870->{'^' (94) 7:2.006 "^" []})
313: Shifting token '^' (DEBUGF(f): astree.cc[79] yyprint():
314: toknum = 94, yyvaluep = 0x4c33870
315: 0x4c33870->{'^' (94) 7:2.006 "^" []})
316: Entering state 21
317: Reading a token: --accepting rule at line 33 ("2")
318: DEBUGF(f): astree.cc[23] new_astree():
319: astree 0x4c339b0->{7:2.7: NUMBER: "2"}
320: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
321: toknum = 260, yyvaluep = 0x4c339b0
322: 0x4c339b0->{NUMBER(260) 7:2.007 "2" []})
323: Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
324: toknum = 260, yyvaluep = 0x4c339b0
325: 0x4c339b0->{NUMBER(260) 7:2.007 "2" []})
326: Entering state 6
327: Reducing stack by rule 16 (line 60):
328: $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
329: toknum = 260, yyvaluep = 0x4c339b0
330: 0x4c339b0->{NUMBER(260) 7:2.007 "2" []})
331: -> $$ = 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 44 (";")
335: DEBUGF(f): astree.cc[23] new_astree():
336: astree 0x4c33bf0->{7:2.8: ';'': ";"}
337: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
338: toknum = 59, yyvaluep = 0x4c33bf0
339: 0x4c33bf0->{';' (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 = 0x4c33870
344: 0x4c33870->{'^' (94) 7:2.006 "^" []})
345: $3 = nterm expr ()
```

```
346: DEBUGF(a): astree.cc[32] adopt1():
347: 0x4c33870 (^) adopting 0x4c33730 (r)
348: DEBUGF(a): astree.cc[32] adopt1():
349: 0x4c33870 (^) adopting 0x4c339b0 (2)
350: -> $$ = nterm expr ()
351: Stack now 0 2 11 16 24 19
352: Entering state 27
353: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
354: toknum = 59, yyvaluep = 0x4c33bf0
355: 0x4c33bf0->{';' (59) 7:2.008 ";" []})
356: Reducing stack by rule 9 (line 53):
357:   $1 = nterm expr ()
358:   $2 = token '*' (DEBUGF(f): astree.cc[79] yyprint():
359: toknum = 42, yyvaluep = 0x4c335f0
360: 0x4c335f0->{'*' (42) 7:2.004 "*" []})
361:   $3 = nterm expr ()
362: DEBUGF(a): astree.cc[32] adopt1():
363: 0x4c335f0 (*) adopting 0x4c33510 (pi)
364: DEBUGF(a): astree.cc[32] adopt1():
365: 0x4c335f0 (*) adopting 0x4c33870 (^)
366: -> $$ = 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 = 0x4c33bf0
371: 0x4c33bf0->{';' (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 = 0x4c33430
376: 0x4c33430->{'=' (61) 7:2.001 "=" []})
377:   $3 = nterm expr ()
378: DEBUGF(a): astree.cc[32] adopt1():
379: 0x4c33430 (=) adopting 0x4c332f0 (a)
380: DEBUGF(a): astree.cc[32] adopt1():
381: 0x4c33430 (=) adopting 0x4c335f0 (*)
382: -> $$ = nterm expr ()
383: Stack now 0 2
384: Entering state 11
385: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
386: toknum = 59, yyvaluep = 0x4c33bf0
387: 0x4c33bf0->{';' (59) 7:2.008 ";" []})
388: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
389: toknum = 59, yyvaluep = 0x4c33bf0
390: 0x4c33bf0->{';' (59) 7:2.008 ";" []})
391: Entering state 22
392: Reducing stack by rule 2 (line 44):
393:   $1 = nterm stmtseq ()
394:   $2 = nterm expr ()
395:   $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
