CS471, Programming Language Structure I

Sample answer to Assignment 3, Fall 2002

The syntax is altered with the addition of a multiple assignment statement. This can be done recursively:

```
          S ::= Ilist = Elist | ...
          Ilist ::= I | I,Ilist
          Elist ::= E | E,Elist
```

There is no way in BNF to specify an equal number of Is and Es.

There are no new domains needed, except that there must be operations for processing a list. These are head and tail:

head(I,Ilist) = I
tail(I,Ilist) = Ilist

the new valuation function for the multiple assignment is then:

M⟦Ilist = Elist⟧ s = if tail(Ilist) is empty then M⟦head(Ilist) = head(Elist)⟧ s
else M⟦tail(Ilist) = tail(Elist)⟧ (M⟦head(Ilist) = head(Elist)⟧ s)

Note that this executes the tails of the lists in the store returned by executing the head. Thus the assignment:

```
          x,y=4,x
```

will first change x to 4, and then change y to 4, the new value of x. It also assumes that there are an equal number of Is and Es in the two lists. To do “parallel” assignment, in which the store does not change for each assignment:

M⟦Ilist = Elist⟧ s = M⟦Ilist = Elist⟧ s s
M⟦Ilist = Elist⟧ s_0 s_1 = if Ilist is empty then s_1
else M⟦tail(Ilist) = tail(Elist)⟧ s_0 (M⟦head(Ilist) = head(Elist)⟧ s_0)

In this version, the original store, s_0 remains fixed, but the changes are accumulated in a second store, s_1.

Either version is a correct answer to the problem.