# DFA Minimization



Note these slides are not mine.

http://faculty.kutztown.edu/spiegel

"Minimal"?

- Minimal number of states.

"Unique"?

- A minimal DFA for a given language is unique up to renaming of states.

# DFA Minimization: Algorithm Idea

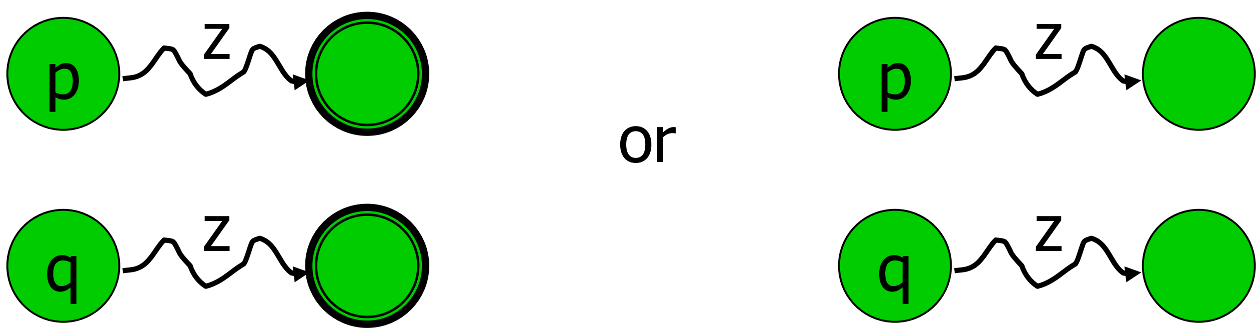


Equate & collapse states having same behavior.

Build equivalence relation on states:

$$p \equiv q \quad \leftrightarrow \quad (\forall z \in \Sigma^*,\ \hat{\delta}(p,z) \in F \ \leftrightarrow\ \hat{\delta}(q,z) \in F)$$

I.e., iff for every string z, one of the following is true:



# DFA Minimization: Algorithm



Build table to compare each unordered pair of distinct states p,q.

Each table entry has

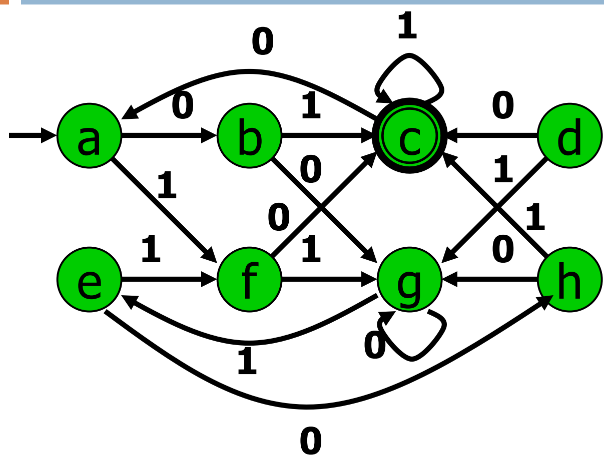
- a “mark” as to whether p & q are known to be not equivalent, and
- a list of entries, recording dependences: “If this entry is later marked, also mark these.”

# DFA Minimization: Algorithm



1. Initialize all entries as unmarked & with no dependences.
2. Mark all pairs of a final & nonfinal state.
3. For each unmarked pair p,q & input symbol a:
   1. Let $r=\delta(p,a)$, $s=\delta(q,a)$.
   2. If (r,s) unmarked, add (p,q) to (r,s)'s dependences,
   3. Otherwise mark (p,q), and recursively mark all dependences of newly-marked entries.
4. Merge unmarked pairs of states.
5. Delete inaccessible states.

