```
 1: // $Id: ambiguous-else.y,v 1.1 2011-10-28 18:07:07-07 - - $
 2:
 3: // Example of solving the problem of the dangling else with an
 4: // ambiguous grammar and precedence declarations.
 5:
 6: %verbose
 7:
 8: %token IF WHILE
 9: %right ELSE
10: %start program
11:
12: %%
13:
14: program    : program statement
15:            |
16:            ;
17:
18: statement  : ifhead statement ELSE statement
19:            | ifhead statement %prec ELSE
20:            | whilehead statement
21:            | otherstmt
22:            ;
23:
24: ifhead     : IF '(' expr ')'
25:            ;
26:
27: whilehead  : WHILE '(' expr ')'
28:            ;
29:
30: otherstmt  : expr ';'
31:            ;
32:
33: expr       : 'x'
34:            ;
35:
36: %%
37:
```

```
bison -v ambiguous-else.y
::::::::::::::
ambiguous-else.stdout
::::::::::::::
::::::::::::::
ambiguous-else.stderr
::::::::::::::
::::::::::::::
ambiguous-else.output
::::::::::::::
Grammar

    0 $accept: program $end

    1 program: program statement
    2        | /* empty */

    3 statement: ifhead statement ELSE statement
    4          | ifhead statement
    5          | whilehead statement
    6          | otherstmt

    7 ifhead: IF '(' expr ')'

    8 whilehead: WHILE '(' expr ')'

    9 otherstmt: expr ';'

   10 expr: 'x'


Terminals, with rules where they appear

$end (0) 0
'(' (40) 7 8
')' (41) 7 8
';' (59) 9
'x' (120) 10
error (256)
IF (258) 7
WHILE (259) 8
ELSE (260) 3


Nonterminals, with rules where they appear

$accept (10)
    on left: 0
program (11)
    on left: 1 2, on right: 0 1
statement (12)
    on left: 3 4 5 6, on right: 1 3 4 5
ifhead (13)
    on left: 7, on right: 3 4
whilehead (14)
    on left: 8, on right: 5
otherstmt (15)
    on left: 9, on right: 6
expr (16)
    on left: 10, on right: 7 8 9


state 0

```

```
  65:     0 $accept: . program $end
  66:
  67:     $default  reduce using rule 2 (program)
  68:
  69:     program  go to state 1
  70:
  71:
  72: state 1
  73:
  74:     0 $accept: program . $end
  75:     1 program: program . statement
  76:
  77:     $end   shift, and go to state 2
  78:     IF     shift, and go to state 3
  79:     WHILE  shift, and go to state 4
  80:     'x'    shift, and go to state 5
  81:
  82:     statement  go to state 6
  83:     ifhead     go to state 7
  84:     whilehead  go to state 8
  85:     otherstmt  go to state 9
  86:     expr       go to state 10
  87:
  88:
  89: state 2
  90:
  91:     0 $accept: program $end .
  92:
  93:     $default  accept
  94:
  95:
  96: state 3
  97:
  98:     7 ifhead: IF . '(' expr ')'
  99:
 100:     '('  shift, and go to state 11
 101:
 102:
 103: state 4
 104:
 105:     8 whilehead: WHILE . '(' expr ')'
 106:
 107:     '('  shift, and go to state 12
 108:
 109:
 110: state 5
 111:
 112:    10 expr: 'x' .
 113:
 114:     $default  reduce using rule 10 (expr)
 115:
 116:
 117: state 6
 118:
 119:     1 program: program statement .
 120:
 121:     $default  reduce using rule 1 (program)
 122:
 123:
 124: state 7
 125:
 126:     3 statement: ifhead . statement ELSE statement
 127:     4          | ifhead . statement
 128:
```

```
129:     IF     shift, and go to state 3
130:     WHILE  shift, and go to state 4
131:     'x'    shift, and go to state 5
132:
133:     statement  go to state 13
134:     ifhead     go to state 7
135:     whilehead  go to state 8
136:     otherstmt  go to state 9
137:     expr       go to state 10
138:
139:
140: state 8
141:
142:     5 statement: whilehead . statement
143:
144:     IF     shift, and go to state 3
145:     WHILE  shift, and go to state 4
146:     'x'    shift, and go to state 5
147:
148:     statement  go to state 14
149:     ifhead     go to state 7
150:     whilehead  go to state 8
151:     otherstmt  go to state 9
152:     expr       go to state 10
153:
154:
155: state 9
156:
157:     6 statement: otherstmt .