396: toknum = 59, yyvaluep = 0x4c33bf0
397: 0x4c33bf0->{';' (59) 7:2.008 ";" []})
398: DEBUGF(f): astree.cc[97] free_ast():
399: free [4C33BF0]-> 7:2.8: ';' : ";"
400: DEBUGF(a): astree.cc[32] adopt1():
401: 0x4c2e3d0 (<<ROOT>>) adopting 0x4c33430 (=)
402: -> $$ = nterm stmtseq ()
403: Stack now 0
```

```
404: Entering state 2
405: Reading a token: --accepting rule at line 31 ("
406: ")
407: --accepting rule at line 33 ("3.141592653589793238462643383280")
408: DEBUGF(f): astree.cc[23] new_astree():
409: astree 0x4c33f00->{7:3.0: NUMBER: "3.141592653589793238462643383280"}
410: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
411: toknum = 260, yyvaluep = 0x4c33f00
412: 0x4c33f00->{NUMBER(260) 7:3.000 "3.141592653589793238462643383280" []})
413: Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
414: toknum = 260, yyvaluep = 0x4c33f00
415: 0x4c33f00->{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 = 0x4c33f00
420: 0x4c33f00->{NUMBER(260) 7:3.000 "3.141592653589793238462643383280" []})
421: -> $$ = nterm expr ()
422: Stack now 0 2
423: Entering state 11
424: Reading a token: --accepting rule at line 44 (";")
425: DEBUGF(f): astree.cc[23] new_astree():
426: astree 0x4c34000->{7:3.32: ';' : ";" }
427: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
428: toknum = 59, yyvaluep = 0x4c34000
429: 0x4c34000->{';' (59) 7:3.032 ";" []})
430: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
431: toknum = 59, yyvaluep = 0x4c34000
432: 0x4c34000->{';' (59) 7:3.032 ";" []})
433: Entering state 22
434: Reducing stack by rule 2 (line 44):
435: $1 = nterm stmtseq ()
436: $2 = nterm expr ()
437: $3 = token ';' (DEBUGF(f): astree.cc[79] yyprint():
438: toknum = 59, yyvaluep = 0x4c34000
439: 0x4c34000->{';' (59) 7:3.032 ";" []})
440: DEBUGF(f): astree.cc[97] free_ast():
441: free [4C34000]-> 7:3.32: ';' : ";" )
442: DEBUGF(a): astree.cc[32] adopt1():
443: 0x4c2e3d0 (<<ROOT>>) adopting 0x4c33f00 (3.14159265358979323846264338328
0)
444: -> $$ = nterm stmtseq ()
445: Stack now 0
446: Entering state 2
447: Reading a token: --accepting rule at line 31 ("
448: ")
449: --accepting rule at line 29 ("# 4 "test4.in" 2")
450: DEBUGF(m): lyutils.cc[97] scanner_include():
451: filename=test4.in, scan_linenr=3
452: --accepting rule at line 31 ("
453: ")
454: --accepting rule at line 33 ("3")
455: DEBUGF(f): astree.cc[23] new_astree():
456: astree 0x4c34270->{8:4.0: NUMBER: "3"}
457: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
458: toknum = 260, yyvaluep = 0x4c34270
459: 0x4c34270->{NUMBER(260) 8:4.000 "3" []})
460: Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
```

```
461: toknum = 260, yyvaluep = 0x4c34270
462: 0x4c34270->{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 = 0x4c34270
467: 0x4c34270->{NUMBER(260) 8:4.000 "3" []})
468: -> $$ = nterm expr ()
469: Stack now 0 2
470: Entering state 11
471: Reading a token: --accepting rule at line 39 ("*")
472: DEBUGF(f): astree.cc[23] new_astree():
473: astree 0x4c343b0->{8:4.1: '*' : ""}