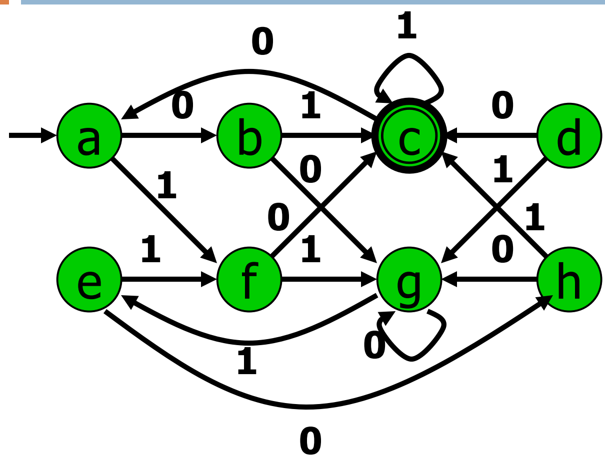
# DFA Minimization: Example



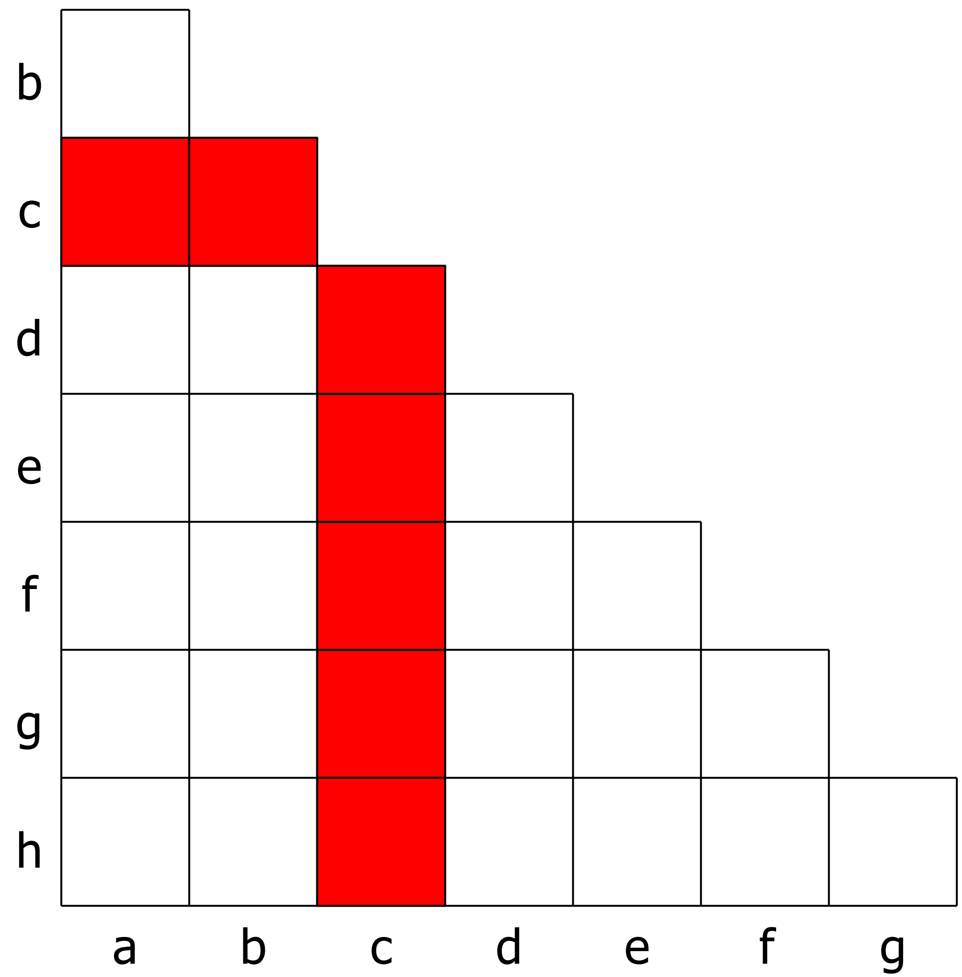

1. Initialize table entries:
   Unmarked, empty list

| | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| b | | | | | | | |
| c | | | | | | | |
| d | | | | | | | |
| e | | | | | | | |
| f | | | | | | | |
| g | | | | | | | |
| h | | | | | | | |