158:
159:     $default  reduce using rule 6 (statement)
160:
161:
162: state 10
163:
164:     9 otherstmt: expr . ';'
165:
166:     ';'  shift, and go to state 15
167:
168:
169: state 11
170:
171:     7 ifhead: IF '(' . expr ')'
172:
173:     'x'  shift, and go to state 5
174:
175:     expr  go to state 16
176:
177:
178: state 12
179:
180:     8 whilehead: WHILE '(' . expr ')'
181:
182:     'x'  shift, and go to state 5
183:
184:     expr  go to state 17
185:
186:
187: state 13
188:
189:     3 statement: ifhead statement . ELSE statement
190:     4          | ifhead statement .
191:
192:     ELSE  shift, and go to state 18
```

```
193:
194:     $default  reduce using rule 4 (statement)
195:
196:
197: state 14
198:
199:     5 statement: whilehead statement .
200:
201:     $default  reduce using rule 5 (statement)
202:
203:
204: state 15
205:
206:     9 otherstmt: expr ';' .
207:
208:     $default  reduce using rule 9 (otherstmt)
209:
210:
211: state 16
212:
213:     7 ifhead: IF '(' expr . ')'
214:
215:     ')'  shift, and go to state 19
216:
217:
218: state 17
219:
220:     8 whilehead: WHILE '(' expr . ')'
221:
222:     ')'  shift, and go to state 20
223:
224:
225: state 18
226:
227:     3 statement: ifhead statement ELSE . statement
228:
229:     IF     shift, and go to state 3
230:     WHILE  shift, and go to state 4
231:     'x'    shift, and go to state 5
232:
233:     statement  go to state 21
234:     ifhead     go to state 7
235:     whilehead  go to state 8
236:     otherstmt  go to state 9
237:     expr       go to state 10
238:
239:
240: state 19
241:
242:     7 ifhead: IF '(' expr ')' .
243:
244:     $default  reduce using rule 7 (ifhead)
245:
246:
247: state 20
248:
249:     8 whilehead: WHILE '(' expr ')' .
250:
251:     $default  reduce using rule 8 (whilehead)
252:
253:
254: state 21
255:
256:     3 statement: ifhead statement ELSE statement .
```

```
257:
258:     $default  reduce using rule 3 (statement)
```

```
 1: // $Id: unambiguous-else.y,v 1.1 2011-10-28 18:07:07-07 - - $
 2:
 3: // Example of solving the problem of the dangling else with an
 4: // ambiguous grammar and precedence declarations.
 5:
 6: %verbose
 7:
 8: %token IF WHILE
 9: %right ELSE
10: %start program
11:
12: %%
13:
14: program    : program statement
15:            |
16:            ;
17:
18: statement  : closedstmt
19:            | openstmt
20:            ;
21:
22: closedstmt : ifhead closedstmt ELSE closedstmt
23:            | whilehead closedstmt
24:            | otherstmt
25:            ;
26:
27: openstmt   : ifhead closedstmt ELSE openstmt
28:            | ifhead statement
29:            | whilehead openstmt
30:
31: ifhead     : IF '(' expr ')'
32:            ;
33:
34: whilehead  : WHILE '(' expr ')'
35:            ;
36:
37: otherstmt  : expr ';'
38:            ;
39:
40: expr       : 'x'
41:            ;
42:
43: %%
44:
```

```
bison -v unambiguous-else.y
::::::::::::::
unambiguous-else.stdout
::::::::::::::
::::::::::::::
unambiguous-else.stderr
::::::::::::::
::::::::::::::
unambiguous-else.output
::::::::::::::
Grammar

    0 $accept: program $end

    1 program: program statement
    2        | /* empty */

    3 statement: closedstmt
    4          | openstmt

    5 closedstmt: ifhead closedstmt ELSE closedstmt
    6           | whilehead closedstmt
    7           | otherstmt

    8 openstmt: ifhead closedstmt ELSE openstmt
    9         | ifhead statement
   10         | whilehead openstmt

   11 ifhead: IF '(' expr ')'

   12 whilehead: WHILE '(' expr ')'

   13 otherstmt: expr ';'

   14 expr: 'x'


Terminals, with rules where they appear

$end (0) 0
'(' (40) 11 12
')' (41) 11 12
';' (59) 13
'x' (120) 14
error (256)
IF (258) 11
WHILE (259) 12
ELSE (260) 5 8


Nonterminals, with rules where they appear

$accept (10)
    on left: 0
program (11)
    on left: 1 2, on right: 0 1
statement (12)
    on left: 3 4, on right: 1 9
closedstmt (13)
    on left: 5 6 7, on right: 3 5 6 8
openstmt (14)
    on left: 8 9 10, on right: 4 8 10
ifhead (15)
    on left: 11, on right: 5 8 9
```

```
whilehead (16)
    on left: 12, on right: 6 10
otherstmt (17)
    on left: 13, on right: 7
expr (18)
    on left: 14, on right: 11 12 13


state 0

    0 $accept: . program $end

    $default  reduce using rule 2 (program)

    program  go to state 1


state 1

    0 $accept: program . $end
    1 program: program . statement

    $end   shift, and go to state 2
    IF     shift, and go to state 3
    WHILE  shift, and go to state 4
    'x'    shift, and go to state 5

    statement   go to state 6
    closedstmt  go to state 7
    openstmt    go to state 8
    ifhead      go to state 9
    whilehead   go to state 10
    otherstmt   go to state 11
    expr        go to state 12


state 2

    0 $accept: program $end .