474: Next token is token '*' (DEBUGF(f): astree.cc[79] yyprint():
475: toknum = 42, yyvaluep = 0x4c343b0
476: 0x4c343b0->{'*' (42) 8:4.001 "" []})
477: Shifting token '*' (DEBUGF(f): astree.cc[79] yyprint():
478: toknum = 42, yyvaluep = 0x4c343b0
479: 0x4c343b0->{'*' (42) 8:4.001 "" []})
480: Entering state 19
481: Reading a token: --accepting rule at line 33 ("4")
482: DEBUGF(f): astree.cc[23] new_astree():
483: astree 0x4c34490->{8:4.2: NUMBER: "4"}
484: Next token is token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
485: toknum = 260, yyvaluep = 0x4c34490
486: 0x4c34490->{NUMBER(260) 8:4.002 "4" []})
487: Shifting token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
488: toknum = 260, yyvaluep = 0x4c34490
489: 0x4c34490->{NUMBER(260) 8:4.002 "4" []})
490: Entering state 6
491: Reducing stack by rule 16 (line 60):
492:     $1 = token NUMBER (DEBUGF(f): astree.cc[79] yyprint():
493: toknum = 260, yyvaluep = 0x4c34490
494: 0x4c34490->{NUMBER(260) 8:4.002 "4" []})
495: -> $$ = nterm expr ()
496: Stack now 0 2 11 19
497: Entering state 27
498: Reading a token: --accepting rule at line 44 (";")
499: DEBUGF(f): astree.cc[23] new_astree():
500: astree 0x4c345d0->{8:4.3: ';' : ";" }
501: Next token is token ';' (DEBUGF(f): astree.cc[79] yyprint():
502: toknum = 59, yyvaluep = 0x4c345d0
503: 0x4c345d0->{';' (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 = 0x4c343b0
508: 0x4c343b0->{'*' (42) 8:4.001 "" []})
509:     $3 = nterm expr ()
510: DEBUGF(a): astree.cc[32] adopt1():
511: 0x4c343b0 (*) adopting 0x4c34270 (3)
512: DEBUGF(a): astree.cc[32] adopt1():
513: 0x4c343b0 (*) adopting 0x4c34490 (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 = 0x4c345d0
```

```
519: 0x4c345d0->{';' (59) 8:4.003 ";" []})
520: Shifting token ';' (DEBUGF(f): astree.cc[79] yyprint():
521: toknum = 59, yyvaluep = 0x4c345d0
522: 0x4c345d0->{';' (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 = 0x4c345d0
529: 0x4c345d0->{';' (59) 8:4.003 ";" []})
530: DEBUGF(f): astree.cc[97] free_ast():
531: free [4C345D0]-> 8:4.3: ';' : ";"
532: DEBUGF(a): astree.cc[32] adopt1():
533: 0x4c2e3d0 (<<ROOT>>) adopting 0x4c343b0 (*)
534: -> $$ = nterm stmtseq ()
535: Stack now 0
536: Entering state 2
537: Reading a token: --accepting rule at line 31 ("
538: ")
539: --(end of buffer or a NUL)
540: --EOF (start condition 0)
541: Now at end of input.
542: Reducing stack by rule 1 (line 41):
543:   $1 = nterm stmtseq ()
544: -> $$ = 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 = 0x4c345d0
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 = 0x4c345d0
556: $end(0)
557: )
558: Cleanup: popping nterm program ()
559: DEBUGF(a): main.cc[87] main():
560:
561: <<ROOT>> 0x4c2e3d0->{ROOT(258) 0:0.000 "<<ROOT>>" [0x4c33000 0x4c33430 0
x4c33f00 0x4c343b0]}
562:   pi 0x4c33000->{IDENT(259) 5:4.000 "pi" []}
563:   = 0x4c33430->{'=' (61) 7:2.001 "=" [0x4c332f0 0x4c335f0]}
564:   a 0x4c332f0->{IDENT(259) 7:2.000 "a" []}