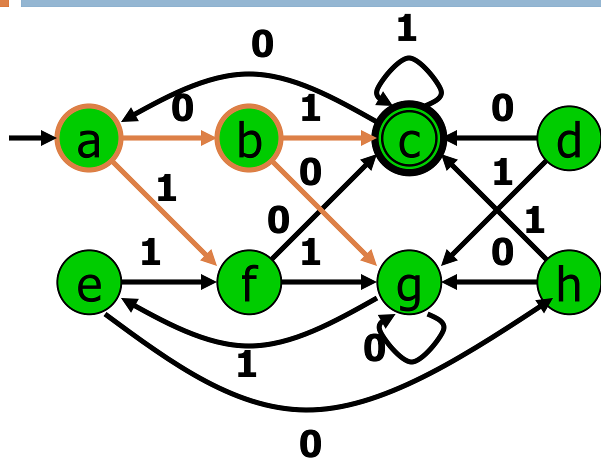
# DFA Minimization: Example



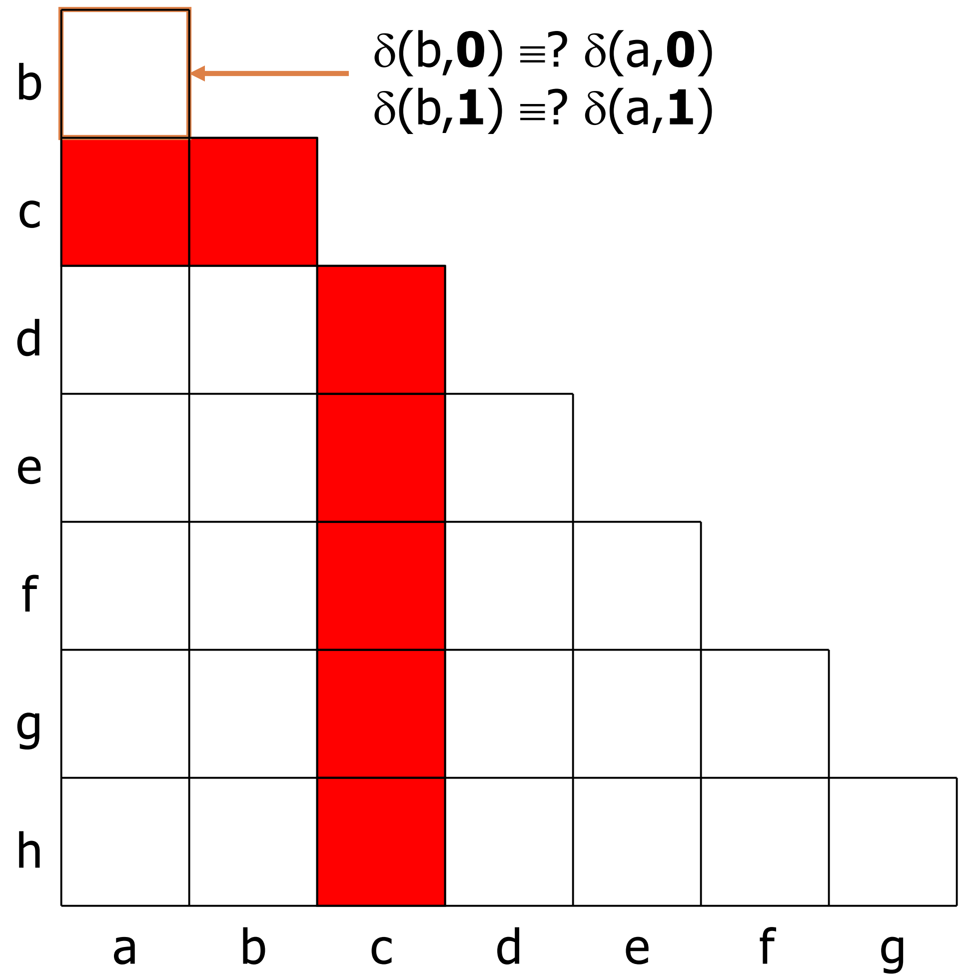

2. Mark pairs of final and non-final states



# DFA Minimization: Example



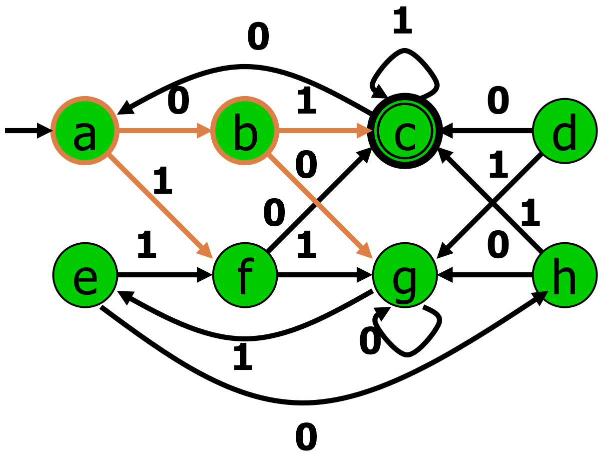

3. For each unmarked pair and symbol, …



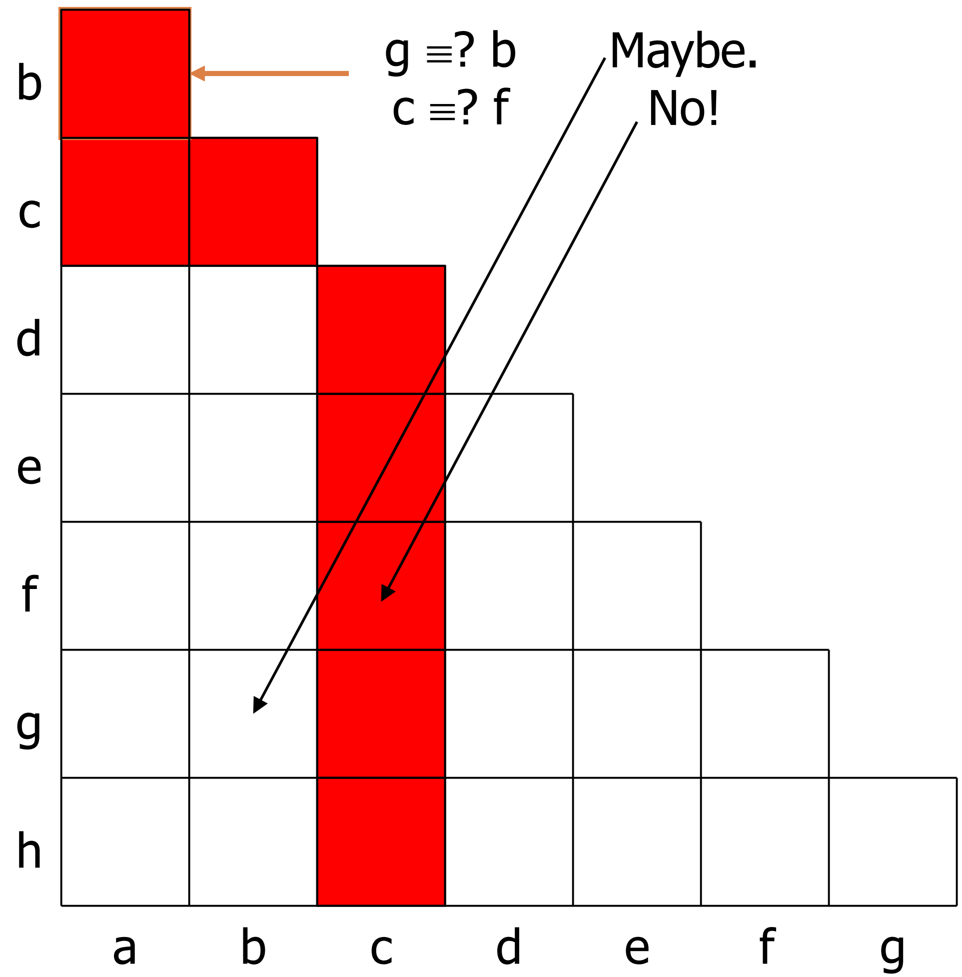

