CS154

Name: Date:

Reducing of the Number of Sates in Finite Automata

Minimize the states in the dfa depicted in the following diagram. Use the *algorithm* given below.

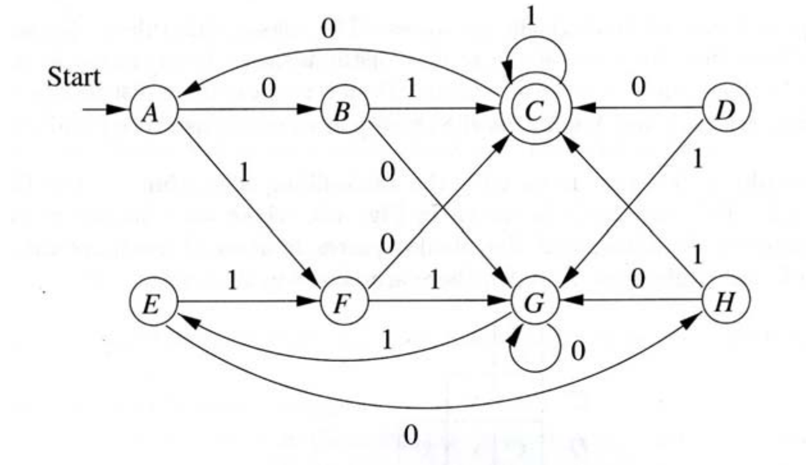

Table-filling algorithm is a recursive discovery of distinguishable pairs in a DFA

$$M = (Q, \Sigma, \delta, q_o, F)$$

We say that states *p* and *q* are *equivalent* if:

- For all input string w, $\hat{\delta}(p, w)$ is an accepting state *iff* $\hat{\delta}(q, w)$ is an accepting state.

If two states are *not* equivalent, then we say they are *distinguishable*. That is, that *p* is distinguishable from state *q* if there is at least one string *w* such that one of $\hat{\delta}(p, w)$ and $\hat{\delta}(q, w)$ is accepting, and the other is not accepting.

BASIS: if *p* is an accepting state and *q* is nonaccepting, then the pair $\{p, q\}$ is distinguishable

INDUCTION: Let *p* and *q* be states such that for some input symbol *a*, $r = \delta(p, a)$ and $s = \delta(q, a)$ are a pair of states known to be distinguishable. Then $\{p, q\}$ is a pair of distinguishable states.

a) Execute the *table-filling algorithm* on the given DFA, fill-in the table shown below. Fill-in *x* to indicate pairs of distinguishable states, and *blank* squares to indicate those pairs have been found equivalent. Initially, there are no *x's* in the table.

b) Minimize the DFA using the following algorithm:
   1. First, eliminate any state that cannot be reached from the start state.
   2. Then, partition the remaining states into blocks, so that all the states in the same block are equivalent.

B							
C							
D							
E							
F							
G							
H							
	A	*B*	*C*	*D*	*E*	*F*	*G*

a) Table of the sate inequivalences

b) Minimum-state DFA equivalent to the given DFA