Morgan Baccus  
Homework #0  
CptS 317

**1.** Program the robot such that the set of all its observable behaviors are exactly L1:

I wrote pseudo code like was shown in lectures. The word must have three a’s in order to satisfies the criteria of L1. The variable otherCriteria indicates that once there have been 3 a’s printed, the word could end but does not have to. The comment “M shows flower” shows when the machine M will execute its function that generates the events. I assumed we did not need to write a function that generates the events, only when to stop generating the events to create a word that will fit in L1.

int aCounter = 0;  
while(keepGoing == true)  
{  
 //M shows flower  
 if(flower == red)  
 aCounter++;  
 if(aCounter >= 3)  
 keepGoing = false;  
 else if(otherCriteria)  
 keepGoing = true;  
}

**2.** Program the robot such that the set of all its observable behaviors are exactly L2:

If the number of flowers shown is greater than what an integer variable can hold and the function that shows the flowers was not written to prevent this issue, then memory problems could occur. Since there is no limit to how long the word can be and the only condition is that the number of a’s must be equal the number of b’s, the program could run infinitely if the machine never shows enough a’s.

int aCounter = 0;  
int bCounter = 0;  
while(keepGoing == true)  
{  
 //M shows flower  
 if(flower == red)  
 aCounter++;  
 if(flower == blue)  
 bCounter++;

if(aCounter == bCounter)  
 keepGoing = false;  
}

**3.** Argue intuitively why you only need a fixed and finite amount memory for the program in 1. while you have to use an unbounded amount of memory for the program in 2.:

The only variable that we must keep track of in 1. is aCounter. We are only concerned with aCounter until it reaches 3, then the conditions have been met and we no longer need the variable. This means that aCounter will never go over the bounds of what an integer variable can hold and thus, the memory is finite. Our counter variables in 2. might need to be infinitely large depending on what flowers the robot shows. This would create problems in the code and therefore cannot be represented in code, so we can’t construct L2 using Finite Automata.

**4.** Suggest a way to measure the complexity of a word:

The process would be similar to how we evaluate the complexity of a language. However, instead of looking at the criteria for creating a word, we would need to look at the criteria for that word to be in a language. The loops to create and check if a word is in a language are similar, with the exception that the loop to create versus check a word are slightly different. We are able to use the same logic as memory usage to evaluate if a word is complex and if a language is complex.