# Computing Theory

COMP 147 (4 units)

## Chapter 1: Minimizing Finite State Machines

# Announcements

- Assignment2 due on Saturday
- Quiz2 on Thursday
    - Closure properties of Regular languages
    - Regular Expressions

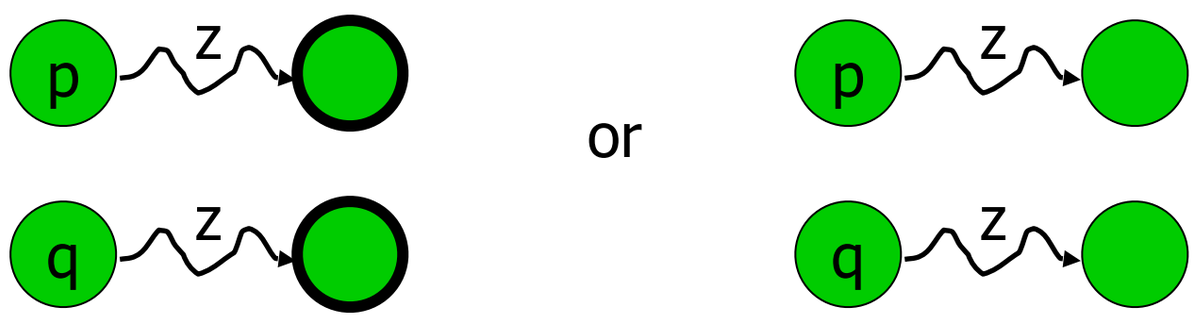
# The Minimum-State DFA for a Regular Language

- Given a DFA A,  can we find the DFA with the fewest states accepting L(A)?
  - Yes! Possible to find minimal number of states
  - Also Unique!

# State Minimization: Algorithm Idea

- Idea: *Equate & collapse states having same behavior.*

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



# Efficient State Minimization

- Construct a table with all pairs of states.
- If you find a string that distinguishes two states (takes exactly one to an accepting state), mark that pair.
- 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.

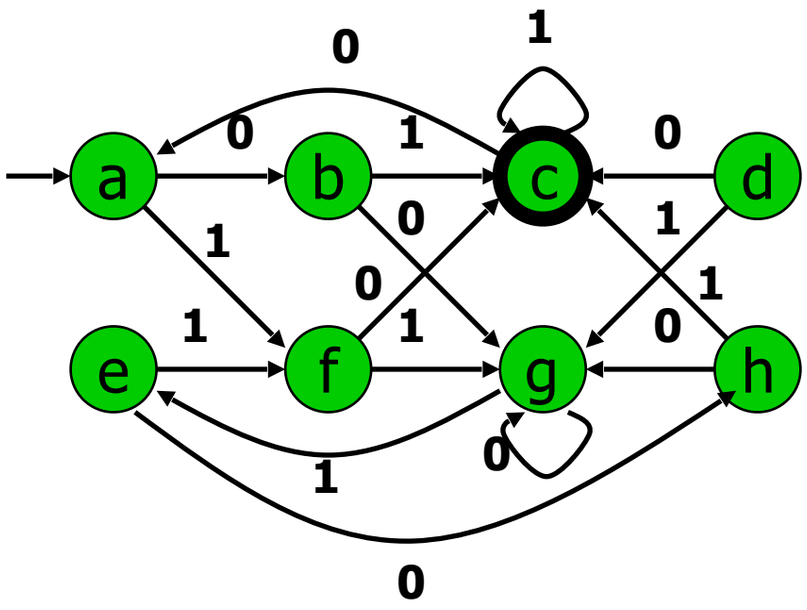
2. Mark all pairs of a final & nonfinal state.

3. For each unmarked pair p,q & input symbol a:
   1. Let r=δ(p,a), s=δ(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.