    $default  accept


state 3

   11 ifhead: IF . '(' expr ')'

    '('  shift, and go to state 13


state 4

   12 whilehead: WHILE . '(' expr ')'

    '('  shift, and go to state 14


state 5

   14 expr: 'x' .

    $default  reduce using rule 14 (expr)


```

```
state 6

    1 program: program statement .

    $default  reduce using rule 1 (program)


state 7

    3 statement: closedstmt .

    $default  reduce using rule 3 (statement)


state 8

    4 statement: openstmt .

    $default  reduce using rule 4 (statement)


state 9

    5 closedstmt: ifhead . closedstmt ELSE closedstmt
    8 openstmt: ifhead . closedstmt ELSE openstmt
    9         | ifhead . statement

    IF     shift, and go to state 3
    WHILE  shift, and go to state 4
    'x'    shift, and go to state 5

    statement   go to state 15
    closedstmt  go to state 16
    openstmt    go to state 8
    ifhead      go to state 9
    whilehead   go to state 10
    otherstmt   go to state 11
    expr        go to state 12


state 10

    6 closedstmt: whilehead . closedstmt
   10 openstmt: whilehead . openstmt

    IF     shift, and go to state 3
    WHILE  shift, and go to state 4
    'x'    shift, and go to state 5

    closedstmt  go to state 17
    openstmt    go to state 18
    ifhead      go to state 9
    whilehead   go to state 10
    otherstmt   go to state 11
    expr        go to state 12


state 11

    7 closedstmt: otherstmt .

    $default  reduce using rule 7 (closedstmt)


```

```
state 12

    13 otherstmt: expr . ';'

    ';'  shift, and go to state 19


state 13

    11 ifhead: IF '(' . expr ')'

    'x'  shift, and go to state 5

    expr  go to state 20


state 14

    12 whilehead: WHILE '(' . expr ')'

    'x'  shift, and go to state 5

    expr  go to state 21


state 15

    9 openstmt: ifhead statement .

    $default  reduce using rule 9 (openstmt)


state 16

    3 statement: closedstmt .
    5 closedstmt: ifhead closedstmt . ELSE closedstmt
    8 openstmt: ifhead closedstmt . ELSE openstmt

    ELSE  shift, and go to state 22

    $default  reduce using rule 3 (statement)


state 17

    6 closedstmt: whilehead closedstmt .

    $default  reduce using rule 6 (closedstmt)


state 18

    10 openstmt: whilehead openstmt .

    $default  reduce using rule 10 (openstmt)


state 19

    13 otherstmt: expr ';' .

    $default  reduce using rule 13 (otherstmt)


```

```
257: state 20
258:
259:    11 ifhead: IF '(' expr . ')'
260:
261:     ')'  shift, and go to state 23
262:
263:
264: state 21
265:
266:    12 whilehead: WHILE '(' expr . ')'
267:
268:     ')'  shift, and go to state 24
269:
270:
271: state 22
272:
273:     5 closedstmt: ifhead closedstmt ELSE . closedstmt
274:     8 openstmt: ifhead closedstmt ELSE . openstmt
275:
276:     IF     shift, and go to state 3
277:     WHILE  shift, and go to state 4
278:     'x'    shift, and go to state 5
279:
280:     closedstmt  go to state 25
281:     openstmt    go to state 26
282:     ifhead      go to state 9
283:     whilehead   go to state 10
284:     otherstmt   go to state 11
285:     expr        go to state 12
286:
287:
288: state 23
289:
290:    11 ifhead: IF '(' expr ')' .
291:
292:     $default  reduce using rule 11 (ifhead)
293:
294:
295: state 24
296:
297:    12 whilehead: WHILE '(' expr ')' .
