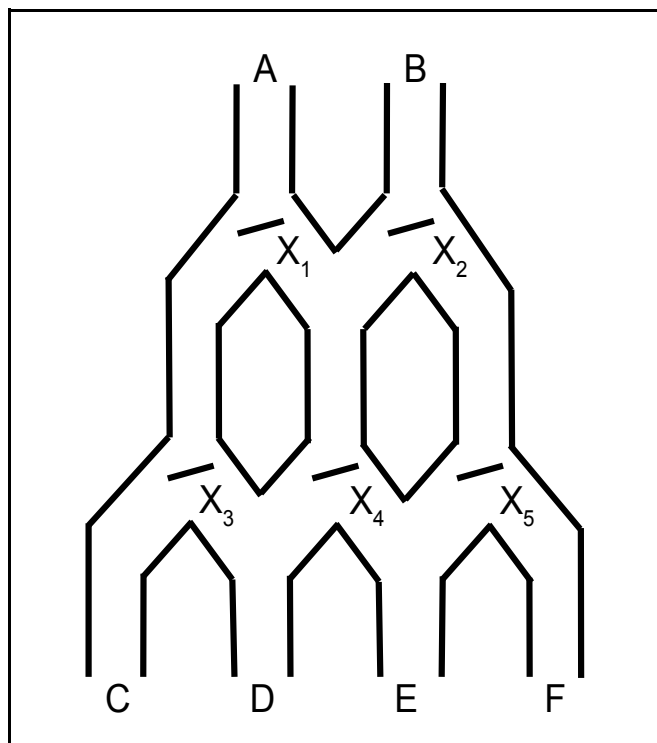


Due 2/25

Construct a DFA using JFLAP that simulates the following game (modified problem 2.3 from Hopcroft/Ullman). We drop a marble down the chute labeled A or B. If the last marble that we drop down the chutes comes out one of the chutes labeled C or F, then we loose. If the last marble that we drop down the chutes comes out one of the chutes labeled D or E, then we win.

To simulate this we have an input alphabet of $\{0, 1\}$. An input symbol of '0' simulates dropping a marble down chute A and an input symbol of '1' simulates dropping a marble down chute B. Each time a marble touches a gate it toggles the gate to the other direction and the marble goes in the direction that the gate was set to prior to being toggled. Initially all of the gates are configured to the left (as shown in the diagram). As an example, if all of the gates are positioned to the left, then dropping a marble down chute A results in the marble exiting gate C and toggling gates X_1 and X_3 to the right. If the gates are configured as X_1 and X_2 to the right, and X_3 , X_4 , and X_5 to the left, then dropping a marble down chute A results in the marble exiting chute D and toggling gates X_1 to the left and X_4 to the right.



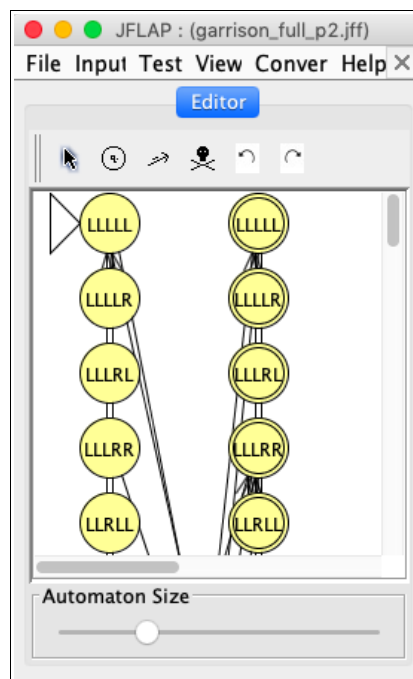
You are to construct a DFA using JFLAP that simulates this marble game, with the DFA accepting those strings that result in winning and rejecting those strings that result in loosing.

For your initial submission, name your file group_x_y_p2.jff, where x is your group number and y is your section (a or b). Submit your group submission, copying everyone from your group on the e-mail, within 2 hours after your class ends (8pm for the 4:25pm –

Due 2/25

5:50pm section and 9:35pm for the 6:00pm – 7:25pm section). Once I've posted the group grades, if your grade is less than 100, you can submit individual updates of the group submission (use the normal naming convention) until midnight one week after the grades are posted. Your final grade will be the larger of the group submission, average of the group submission and your individual submission, and 80% of your individual submission. At the bottom of this assignment are 6 of my test strings, three should be accepted and three rejected. If you submit during class and do not get the correct answer for the six strings, I will tell you the first string you get the incorrect answer for. You can get feedback up to two times during class.

Below is a screen shot of part of my finite automata. I've labeled the states with the paddle configuration, in an attempt to make the finite automata easier to read. This version has all 64 potential states.



E-mail the JFLAP file to me (david.garrison@binghamton.edu) by the specified time on the date due. For individual submissions the filename is to be your last name followed by "_p2.jff" (my filename would be "garrison_p2.jff"). I've already specified the group submission filename earlier in these instructions. The subject of your e-mail is to be "CS 373 program 2".

Grading will be done by testing 12 strings with your finite automata, half winning sequences and half losing sequences. Each test string is worth 8.33% of your grade for the program. My test strings will all be around 100 symbols in length.

Due 2/25

The DFA is quite simple to understand, but is tedious to construct, since there are potentially 64 states (32 non accept states and 32 accept states).

I made two versions of the DFA, one with all 64 states, and one with only those states that can be reached from the start state (22 states, 15 accept states and 7 non accept states).

Assuming that I got my version of the program correct, the following may be helpful.

Trace of configurations for various inputs (values for $X_1X_2X_3X_4X_5$).

1111
LLLLL->LRLRL->LLLRR->LRLLR->LLLLL

0000
LLLLL->RLRLL->LLRRL->RLLRL->LLLLL

10101010
LLLLL->LRLRL->RRRRL->RLRRR->LLRLR->LRRRR->RRLRR->RLLRL->LLLLL

01010101
LLLLL->RLRLL->RRRRL->LRRL->LLRLR->RLLLR->RRLRR->LRLLR->LLLLL

Sample strings (half accepted and half rejected)

1111111110
0111001100
0001011010
0110100110
1000001010
1001011101
0011101011
0101000101
1100010001
1010011101

Six of my test strings (half accepted and half rejected).

011111100010101001011010101111001110001011001010010011001011011101000010100000101001011110110011010
1000000010101000011100101101101100011110010001101111001110011110011110101011010011001001011010011010
10100010110000011110111010100101101111110011001010111011011010010001001110111111010101011100010110
1111100001001111111101110100100010011010010011000010100001011000001100101000000010101010001111100100
1111110001101111011111001110001110110101110011001111101100001010110101000101100011011111001100001110
0110000100111100001110111010001000110110011111100010000001011110001000011111011101101101010001000