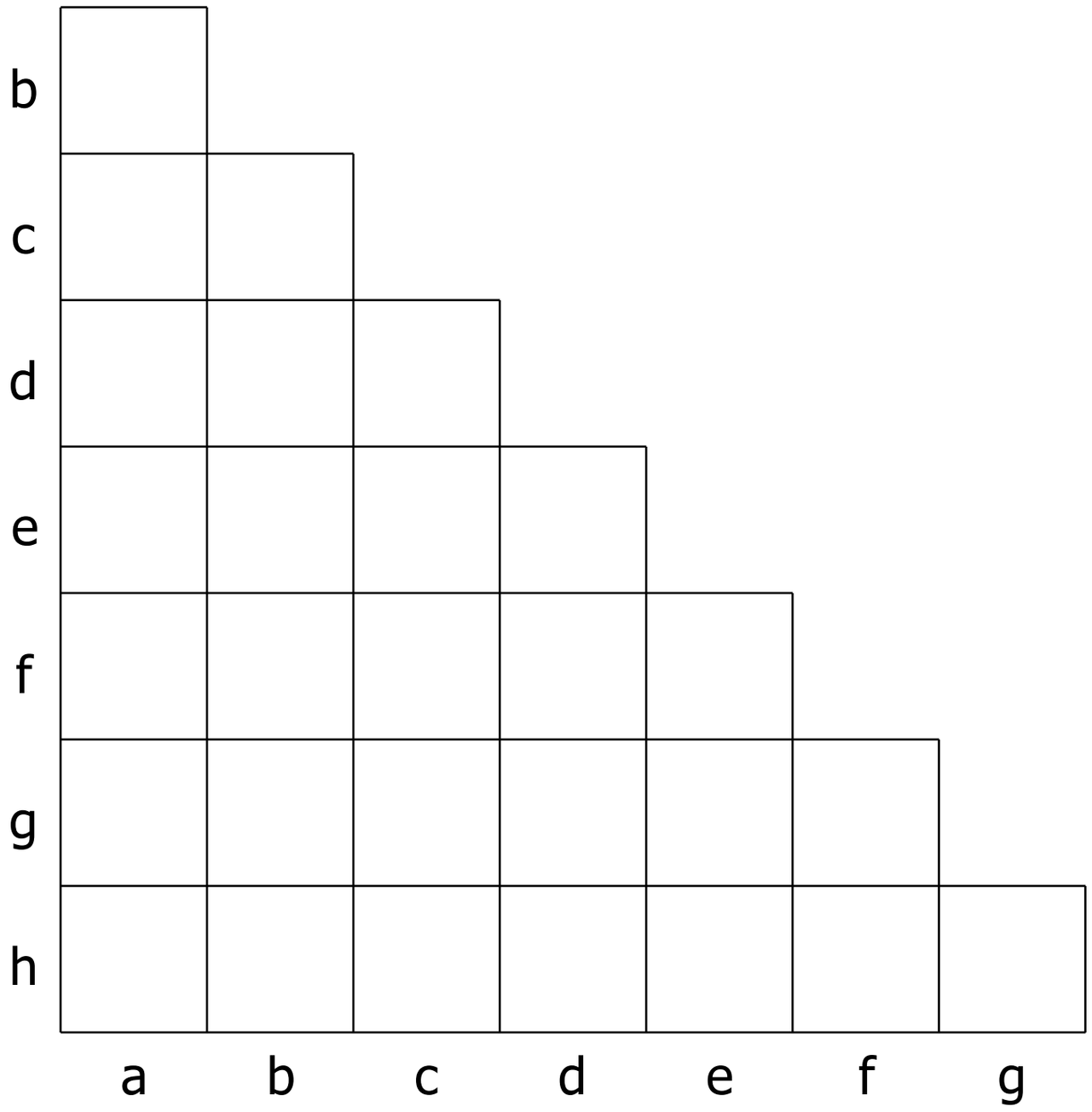
4. Coalesce unmarked pairs of states.