$\delta(b,\mathbf{0}) \equiv? \delta(a,\mathbf{0})$

$\delta(b,\mathbf{1}) \equiv? \delta(a,\mathbf{1})$

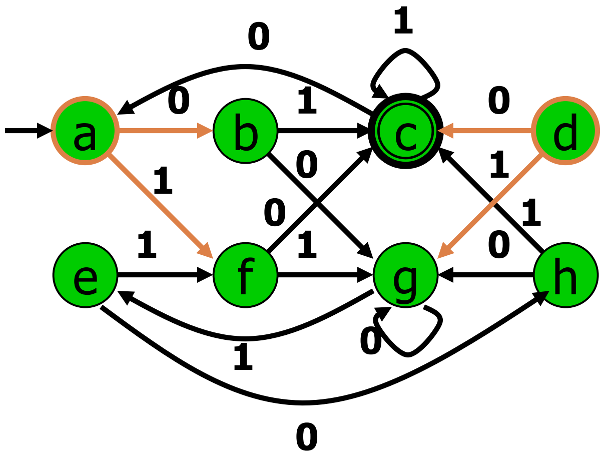
# DFA Minimization: Example



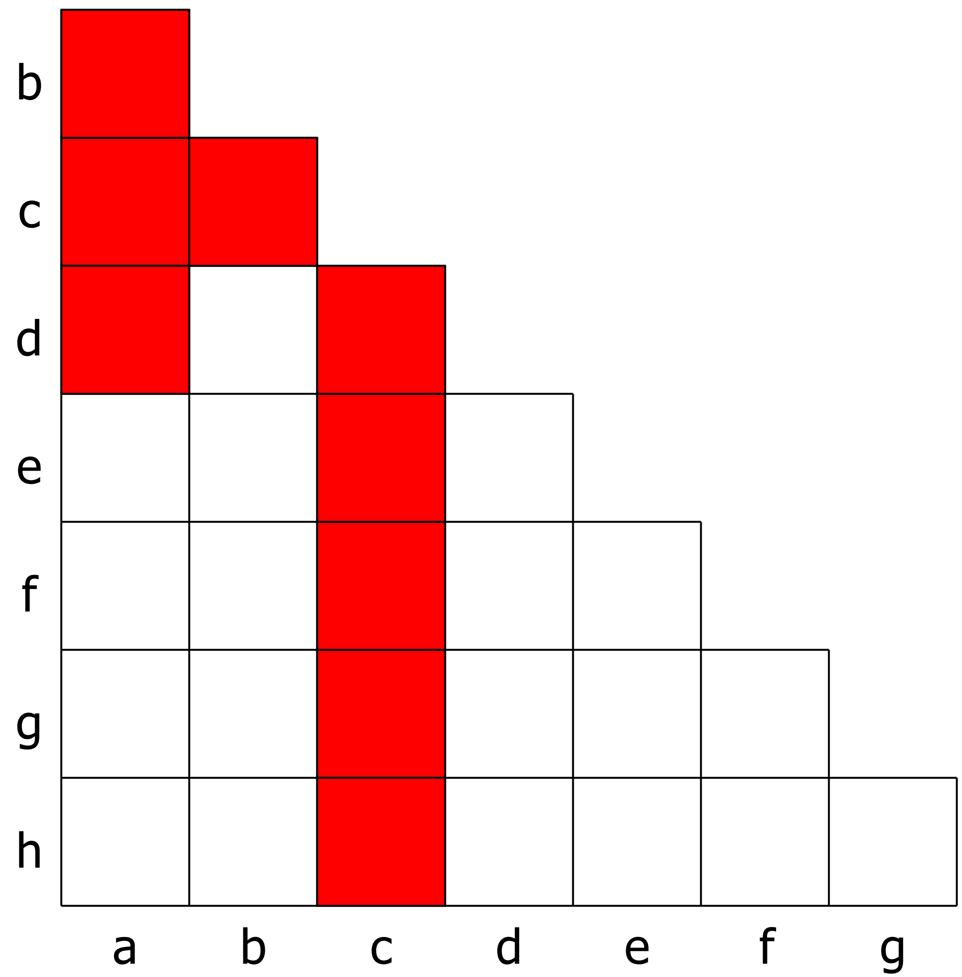

3. For each unmarked pair and symbol, …



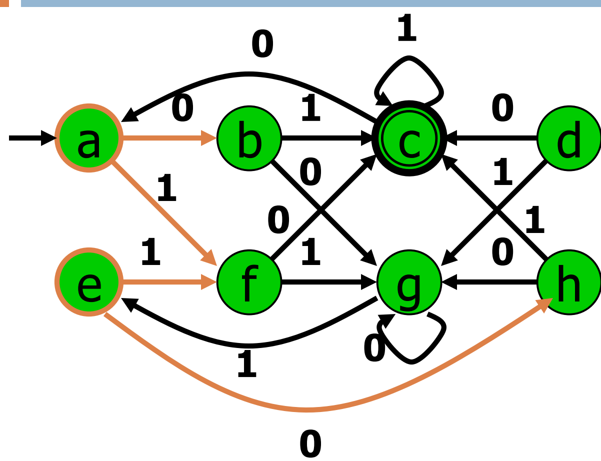
# DFA Minimization: Example



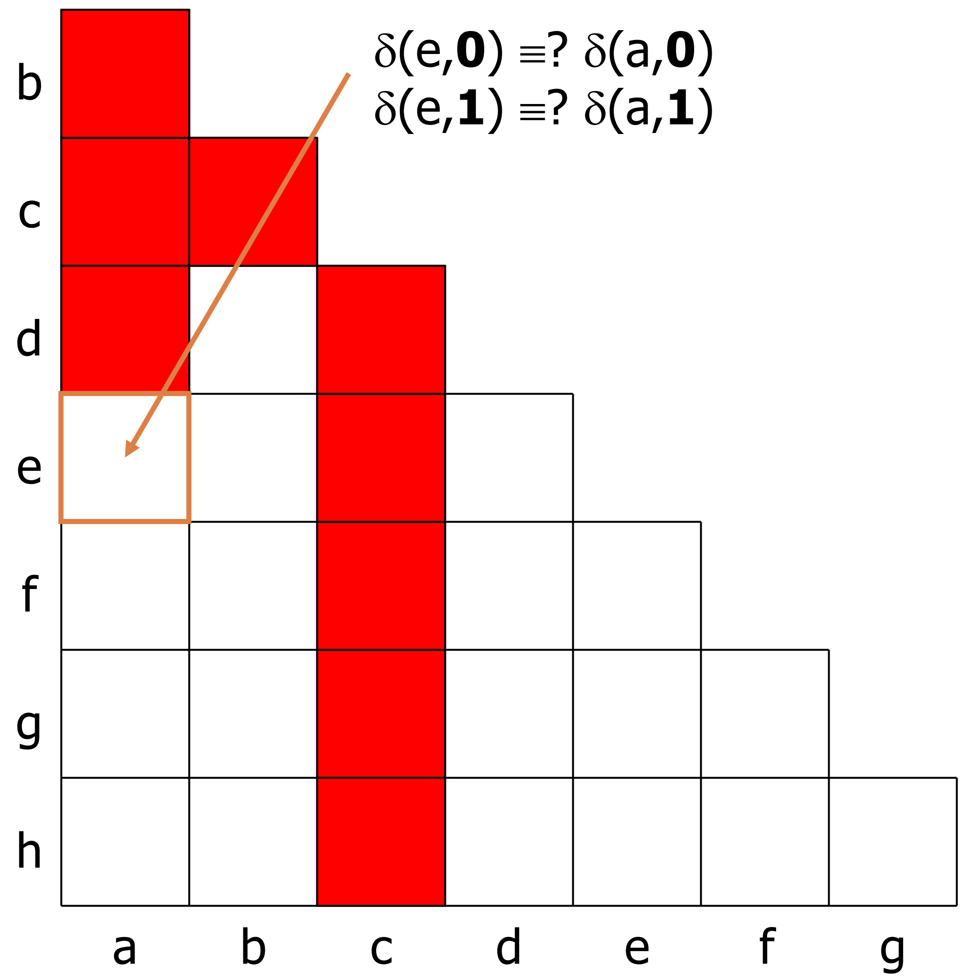

3. For each unmarked pair and symbol, …



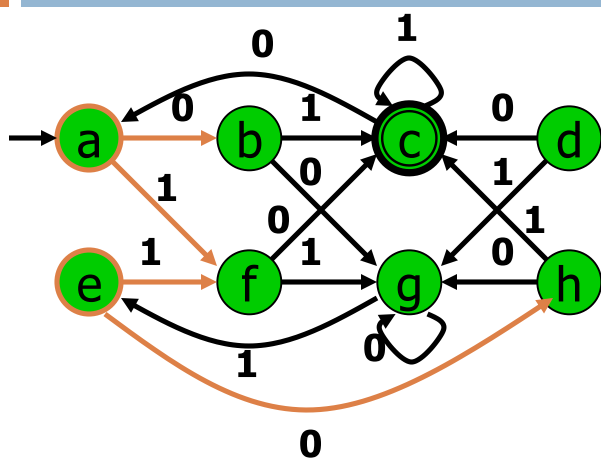
# DFA Minimization: Example



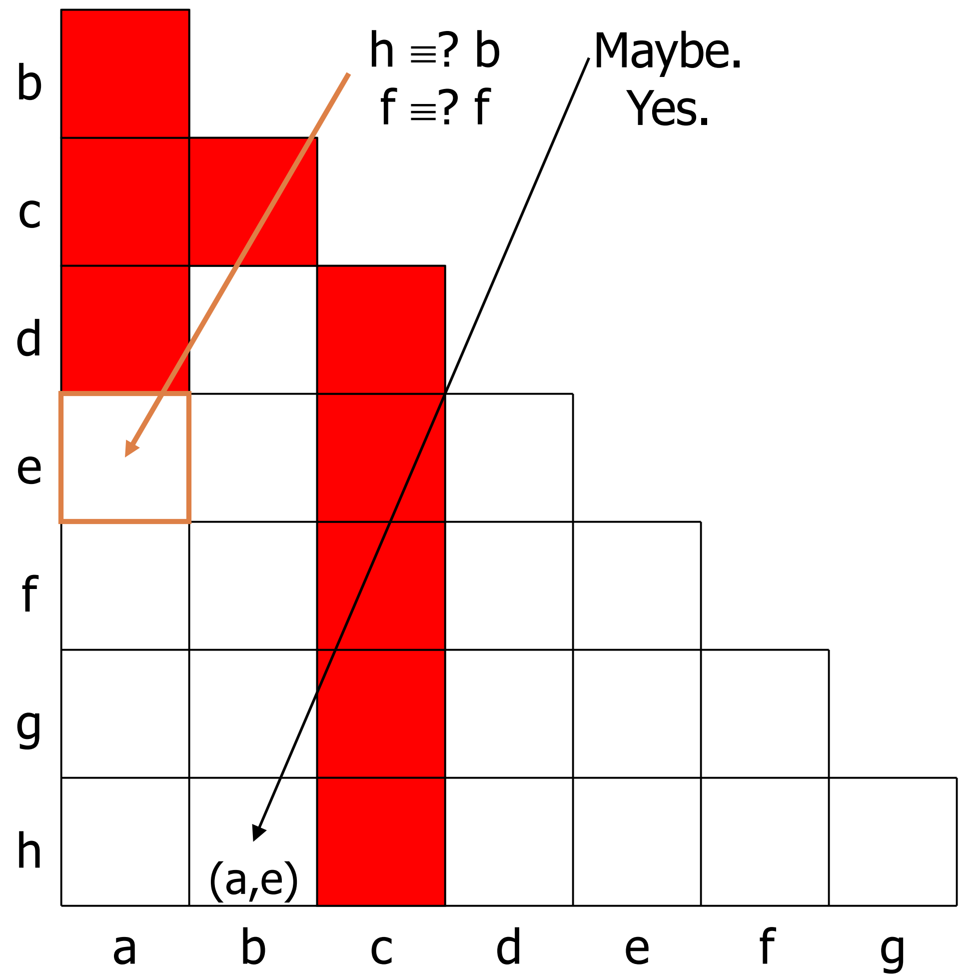

3. For each unmarked pair and symbol, …



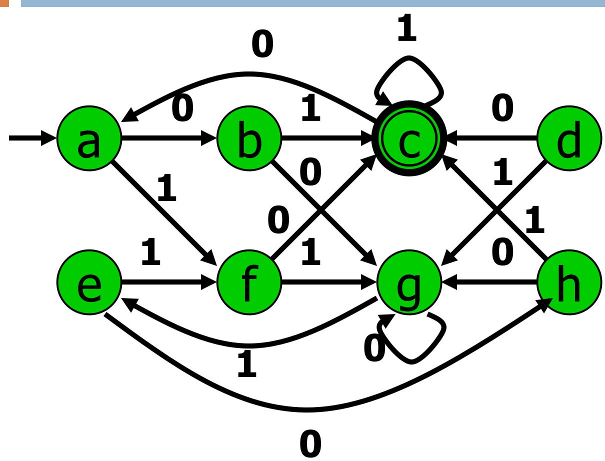
# DFA Minimization: Example




3. For each unmarked pair and symbol, …



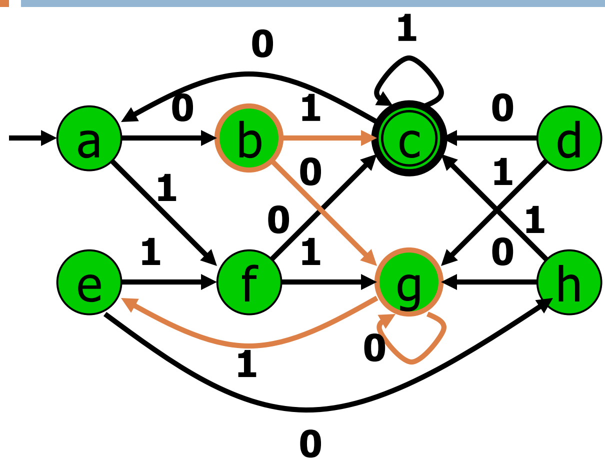