565:   * 0x4c335f0->{'*' (42) 7:2.004 "*" [0x4c33510 0x4c33870]}
566:     pi 0x4c33510->{IDENT(259) 7:2.002 "pi" []}
567:     ^ 0x4c33870->{'^' (94) 7:2.006 "^" [0x4c33730 0x4c339b0]}
568:     r 0x4c33730->{IDENT(259) 7:2.005 "r" []}
569:     2 0x4c339b0->{NUMBER(260) 7:2.007 "2" []}
570:     3.141592653589793238462643383280 0x4c33f00->{NUMBER(260) 7:3.000 "3.1
41592653589793238462643383280" []}
571:     * 0x4c343b0->{'*' (42) 8:4.001 "*" [0x4c34270 0x4c34490]}
572:     3 0x4c34270->{NUMBER(260) 8:4.000 "3" []}
573:     4 0x4c34490->{NUMBER(260) 8:4.002 "4" []}
574: DEBUGF(f): astree.cc[97] free_ast():
```

```
575: free [4C34490]-> 8:4.2: NUMBER: "4")
576: DEBUGF(f): astree.cc[97] free_ast():
577: free [4C34270]-> 8:4.0: NUMBER: "3")
578: DEBUGF(f): astree.cc[97] free_ast():
579: free [4C343B0]-> 8:4.1: '*' : "*"")
580: DEBUGF(f): astree.cc[97] free_ast():
581: free [4C33F00]-> 7:3.0: NUMBER: "3.141592653589793238462643383280")
582: DEBUGF(f): astree.cc[97] free_ast():
583: free [4C339B0]-> 7:2.7: NUMBER: "2")
584: DEBUGF(f): astree.cc[97] free_ast():
585: free [4C33730]-> 7:2.5: IDENT: "r")
586: DEBUGF(f): astree.cc[97] free_ast():
587: free [4C33870]-> 7:2.6: '^' : "^")
588: DEBUGF(f): astree.cc[97] free_ast():
589: free [4C33510]-> 7:2.2: IDENT: "pi")
590: DEBUGF(f): astree.cc[97] free_ast():
591: free [4C335F0]-> 7:2.4: '*' : "*"")
592: DEBUGF(f): astree.cc[97] free_ast():
593: free [4C332F0]-> 7:2.0: IDENT: "a")
594: DEBUGF(f): astree.cc[97] free_ast():
595: free [4C33430]-> 7:2.1: '=' : "=")
596: DEBUGF(f): astree.cc[97] free_ast():
597: free [4C33000]-> 5:4.0: IDENT: "pi")
598: DEBUGF(f): astree.cc[97] free_ast():
599: free [4C2E3D0]-> 0:0.0: ROOT: "<<ROOT>>")
600: DEBUGF(s): main.cc[92] main():
601:
602: stringset[ 1]:      831242270113464010 0x4c32a18->"t"
603: stringset[ 3]:      8248777770799913213 0x4c34358->"3"
604:      6637313742931709005 0x4c336d8->"*"
605:      11597697714117577063 0x4c32d08->"="
606:      5344662657899890615 0x4c32fa8->" ; "
607: stringset[ 6]:      3729804957429652673 0x4c2e4c8->"<<ROOT>>"
608: stringset[ 9]:      4551451650890805270 0x4c33a98->"2"
609: stringset[11]:      1998636459596678225 0x4c33958->"^"
610: stringset[12]:      14518329498511883088 0x4c33818->"r"
611:      15530967369654070964 0x4c32bc8->"pi"
612: stringset[17]:      4993892634952068459 0x4c333d8->"a"
613: stringset[20]:      17027087115628340017 0x4c32e68->"3.141592653589793238
462643383280"
614: stringset[21]:      16215888864653804456 0x4c34578->"4"
615: load_factor = 0.565
616: bucket_count = 23
617: max_bucket_size = 4
618: ==8984==
619: ==8984== HEAP SUMMARY:
620: ==8984==      in use at exit: 0 bytes in 0 blocks
621: ==8984==    total heap usage: 90 allocs, 90 frees, 20,145 bytes allocated
622: ==8984==
623: ==8984== All heap blocks were freed -- no leaks are possible
624: ==8984==
625: ==8984== For counts of detected and suppressed errors, rerun with: -v
626: ==8984== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
627: EXIT STATUS 1
```