298:
299:     $default  reduce using rule 12 (whilehead)
300:
301:
302: state 25
303:
304:     5 closedstmt: ifhead closedstmt ELSE closedstmt .
305:
306:     $default  reduce using rule 5 (closedstmt)
307:
308:
309: state 26
310:
311:     8 openstmt: ifhead closedstmt ELSE openstmt .
312:
313:     $default  reduce using rule 8 (openstmt)
```

```
 1: // $Id: reduce-reduce.y,v 1.3 2011-10-28 18:41:59-07 - - $
 2:
 3: // Example of a reduce/reduce conflict.
 4:
 5: %verbose
 6:
 7: %token CHAR INT IDENT
 8: %start program
 9:
10: %%
11:
12: program     : program statement
13:             |
14:             ;
15:
16: statement   : declaration ';'
17:             | expression ';'
18:             ;
19:
20: declaration : basetype '[' ']' IDENT
21:             | basetype IDENT
22:             ;
23:
24: basetype    : CHAR
25:             | INT
26:             | IDENT
27:             ;
28:
29: expression  : expression '[' expression ']'
30:             | IDENT
31:             ;
32:
33: %%
34:
```

```
bison -v reduce-reduce.y
::::::::::::::
reduce-reduce.stdout
::::::::::::::
::::::::::::::
reduce-reduce.stderr
::::::::::::::
reduce-reduce.y: conflicts: 1 reduce/reduce
::::::::::::::
reduce-reduce.output
::::::::::::::
State 5 conflicts: 1 reduce/reduce


Grammar

    0 $accept: program $end

    1 program: program statement
    2        | /* empty */

    3 statement: declaration ';'
    4          | expression ';'

    5 declaration: basetype '[' ']' IDENT
    6            | basetype IDENT

    7 basetype: CHAR
    8         | INT
    9         | IDENT

   10 expression: expression '[' expression ']'
   11           | IDENT


Terminals, with rules where they appear

$end (0) 0
';' (59) 3 4
'[' (91) 5 10
']' (93) 5 10
error (256)
CHAR (258) 7
INT (259) 8
IDENT (260) 5 6 9 11


Nonterminals, with rules where they appear

$accept (9)
    on left: 0
program (10)
    on left: 1 2, on right: 0 1
statement (11)
    on left: 3 4, on right: 1
declaration (12)
    on left: 5 6, on right: 3
basetype (13)
    on left: 7 8 9, on right: 5 6
expression (14)
    on left: 10 11, on right: 4 10


state 0
```

```
 65:
 66:     0 $accept: . program $end
 67:
 68:     $default  reduce using rule 2 (program)
 69:
 70:     program  go to state 1
 71:
 72:
 73: state 1
 74:
 75:     0 $accept: program . $end
 76:     1 program: program . statement
 77:
 78:     $end   shift, and go to state 2
 79:     CHAR   shift, and go to state 3
 80:     INT    shift, and go to state 4
 81:     IDENT  shift, and go to state 5
 82:
 83:     statement    go to state 6
 84:     declaration  go to state 7
 85:     basetype     go to state 8
 86:     expression   go to state 9
 87:
 88:
 89: state 2
 90:
 91:     0 $accept: program $end .
 92:
 93:     $default  accept
 94:
 95:
 96: state 3
 97:
 98:     7 basetype: CHAR .
 99:
100:     $default  reduce using rule 7 (basetype)
101:
102:
103: state 4
104:
105:     8 basetype: INT .
106:
107:     $default  reduce using rule 8 (basetype)
108:
109:
110: state 5
111:
112:     9 basetype: IDENT .
113:    11 expression: IDENT .
114:
115:     ';'       reduce using rule 11 (expression)
116:     '['       reduce using rule 9 (basetype)
117:     '['       [reduce using rule 11 (expression)]
118:     $default  reduce using rule 9 (basetype)
119:
120:
121: state 6
122:
123:     1 program: program statement .
124:
125:     $default  reduce using rule 1 (program)
126:
127:
128: state 7
```

```
129:
130:     3 statement: declaration . ';'
131:
132:     ';'  shift, and go to state 10
133:
134:
135: state 8
136:
137:     5 declaration: basetype . '[' ']' IDENT
138:     6            | basetype . IDENT
139:
140:     IDENT  shift, and go to state 11
141:     '['    shift, and go to state 12
142:
143:
144: state 9
145:
146:     4 statement: expression . ';'
147:    10 expression: expression . '[' expression ']'
148:
149:     ';'  shift, and go to state 13
150:     '['  shift, and go to state 14
151:
152:
153: state 10
154:
155:     3 statement: declaration ';' .
156:
157:     $default  reduce using rule 3 (statement)
158:
159:
160: state 11
161:
162:     6 declaration: basetype IDENT .