# DFA Minimization: Example



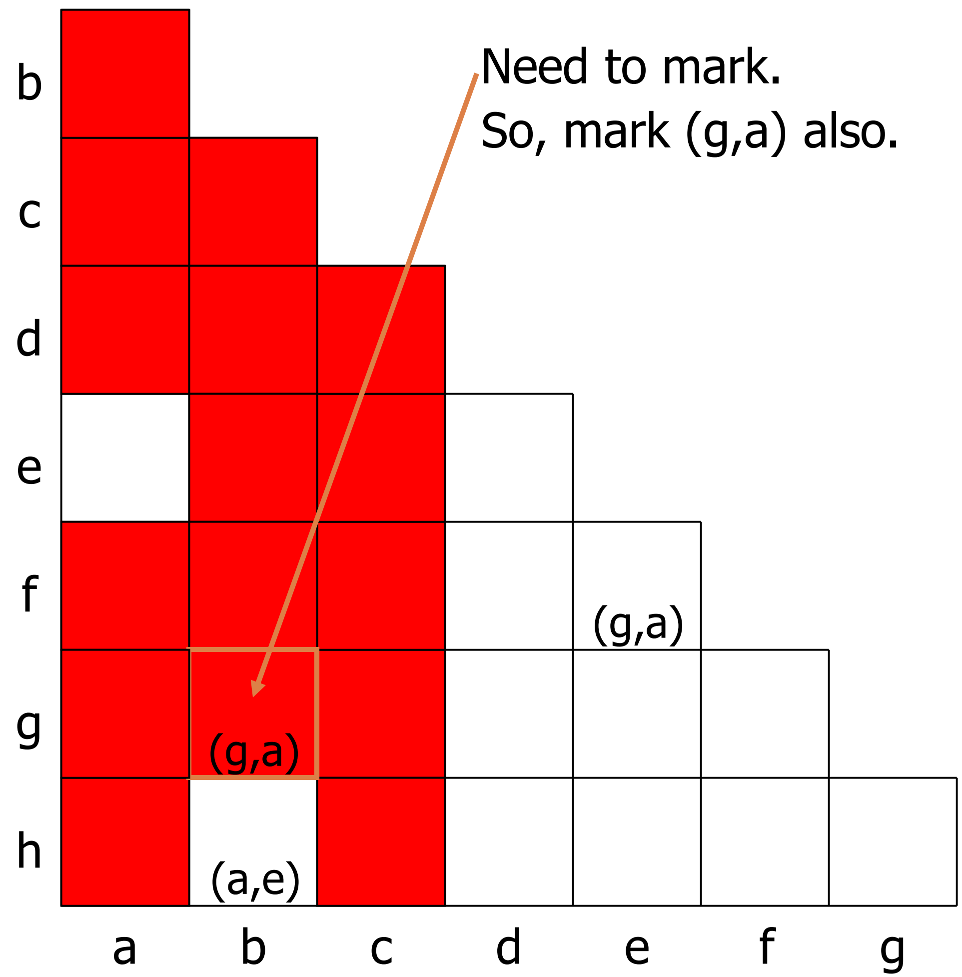

3. For each unmarked pair and symbol, …

| | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| b | | | | | | | |
| c | | | | | | | |
| d | | | | | | | |
| e | | | | | | | |
| f | | | | | (g,a) | | |
| g | | (g,a) | | | | | |
| h | | (a,e) | | | | | |