5. Delete inaccessible states.

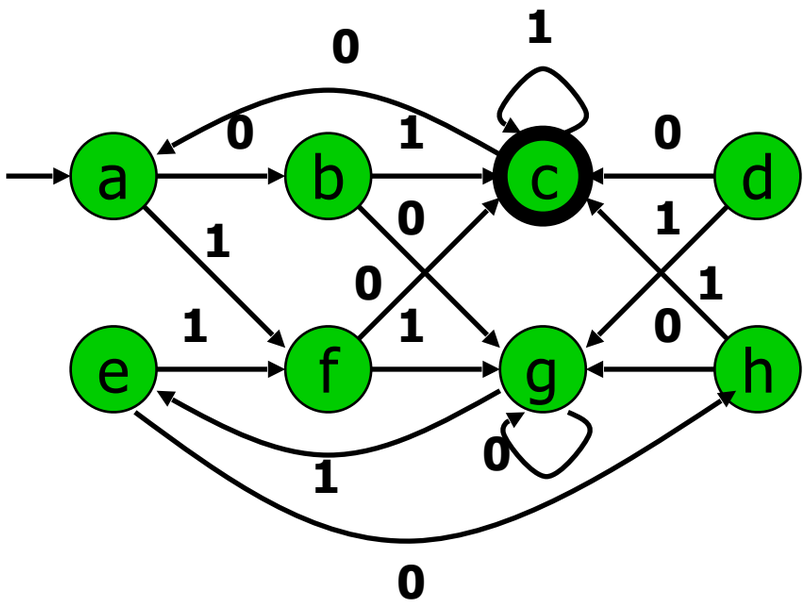
# DFA Minimization: Example



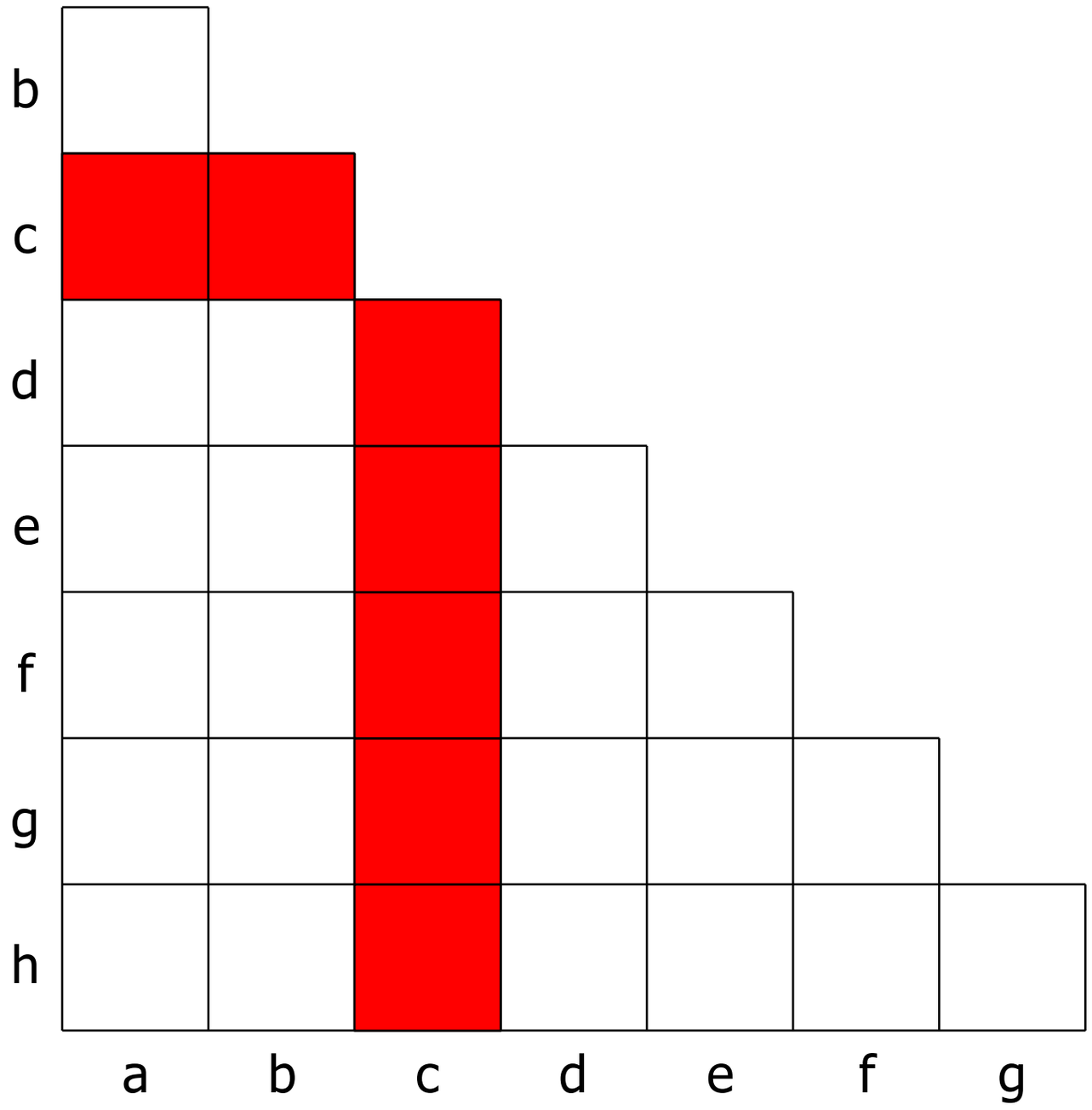

1. Initialize table entries:
   Unmarked, empty list



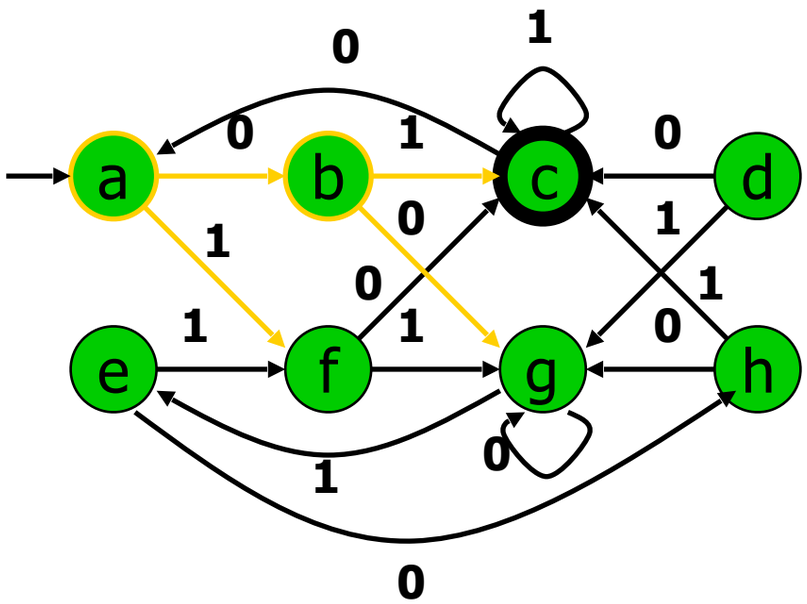