163:
164:     $default  reduce using rule 6 (declaration)
165:
166:
167: state 12
168:
169:     5 declaration: basetype '[' . ']' IDENT
170:
171:     ']'  shift, and go to state 15
172:
173:
174: state 13
175:
176:     4 statement: expression ';' .
177:
178:     $default  reduce using rule 4 (statement)
179:
180:
181: state 14
182:
183:    10 expression: expression '[' . expression ']'
184:
185:     IDENT  shift, and go to state 16
186:
187:     expression  go to state 17
188:
189:
190: state 15
191:
192:     5 declaration: basetype '[' ']' . IDENT
```

```
193:
194:     IDENT  shift, and go to state 18
195:
196:
197: state 16
198:
199:    11 expression: IDENT .
200:
201:     $default  reduce using rule 11 (expression)
202:
203:
204: state 17
205:
206:    10 expression: expression . '[' expression ']'
207:    10           | expression '[' expression . ']'
208:
209:     '['  shift, and go to state 14
210:     ']'  shift, and go to state 19
211:
212:
213: state 18
214:
215:     5 declaration: basetype '[' ']' IDENT .
216:
217:     $default  reduce using rule 5 (declaration)
218:
219:
220: state 19
221:
222:    10 expression: expression '[' expression ']' .
223:
224:     $default  reduce using rule 10 (expression)
```

```
 1:
 2: %start S
 3: %token a b c d
 4:
 5: %%
 6:
 7: S : A a ;
 8: S : b A c ;
 9: S : B c ;
10: S : b B a ;
11: A : d ;
12: B : d ;
13:
14: %%
15:
```

```
bison -v notlalr1.y
::::::::::::::
notlalr1.stdout
::::::::::::::
::::::::::::::
notlalr1.stderr
::::::::::::::
notlalr1.y: conflicts: 2 reduce/reduce
notlalr1.y:12.5: warning: rule never reduced because of conflicts: B: d
::::::::::::::
notlalr1.output
::::::::::::::
Rules never reduced

    6 B: d


State 2 conflicts: 2 reduce/reduce


Grammar

    0 $accept: S $end

    1 S: A a
    2  | b A c
    3  | B c
    4  | b B a

    5 A: d

    6 B: d


Terminals, with rules where they appear

$end (0) 0
error (256)
a (258) 1 4
b (259) 2 4
c (260) 2 3
d (261) 5 6


Nonterminals, with rules where they appear

$accept (7)
    on left: 0
S (8)
    on left: 1 2 3 4, on right: 0
A (9)
    on left: 5, on right: 1 2
B (10)
    on left: 6, on right: 3 4


state 0

    0 $accept: . S $end

    b  shift, and go to state 1
    d  shift, and go to state 2

    S  go to state 3
```

```
 65:     A  go to state 4
 66:     B  go to state 5
 67:
 68:
 69: state 1
 70:
 71:     2 S: b . A c
 72:     4  | b . B a
 73:
 74:     d  shift, and go to state 2
 75:
 76:     A  go to state 6
 77:     B  go to state 7
 78:
 79:
 80: state 2
 81:
 82:     5 A: d .
 83:     6 B: d .
 84:
 85:     a         reduce using rule 5 (A)
 86:     a         [reduce using rule 6 (B)]
 87:     c         reduce using rule 5 (A)
 88:     c         [reduce using rule 6 (B)]
 89:     $default  reduce using rule 5 (A)
 90:
 91:
 92: state 3
 93:
 94:     0 $accept: S . $end
 95:
 96:     $end  shift, and go to state 8
 97:
 98:
 99: state 4
100:
101:     1 S: A . a
102:
103:     a  shift, and go to state 9
104:
105:
106: state 5
107:
108:     3 S: B . c
109:
110:     c  shift, and go to state 10
111:
112:
113: state 6
114:
115:     2 S: b A . c
116:
117:     c  shift, and go to state 11
118:
119:
120: state 7
121:
122:     4 S: b B . a
123:
124:     a  shift, and go to state 12
125:
126:
127: state 8
128:
```

```
129:     0 $accept: S $end .
130:
131:     $default  accept
132:
133:
134: state 9
135:
136:     1 S: A a .
137:
138:     $default  reduce using rule 1 (S)
139:
140:
141: state 10
142:
143:     3 S: B c .
144:
145:     $default  reduce using rule 3 (S)
146:
147:
148: state 11
149:
150:     2 S: b A c .
151:
152:     $default  reduce using rule 2 (S)
153:
154:
155: state 12
156:
157:     4 S: b B a .
158:
159:     $default  reduce using rule 4 (S)
```