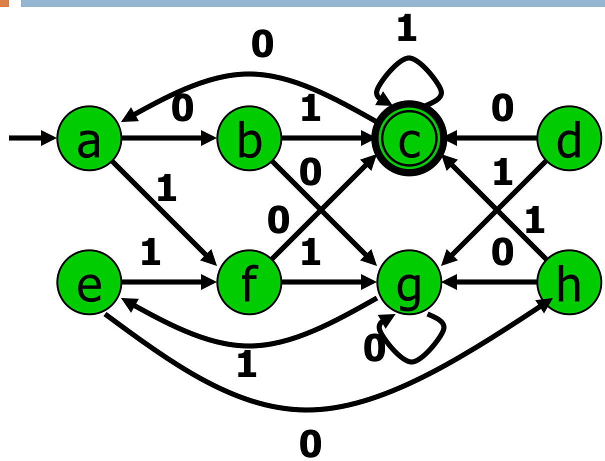
# DFA Minimization: Example



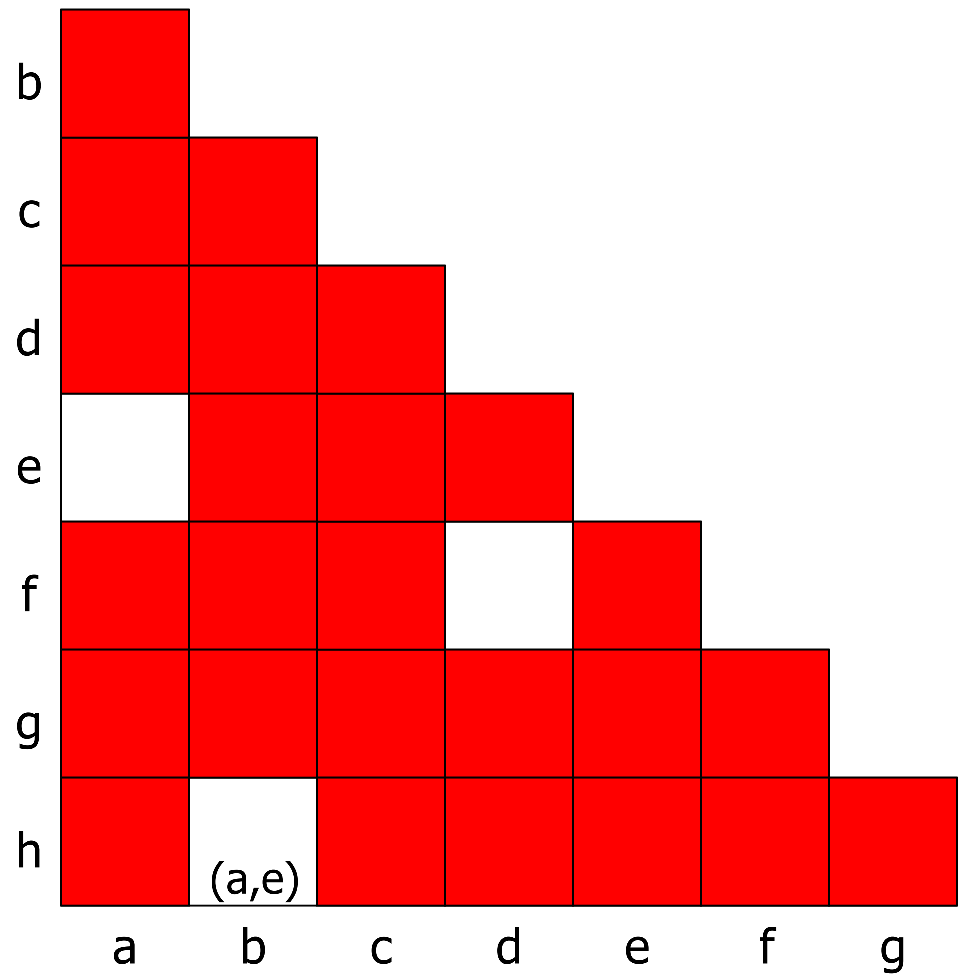

3. For each unmarked pair and symbol, …



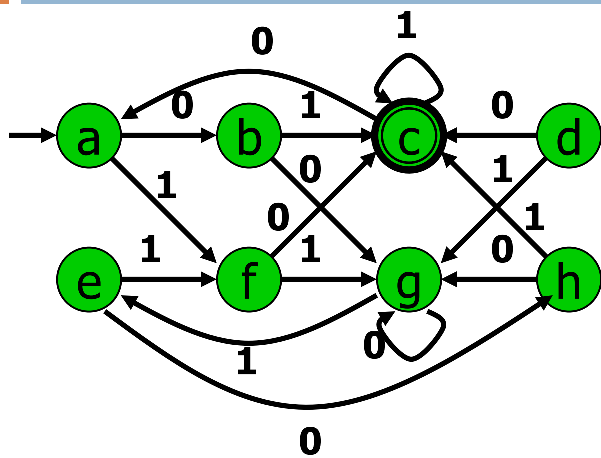
# DFA Minimization: Example




3. For each unmarked pair and symbol, …

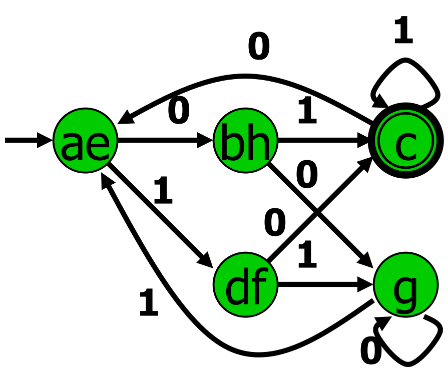


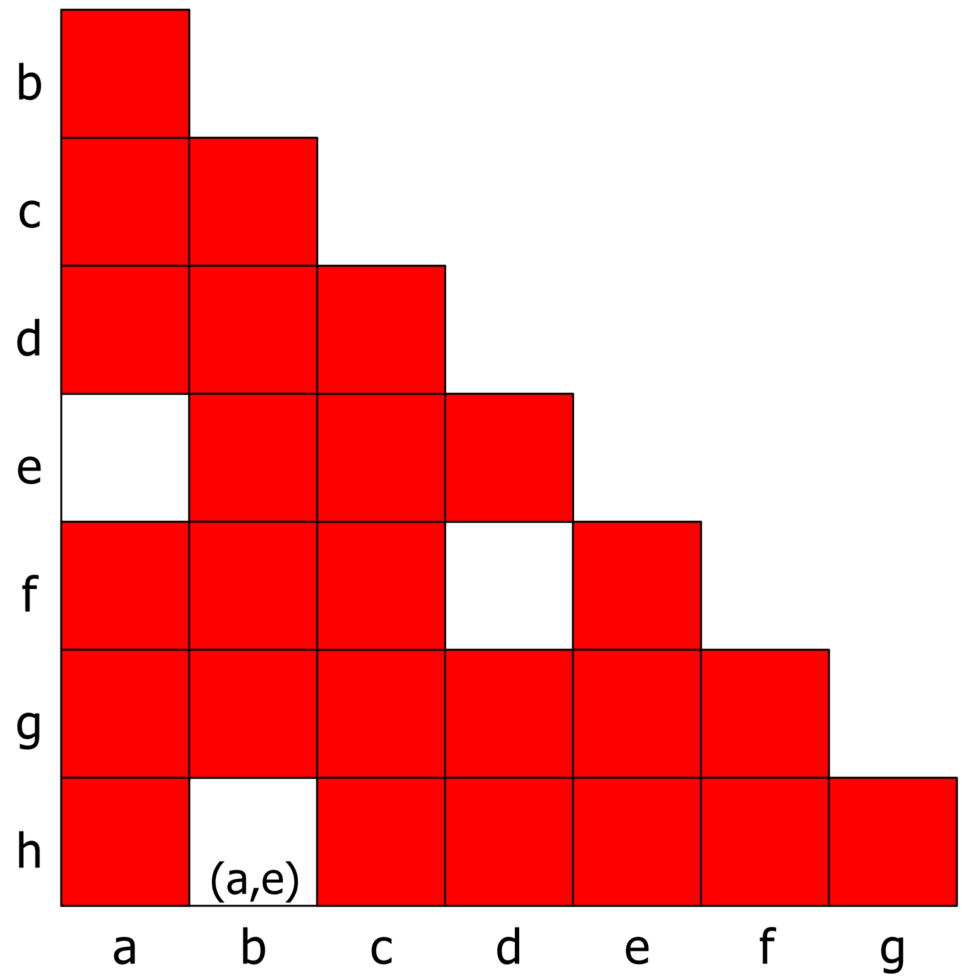
# DFA Minimization: Example



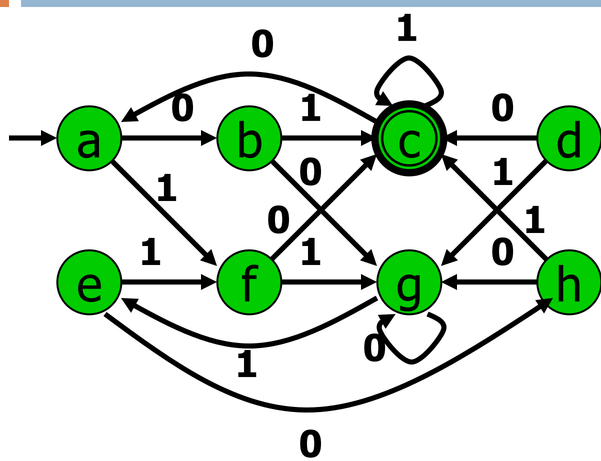

4. Merge unmarked pairs of states.

a ≡ e
b ≡ h
d ≡ f



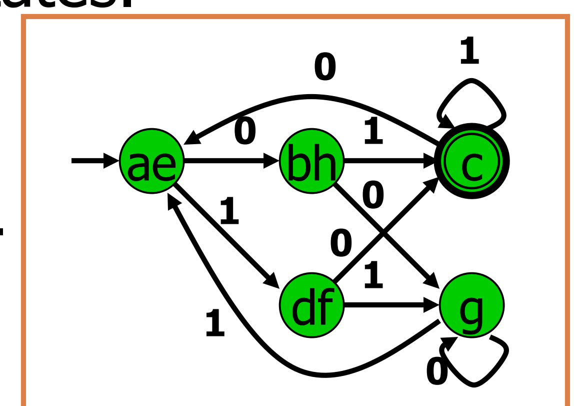



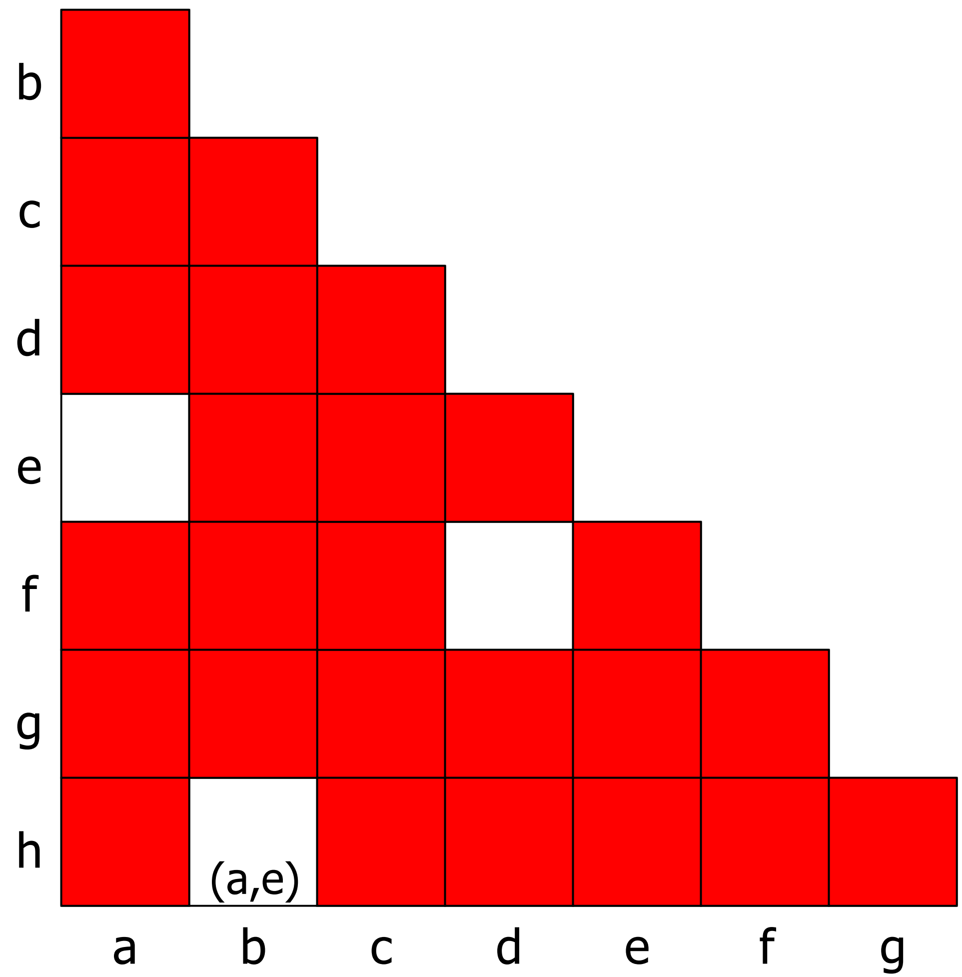
# DFA Minimization: Example




5. Delete unreachable states.

None.



# DFA Minimization: Notes



Order of selecting state pairs was arbitrary, but all orders give the same result.

Can delete unreachable states initially, instead.