# DFA Minimization: Example



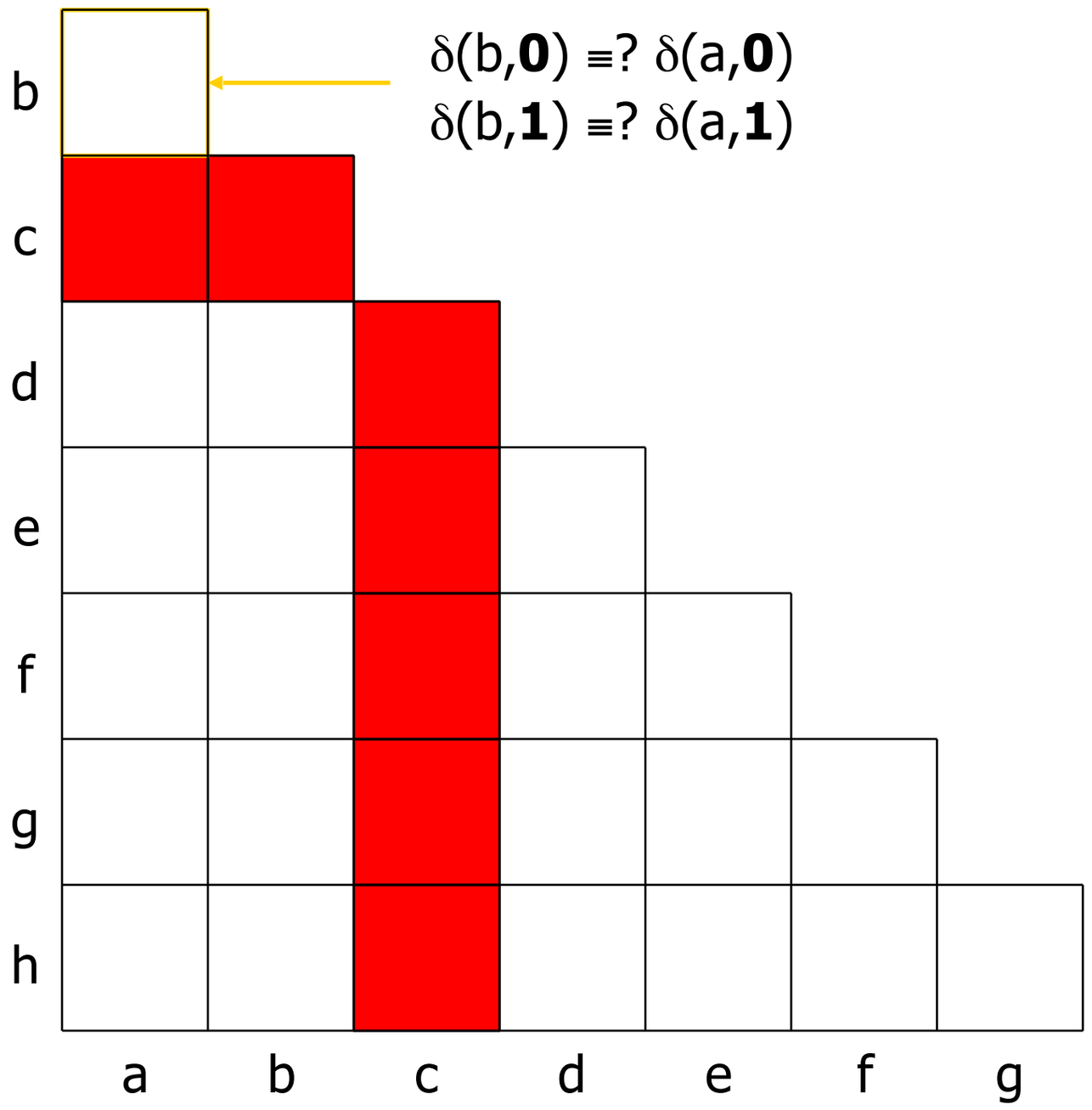

2. Mark pairs of final & nonfinal states



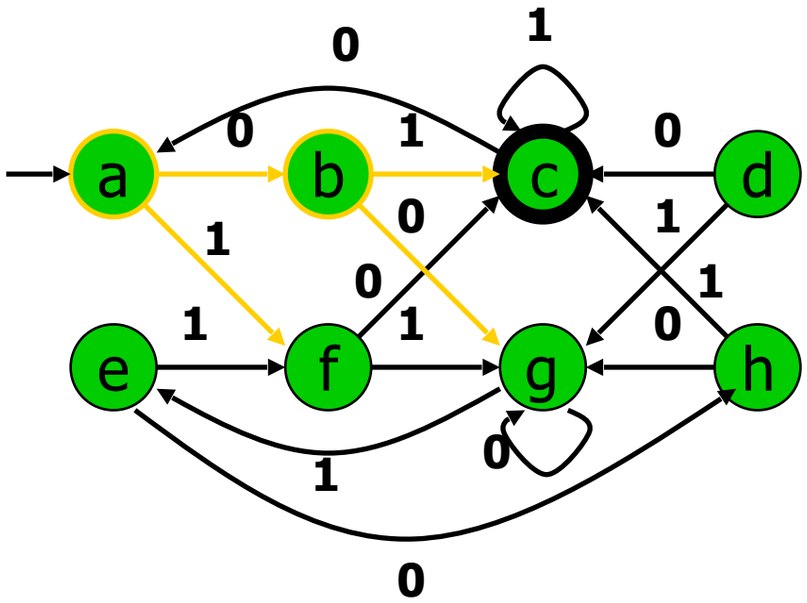
# DFA Minimization: Example



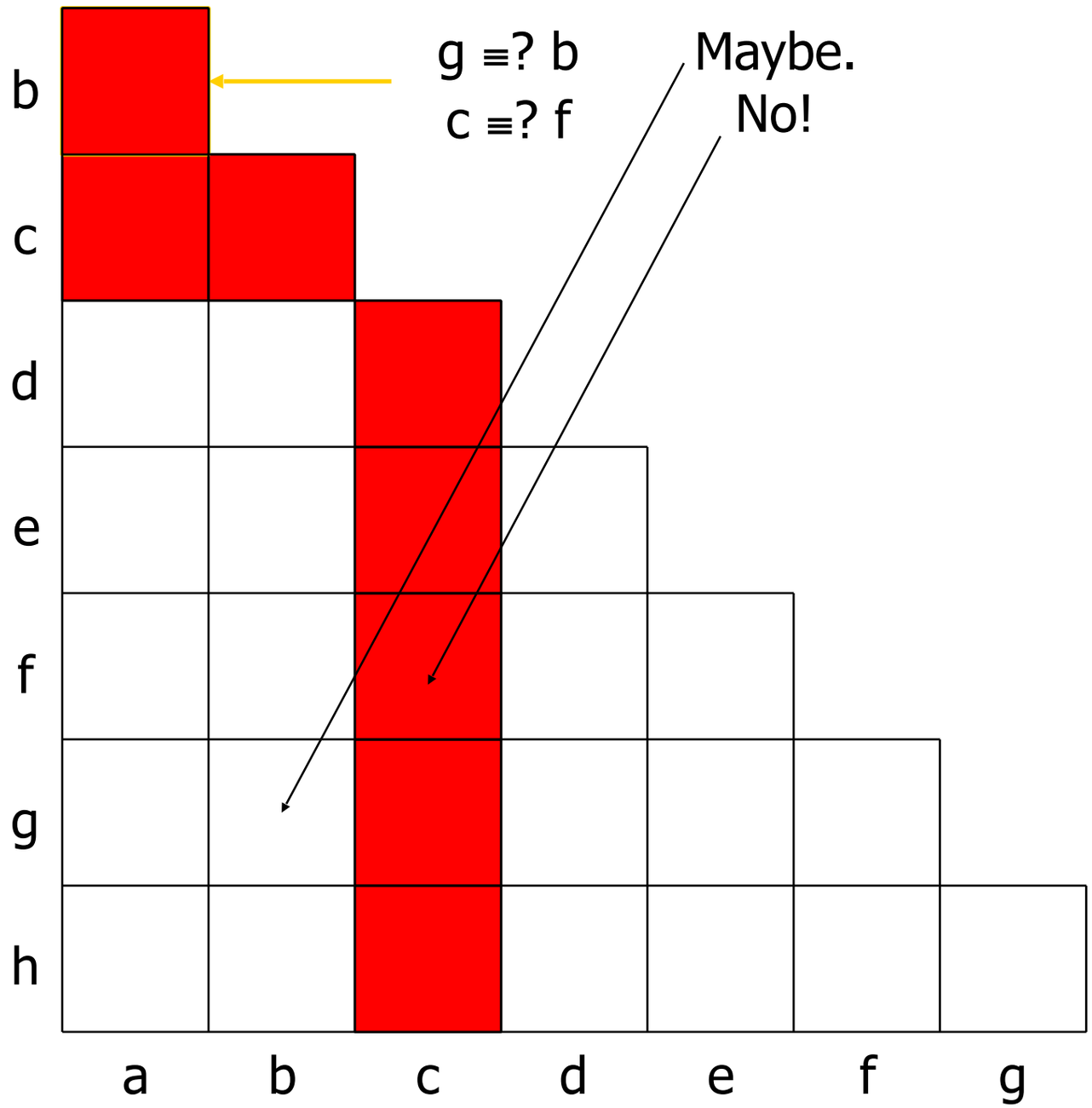

3. For each unmarked pair & symbol, …



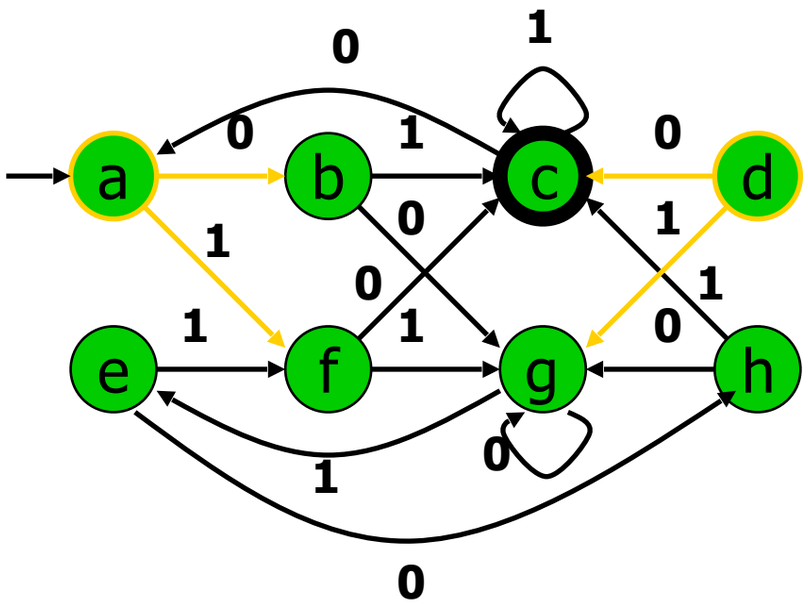
# DFA Minimization: Example



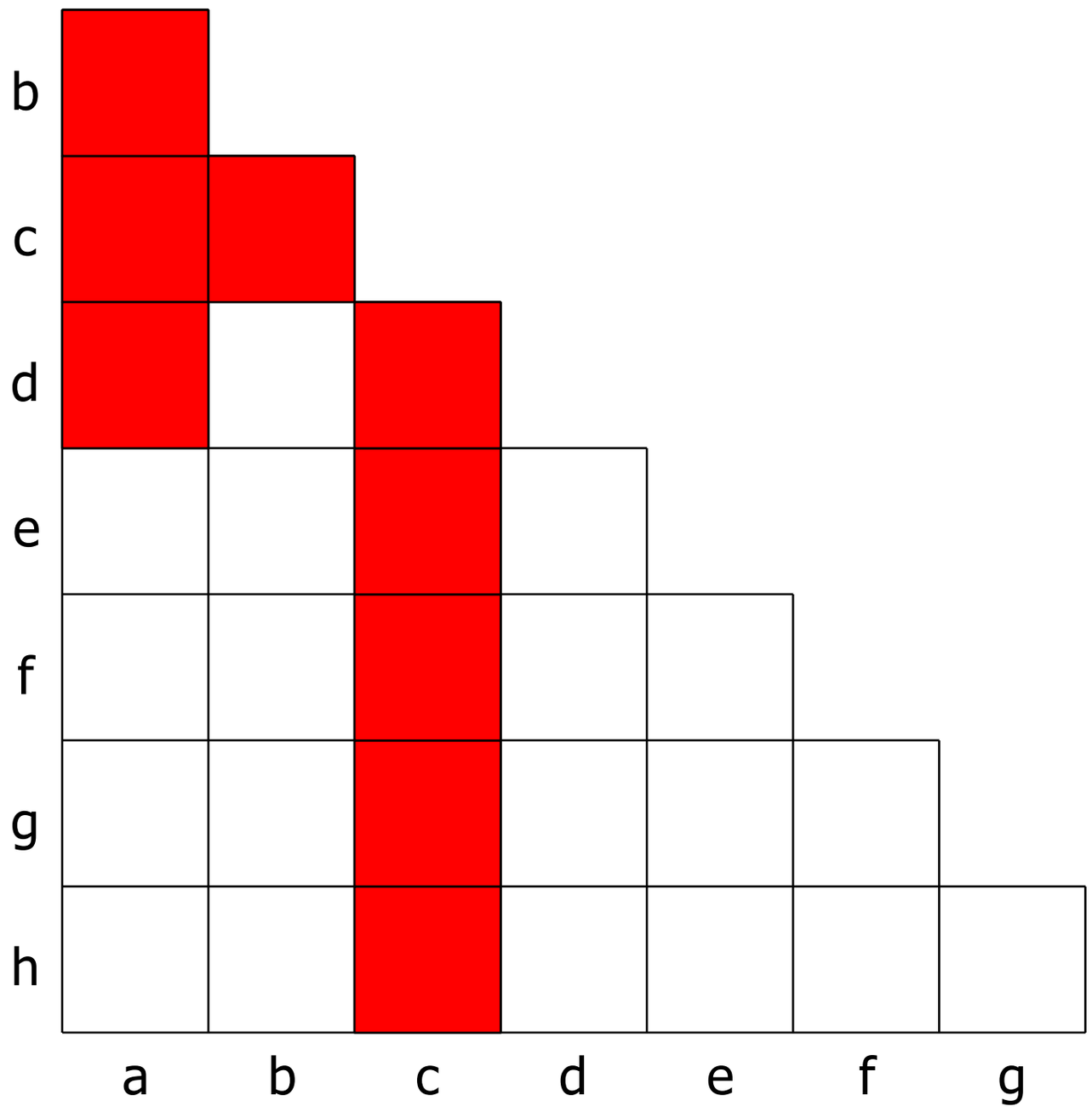

