DFA Minimization: The Idea

"Minimal"?

Minimal number of states.

"Unique"?

Unique up to renaming of states.

I.e., has same shape. *Isomorphic*.

Will show algorithm & detailed example.

Will outline why algorithm is correct.

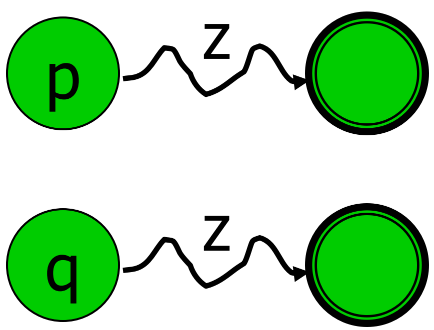
DFA Minimization: Algorithm Idea

Equate & collapse states having same behavior.

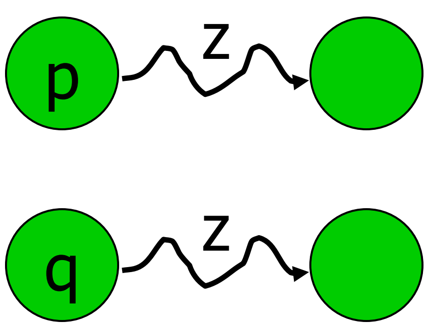
Build equivalence relation on states:

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

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



or



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.

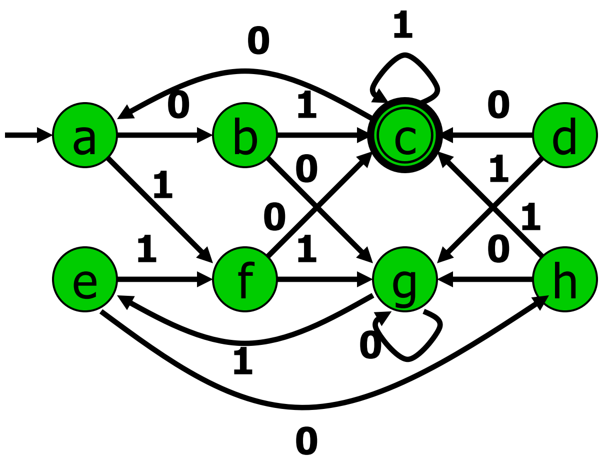
2. Mark all pairs of a final & nonfinal state.

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. Coalesce unmarked pairs of states.

5. Delete inaccessible states.

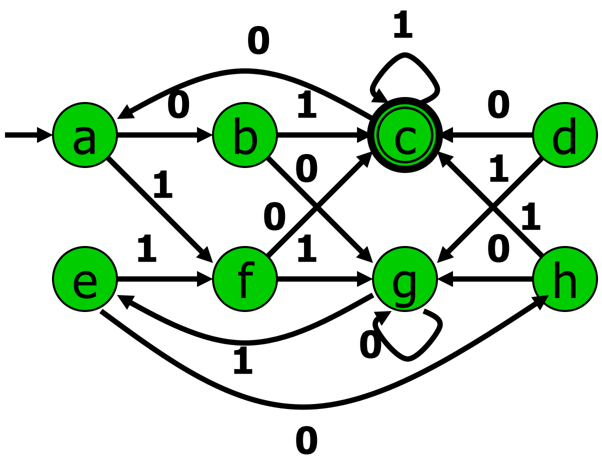
DFA Minimization: Example



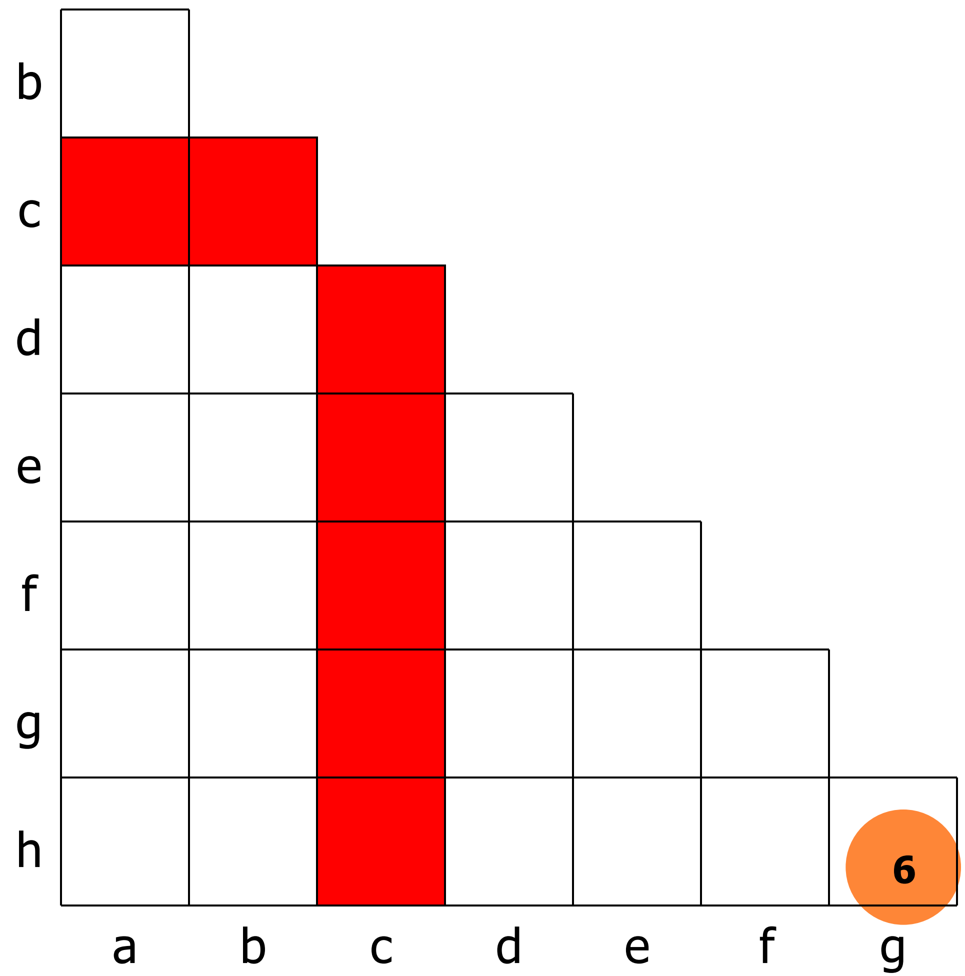
1. Initialize table entries:
   Unmarked, empty list

