3a)

In the Sante Fe Trail, if the ant does not see a food in front of it, or to its side, it will move forward and check again. At location 116 and 136, the ant sees no food in front or to its side. In this case, the ant will choose to move forward. However, the correct choice is to actually move to the left. Once the ant moves forward at position 116 and 136, it will never go on the trail again.

b)

Now the ant needs to take care of the problem seen at position 116 and 136.

The new algorithm will be as follows:

**START)** At each step, the ant first

1. LEFT and checks for food
   1. If yes: MOVE and program goes to **START**
   2. If no: MOVE
      1. if left, right or forward has food, (possibly turn and) MOVE there and program goes to **START**
      2. if not, MOVE
         1. if left, right or forward has food, (possibly turn and) MOVE there and program goes to **START**
         2. MOVE back to original position at this step (**step 1**). This can be done by turning LEFT twice and MOVE twice.
2. RIGHT and checks for food
   1. If yes: MOVE and program goes to **START**
   2. If no: MOVE
      1. if left, right or forward has food, (possibly turn and) MOVE there and program goes to **START**
      2. if not, MOVE
         1. if left, right or forward has food, (possibly turn and) MOVE there and program goes to **START**
         2. MOVE back to original position at this step (**step 1**). This can be done by turning LEFT twice and MOVE twice.
3. MOVE and checks for food
   1. If yes: MOVE and program goes to **START**
   2. If no: MOVE
      1. if left, right or forward has food, (possibly turn and) MOVE there and program goes to **START**
      2. if not, MOVE and program goes to **START**

The terminals are all still the same: MOVE, LEFT, RIGHT, since the ant only needs these terminals to execute the algorithm described above.

The functions include: IF-FOOD-AHEAD and PROGNm, since the ant only needs these terminals to execute the algorithm described above. However, since the tree is significantly deeper and more complex now, there will be many more functions.

With regards to which PROGNm we need, we note that PROGN2 is all we need for any types of PROGNm functions (for m >=2), because we can get any PROGNm function by chaining together PROGN2. For example:

is equivalent to: