cpt_s 350

Homework 7

11641327 Yu-Chieh Wang

2020/3/30

1. LTL: Where there is an infinite walk w-path on (a color graph)G such that the color sequence satisfies the following property?

    (1) An algorithm to decide wether w-path satisfies $\Box(Yellow \vee \Diamond Blue)$ or not.

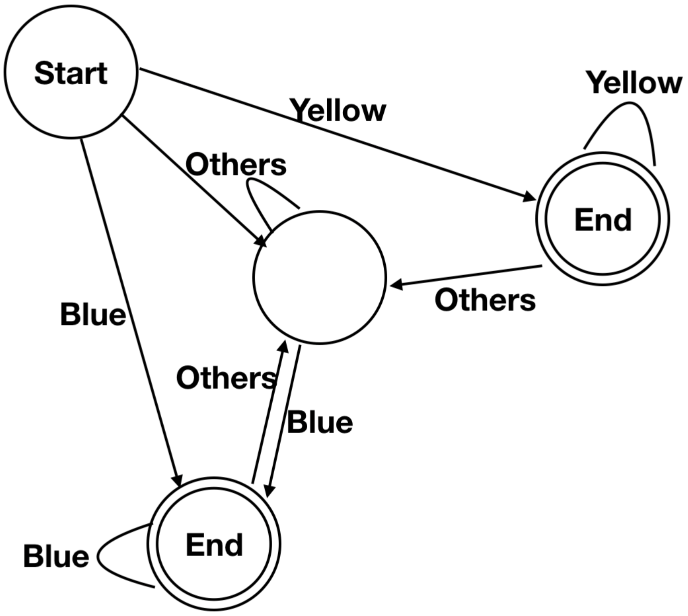
    Consider to all situations of the property:

    A. All yellow's.

    B. (others)…, blue,…(others)…, blue.

    C. (others)…, blue, yellow, (all yellow).

    We get a FA like this:



Run the FA on a path, if it stops on any end point, the FA returns True.

Else, it returns False.

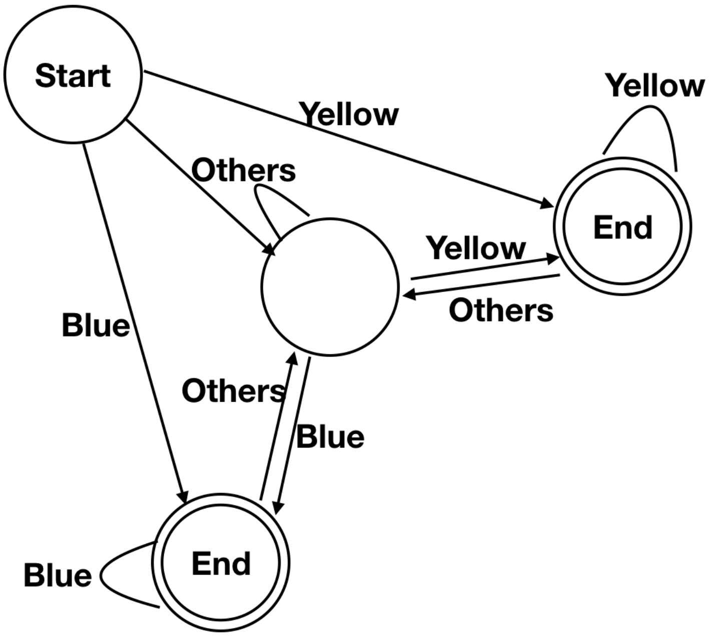
(2) An algorithm to decide wether w-path satisfies $\Box\Diamond(Yellow \vee \Diamond Blue)$ or not.

Consider to all situations of the property:

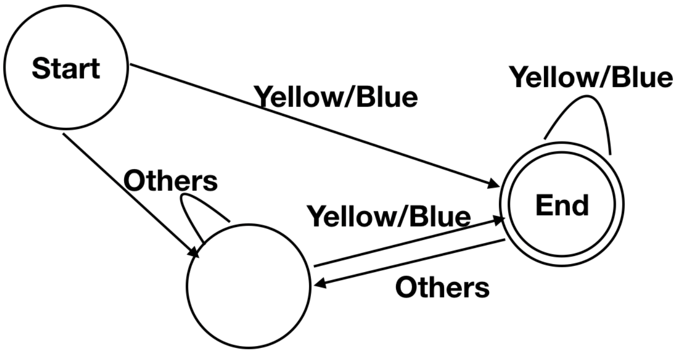
A. (others)…, blue,…(others)…, blue.

B. (others)…, yellow,…(others)…, yellow.

We get a FA like this:



Then, we found that the graph is symmetrical, so we cam simplify it as follow:



Run the FA on a path, if it stops on end point, the FA returns True.

Else, it returns False.

2. An algorithm to decide wether w-path passes infinite red nodes and finite blue nodes.

   In other to get a path that passes infinite red nodes, the most important thing is that finding at least one loop which has more than one red node but no blue node. After finding the loop, we can make a path run the loop again and again, so that the number of red nodes passing will approach infinity; or even run forever. Next, in addition to find loops with red nodes, we need to check and make sure that the loops don't include any blue node to ensure that the number of passing blue nodes is limited.

3. An algorithm to decide wether w-path passes the same number of red and blue nodes.

   First, assuming there are two infinitely long stacks (blue and red), so we won't be hampered by the variable memory limit. Second, design a method to count the number of red and blue nodes:

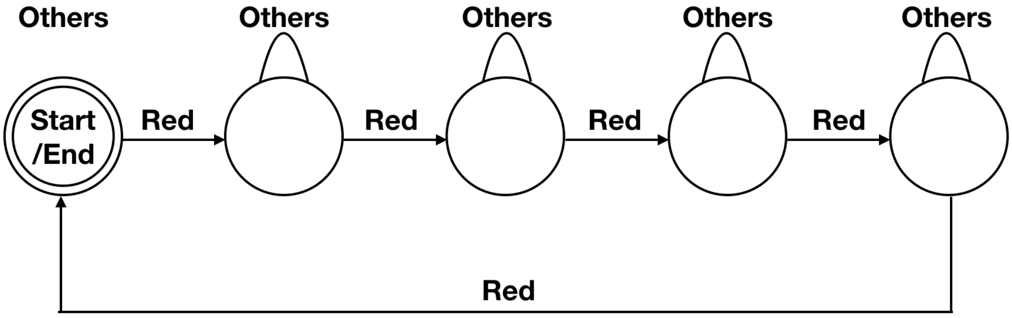
```
If pass a red node{
   check if the blue stack is empty
   If so, push 1 to the red stack.
   Else, pop 1 from the blue stack.
}
If pass a blue node{
   check if the red stack is empty
   If so, push 1 to the blue stack.
   Else, pop 1 from the red stack.
}
```

   Finally, check if both stacks are empty. If so, return True(it is a good path).

   Else, return False.

4. An algorithm to decide wether w-path passes the number of red nodes is a multiple of five.

   We use a FA to solve the problem as follow:



   Run the FA on a path, if it stops at the end point, it returns True; else, it return False. The start and end point are at the same point because a path passes 0 red node can be accepted by the FA, too.