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

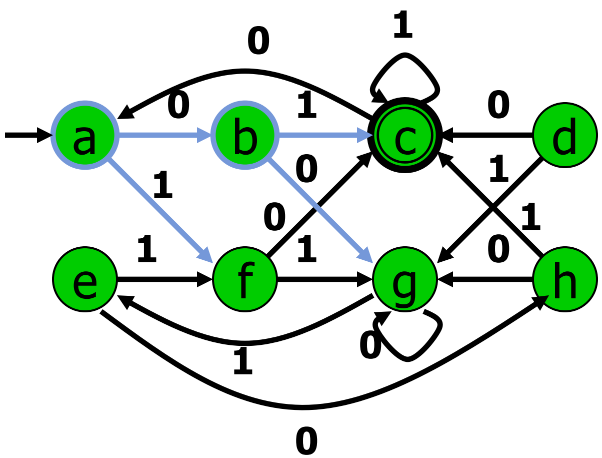
DFA Minimization: Example



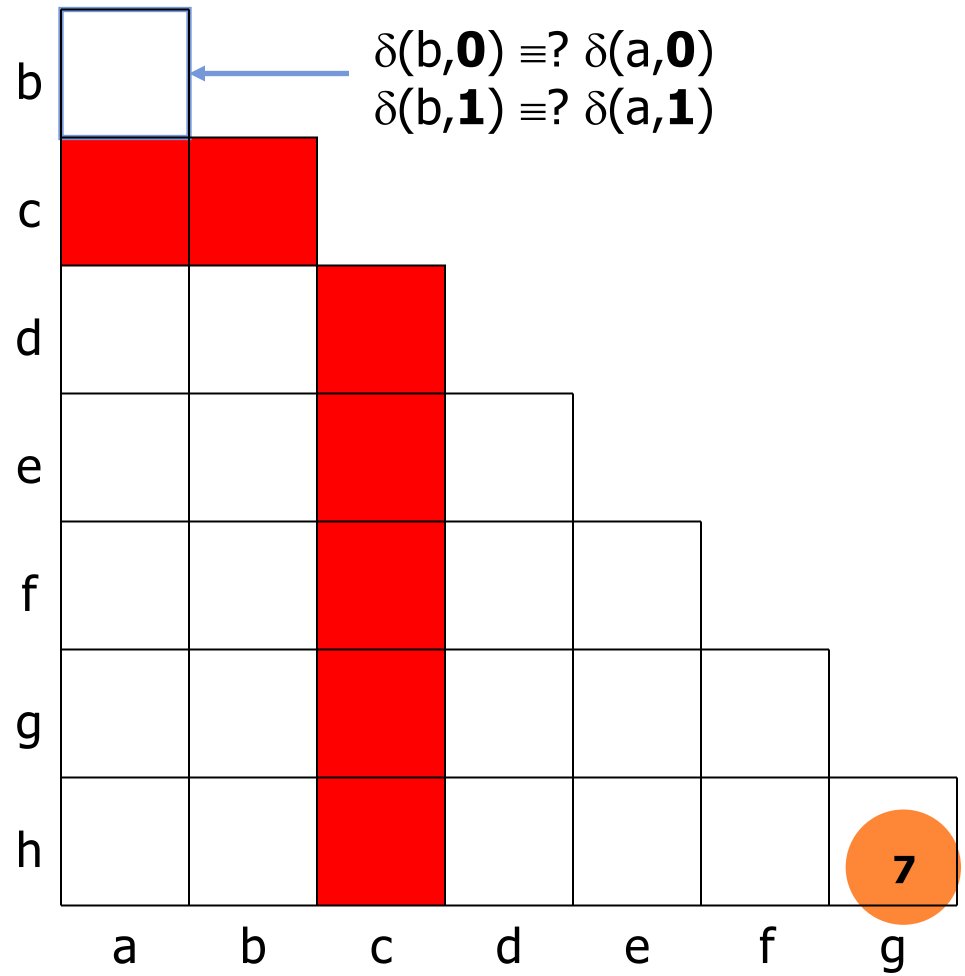
2. Mark pairs of final & nonfinal states



DFA Minimization: Example

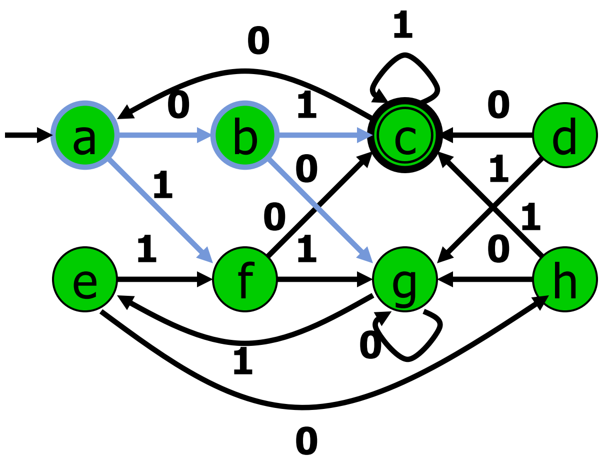


3. For each unmarked pair & symbol, …

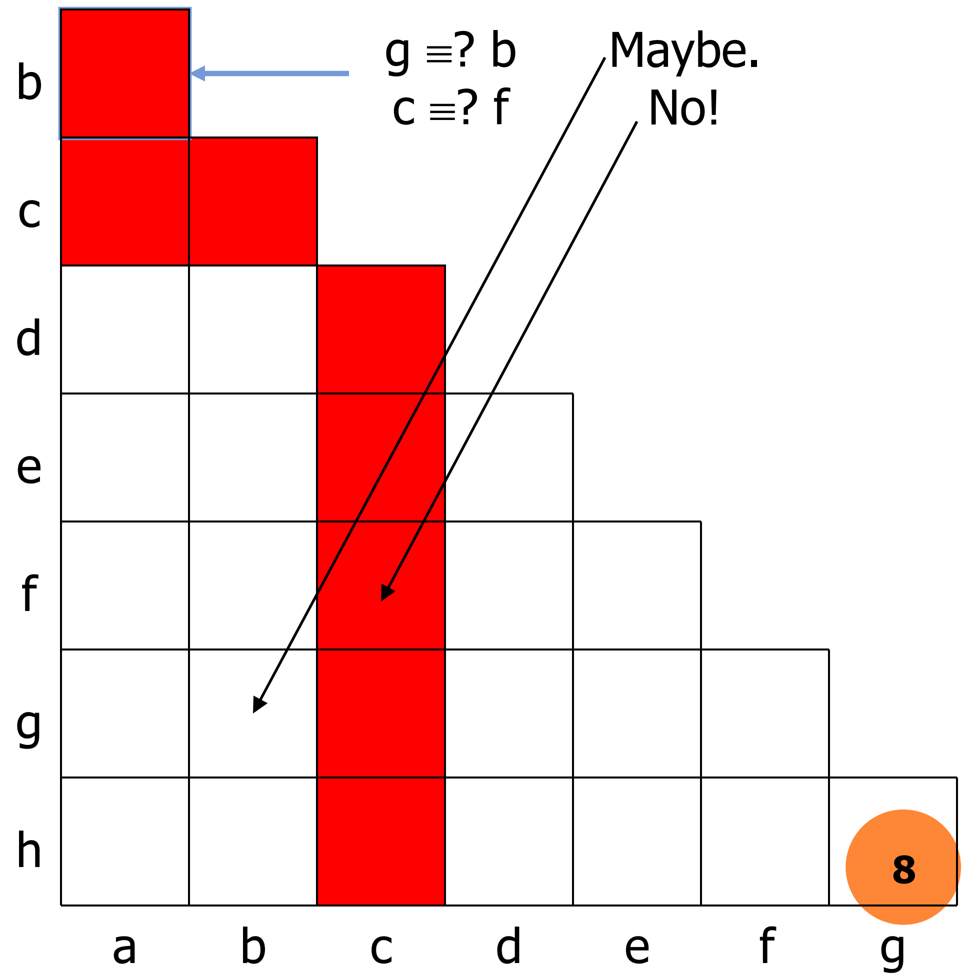


$\delta(b,\mathbf{0}) \equiv? \ \delta(a,\mathbf{0})$
$\delta(b,\mathbf{1}) \equiv? \ \delta(a,\mathbf{1})$

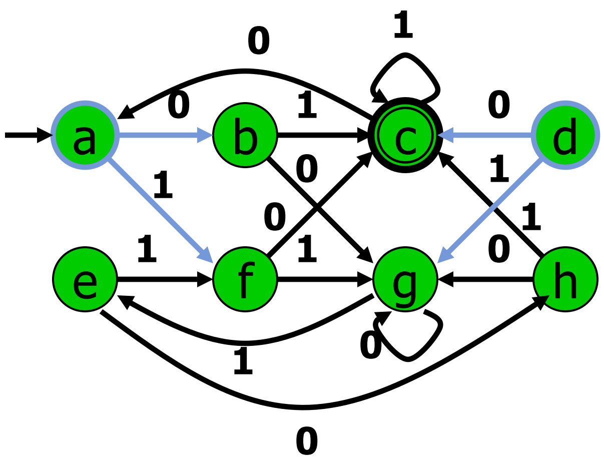
DFA Minimization: Example

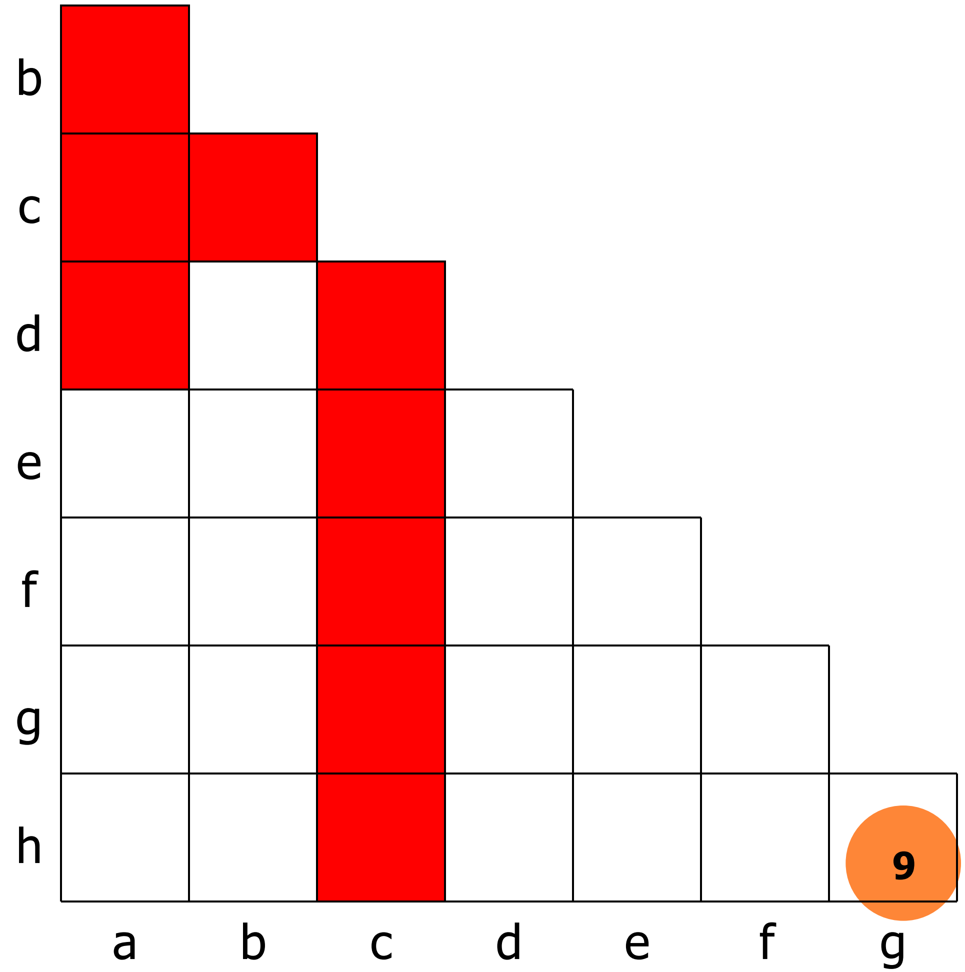


3. For each unmarked pair & symbol, …



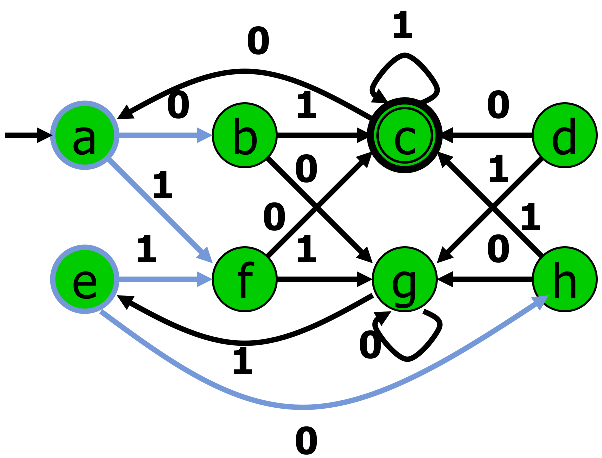
DFA Minimization: Example

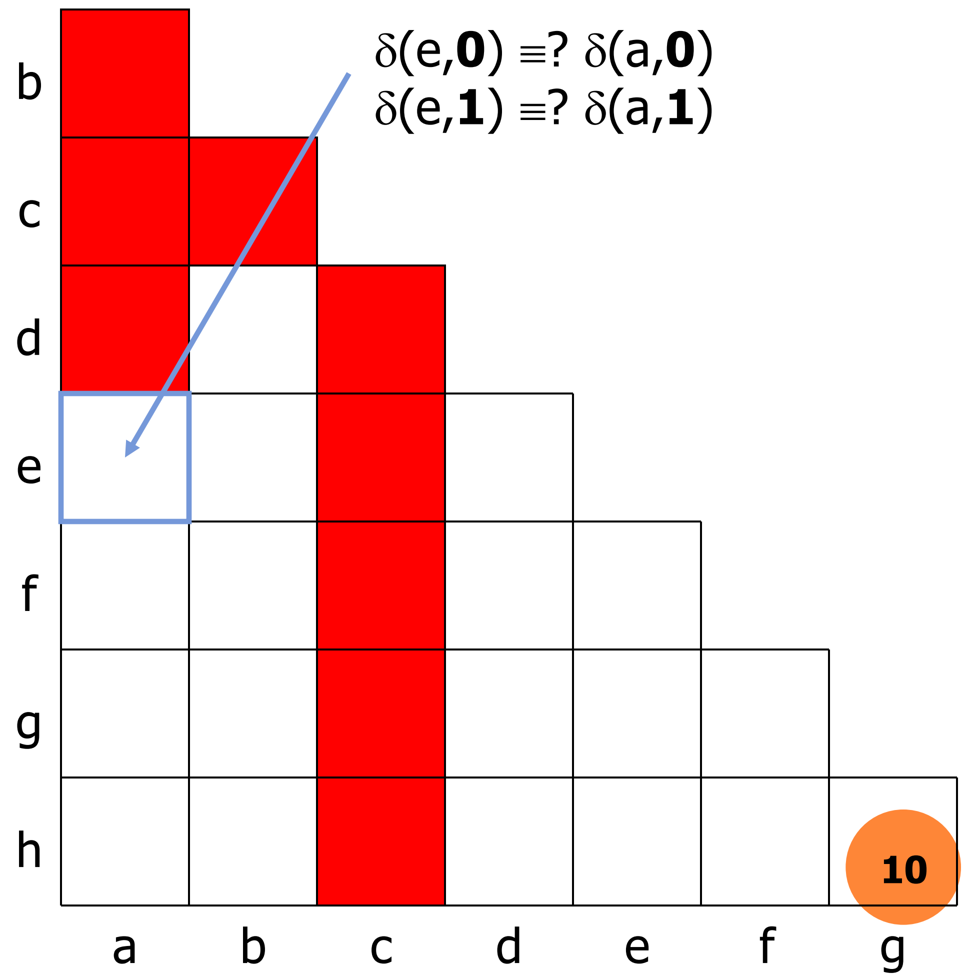




3. For each unmarked pair & symbol, …

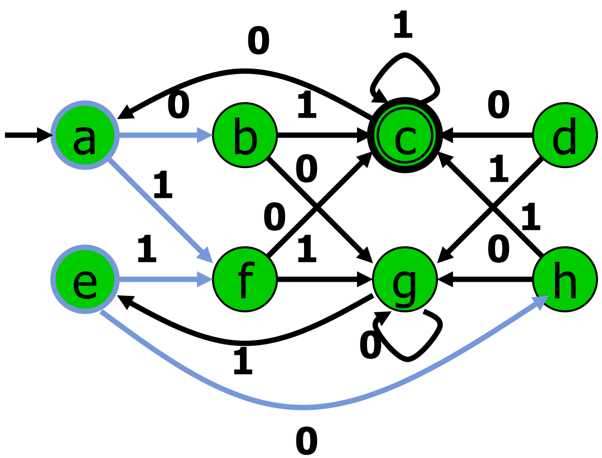
DFA MINIMIZATION: EXAMPLE



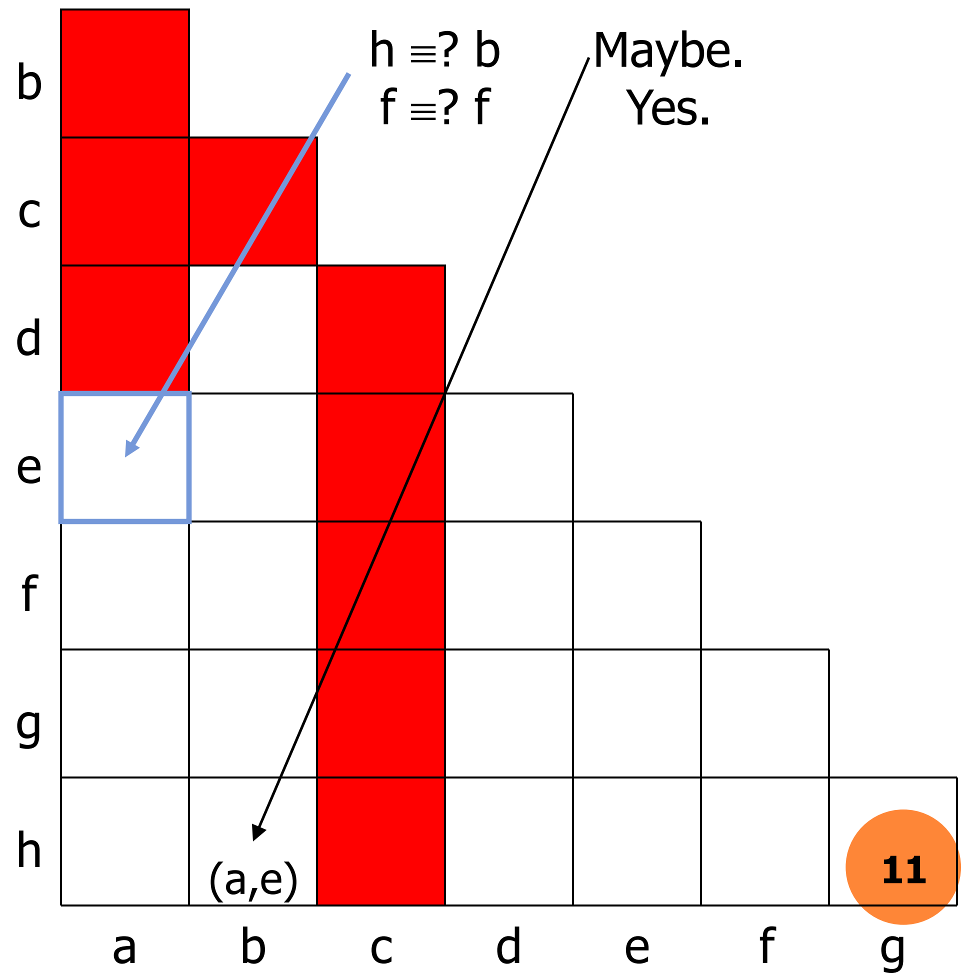


3. For each unmarked pair & symbol, …

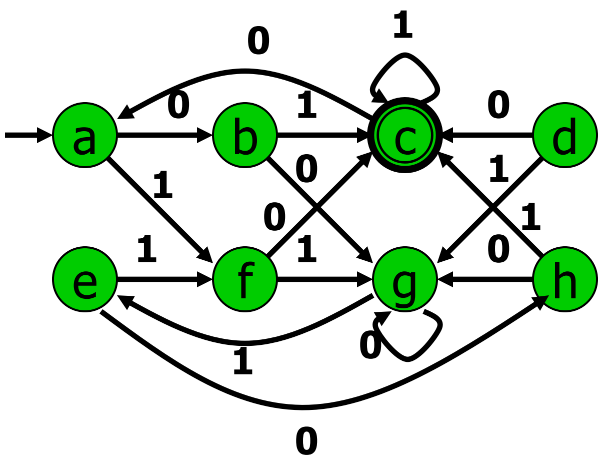
DFA Minimization: Example



3. For each unmarked pair & symbol, …



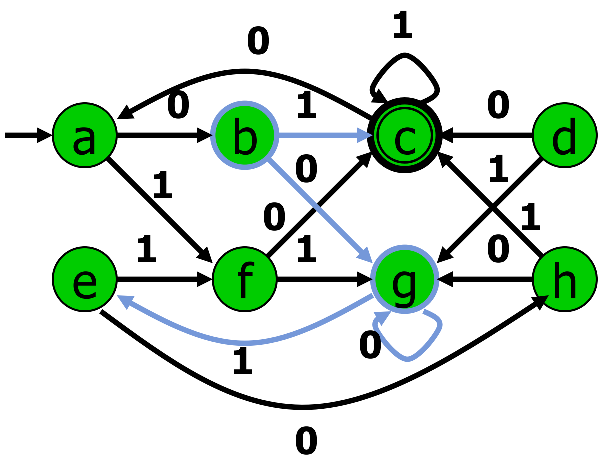
DFA Minimization: Example



3. For each unmarked pair & symbol, …

