Daily Question

(due October 28, 2005)

Consider the grammar below for a simple fragment of C:

S	$\rightarrow$	*statement-list*
statement-list	$\rightarrow$	*statement*
	$\mid$	*statement-list statement*
statement	$\rightarrow$	*assignment*
	$\mid$	*if-statement*
	$\mid$	*compound-statement*
assignment	$\rightarrow$	**id = *expr* ;**
if-statement	$\rightarrow$	**if (** *cond* **)** *statement*
	$\mid$	**if (** *cond* **)** *statement* **else** *statement*
compound-statement	$\rightarrow$	{ *statement-list* }
expr	$\rightarrow$	**id + id**
cond	$\rightarrow$	**id** $<$ **id**

I used **bold** font for terminals, and *italics* for non-terminals. The terminal **id** matches any C identifier (e.g. a, b, c, and foo). The other non-terminals are exactly the corresponding strings, (e.g. **if** matches the string 'i' 'f', and so on). Show that this grammar is ambiguous. In particular, give two parsings for the program:

```
if(a < b)
  if(b < c) foo = a+b;
  else foo = b+c;
```