The Ternery Operator – ?:

$expr_1$? $expr_2$: $expr_3$

Interpretation:

If $expr_1$ is true do $expr_2$

If $expr_1$ is false do $expr_3$

Problem with ?:

We currently have the productions:

stmt: cond ‘?’ stmt colonpart

colonpart: ‘:’ stmt

There are two problems with these productions

1. ?: is allowed only at the statement level
2. The targets of the ternery operator can only be statements – they really should be expressions

Solution to ?: Problem

Change productions to:

stmt: cond ‘?’ expr colonpart
colonpart: ‘:’ expr
and
expr: cond ‘?’ expr colonpart

Then we can insert statements in our program like:

b > c ? d : e;
and
a = b > c ? d : e;

However, these productions generate shift/reduce errors

What Causes the Shift/Reduce

Should a = b ? c : d + e;

be interpreted as:

(a = b ? c : d) + e;

or

a = b ? c : (d + e);