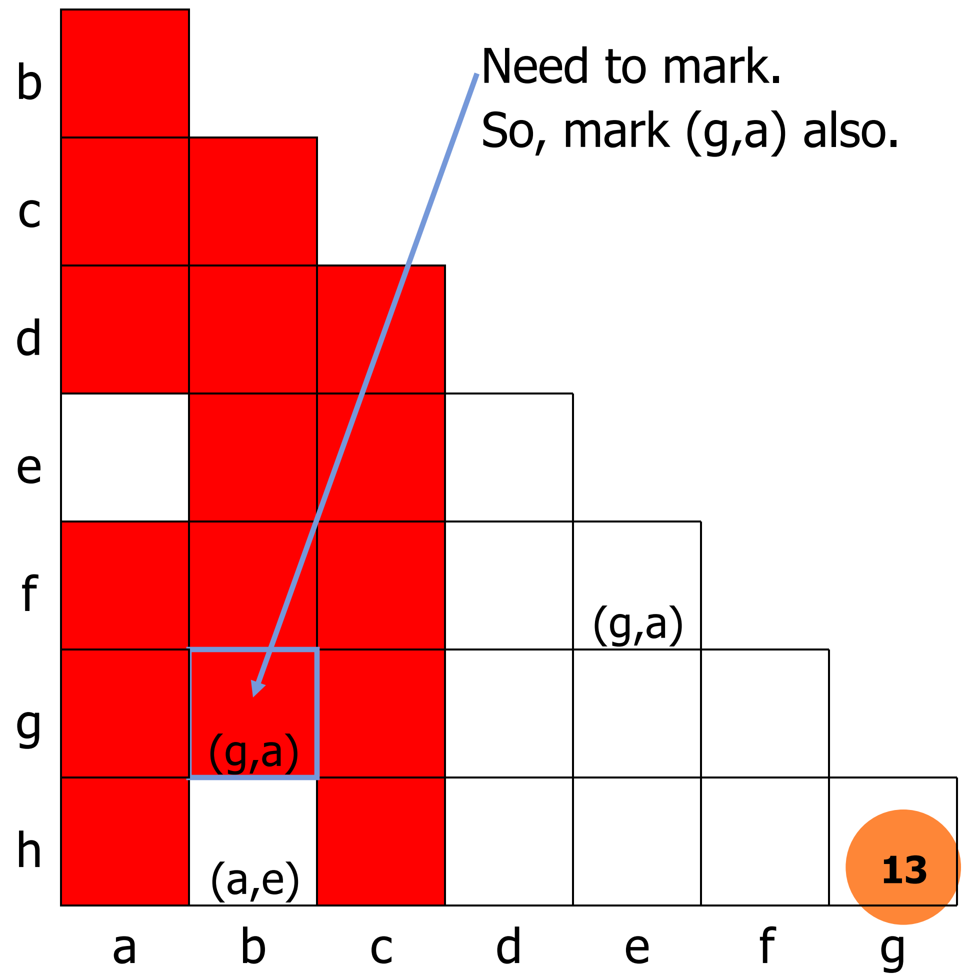
	a	b	c	d	e	f	g
b							
c							
d							
e							
f					(g,a)		
g		(g,a)					
h		(a,e)					

DFA Minimization: Example

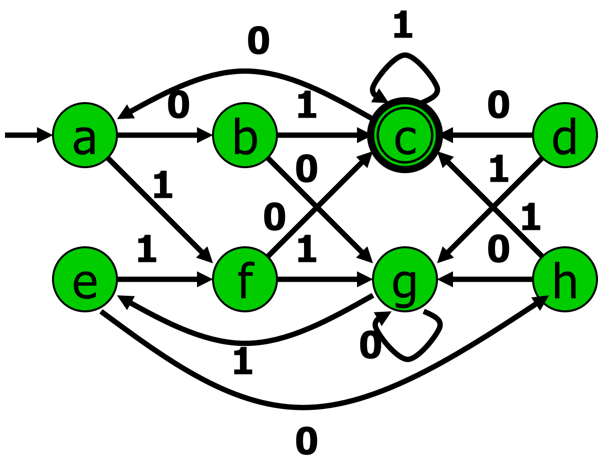


3. For each unmarked pair & symbol, …

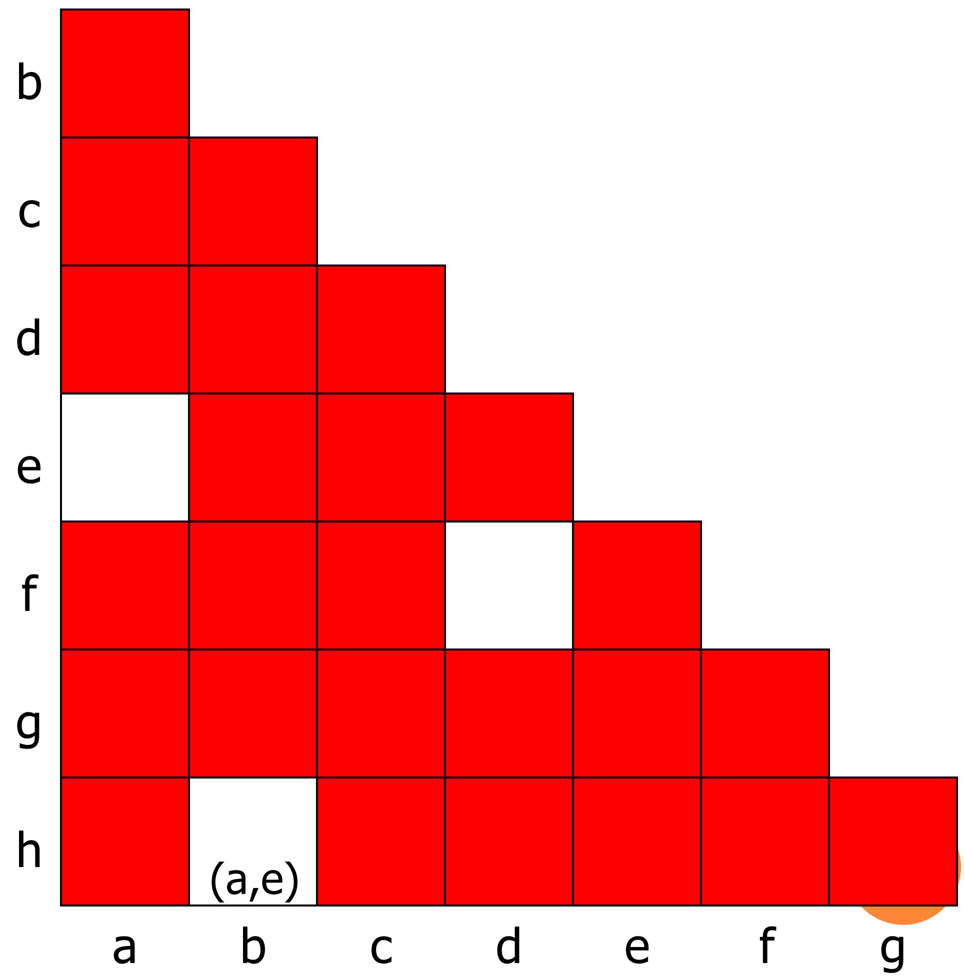
Need to mark.
So, mark (g,a) also.



