Instructions on Problem B2

Input (2 files)

(1) DFA.txt

DFA.txt specifies the transition table of the DFA. Suppose the alphabet has two letters, then each row of DFA.txt consists of two numbers, corresponding to a row in the transition table. The last row (starting with F) specifies all the final states.

For example, DFA.txt for dd in the homework is

2 3

2 4

2 3

...

7 6

F 5 9

All numbers are separated by space. In general, suppose the alphabet has n letters, each row in DFA.txt would have n numbers.

(2) Query.txt

Query.txt specifies the query pairs of states. Each row specifies a pair. If we want to test

pairs (1,6) and (1,7). Query.txt is

1 6

1 7

Output (2 files)

(1) Partition.txt

You program should generate Partition.txt in the same directory. The file shows the partition of the forward closure. Each block will be a list of states separated by a space, and distinct blocks are separated by semicolons.

For example, if the partition is {{1, 3, 7}, {2, 4, 5}, {6, 8, 9}}, then your Partition.txt should output:

1 3 7; 2 4 5; 6 8 9

(2) Test.txt

Your program should also generate Test.txt in the same directory. This is the test result of pairs input in Query.txt. If the pair (p,q) is a pair of equivalent states, print G. Otherwise, print out the bad pair that caused the algorithm to stop. That pair may not be the original pair.

For example, given the input in Query.txt, suppose (1,6) is a pair of equivalent states and (1,7) is not, while the bad pair caused the algorithm to stop is (3,4). Test.txt should be

G

3 4

There is no instruction for the extra credit. Includes some screen shots in your written homework, and submit the code.

Test your program using different inputs.

During submission, zip all your source code, input files and output file and send it to one of the TA's before the deadline.