This algorithm: $O(n^2)$ time; Huffman (1954), Moore (1956).

- Constant work per entry: initial mark test & possibly later chasing of its dependences.
- More efficient algorithms exist, e.g., Hopcroft (1971).

# DFA Minimization: Correctness



Why is new DFA no larger than old DFA?

Only removes states, never introduces new states.

Why is new DFA equivalent to old DFA?

Only identify states that have same behavior.

# What About NFA Minimization?



This algorithm does not find a unique minimal NFA.

Is there a (not necessarily unique) minimal NFA for a given language?

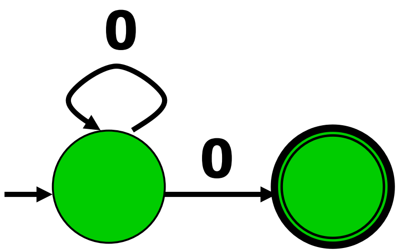
Of course.

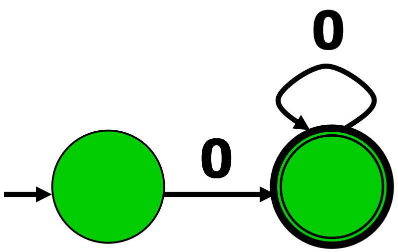
# NFA Minimization



In general, minimal NFA's not unique!

Example NFAs for $\mathbf{0}^+$:




Both minimal, but not isomorphic.