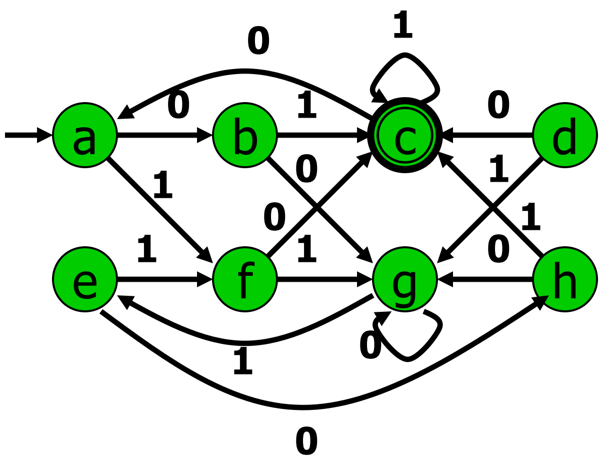
DFA Minimization: Example



3. For each unmarked pair & symbol, …

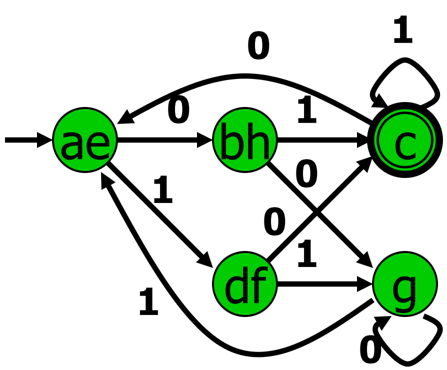


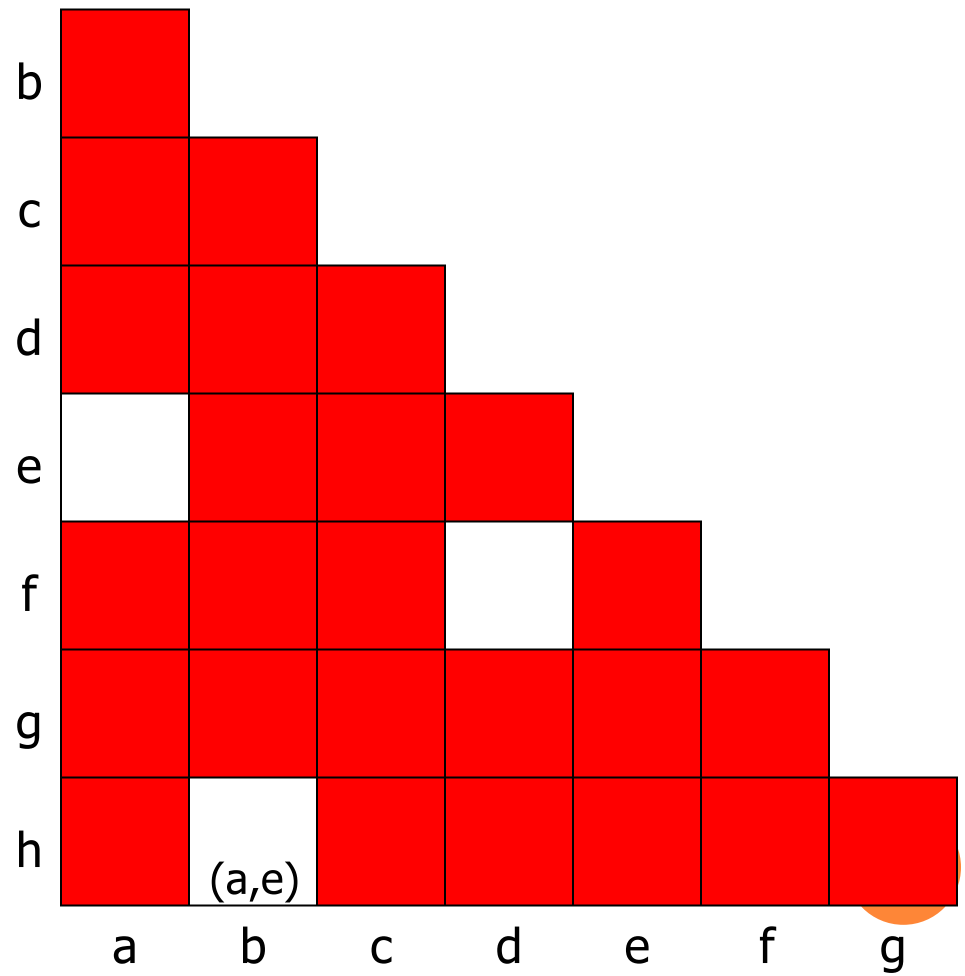
DFA Minimization: Example



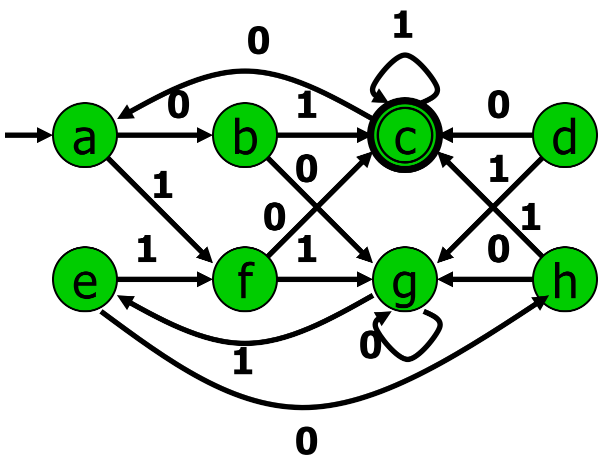
4. Coalesce unmarked pairs of states.

a ≡ e
b ≡ h
d ≡ f



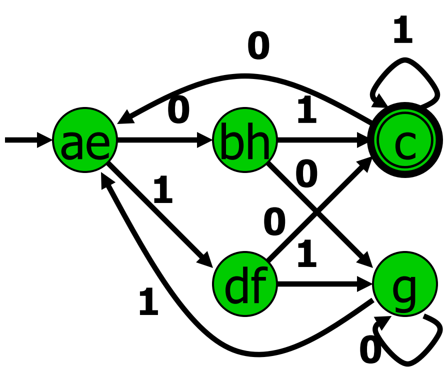


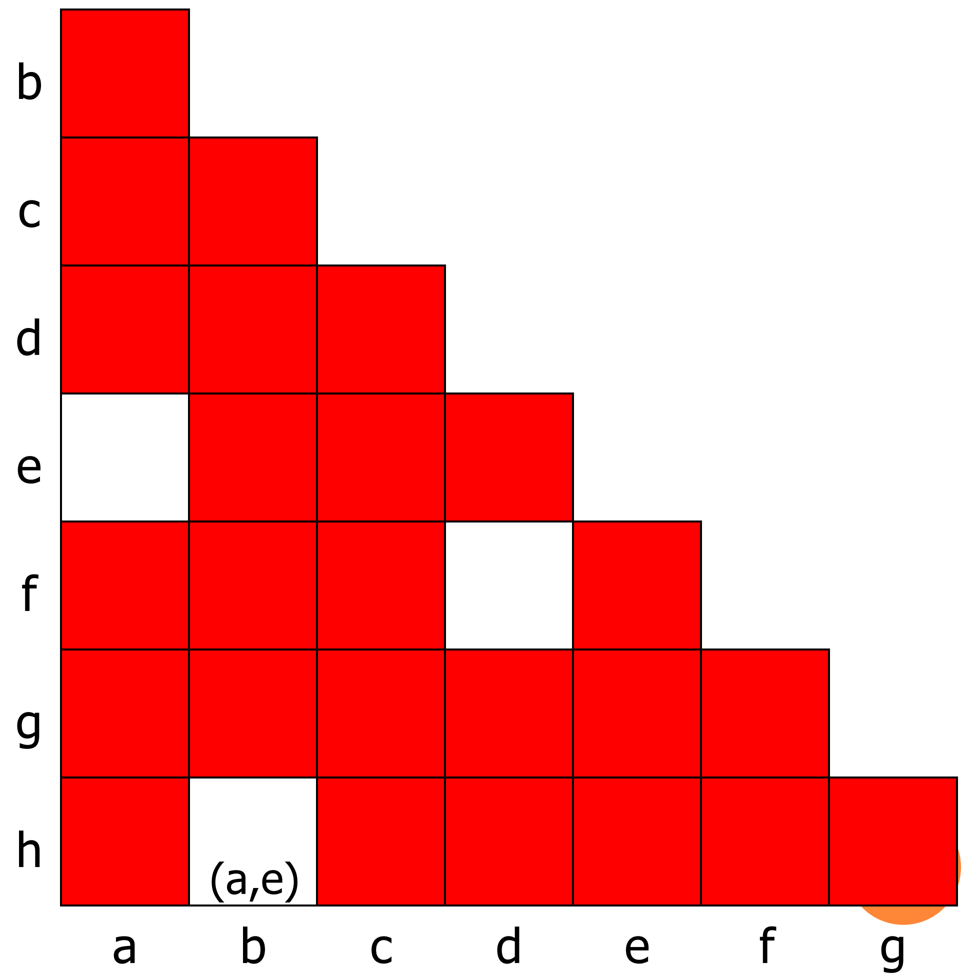
DFA Minimization: Example



5. Delete unreachable states.

None.





DFA Minimization: Notes

Order of selecting state pairs was arbitrary.

- All orders give same ultimate result.
- But, may record more or fewer dependences.
- Choosing states by working backwards from known non-equivalent states produces fewest dependences.

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.
- Obvious.

Why is new DFA equivalent to old DFA?

- Only identify states that provably have same behavior.
- Could prove $x \in L(M) \leftrightarrow x \in L(M')$ by inductions on derivations.

What About NFA Minimization?

This algorithm doesn't find a unique minimal NFA.

?

Is there a (not necessarily unique) minimal NFA?

?

Of course.

NFA MINIMIZATION

In general, minimal NFA not unique!

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

Both minimal, but not isomorphic.