3. For each unmarked pair & symbol, …



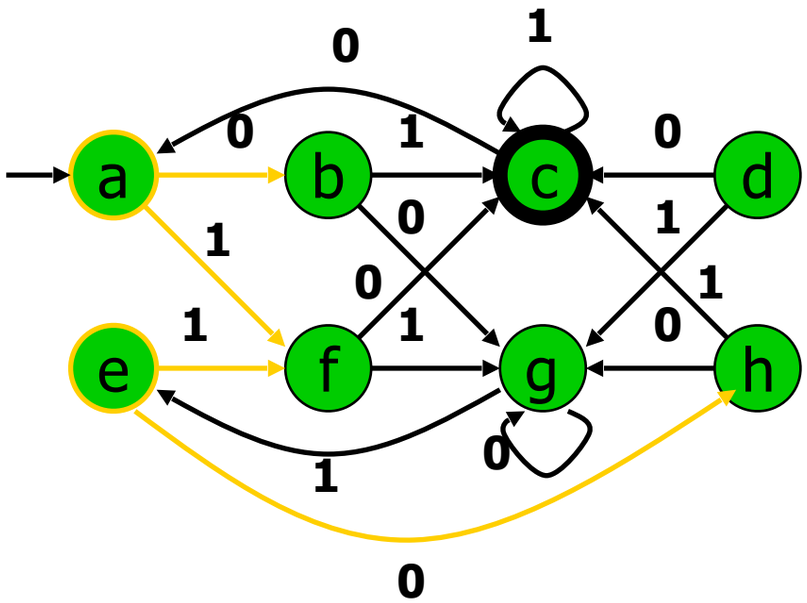
# DFA Minimization: Example



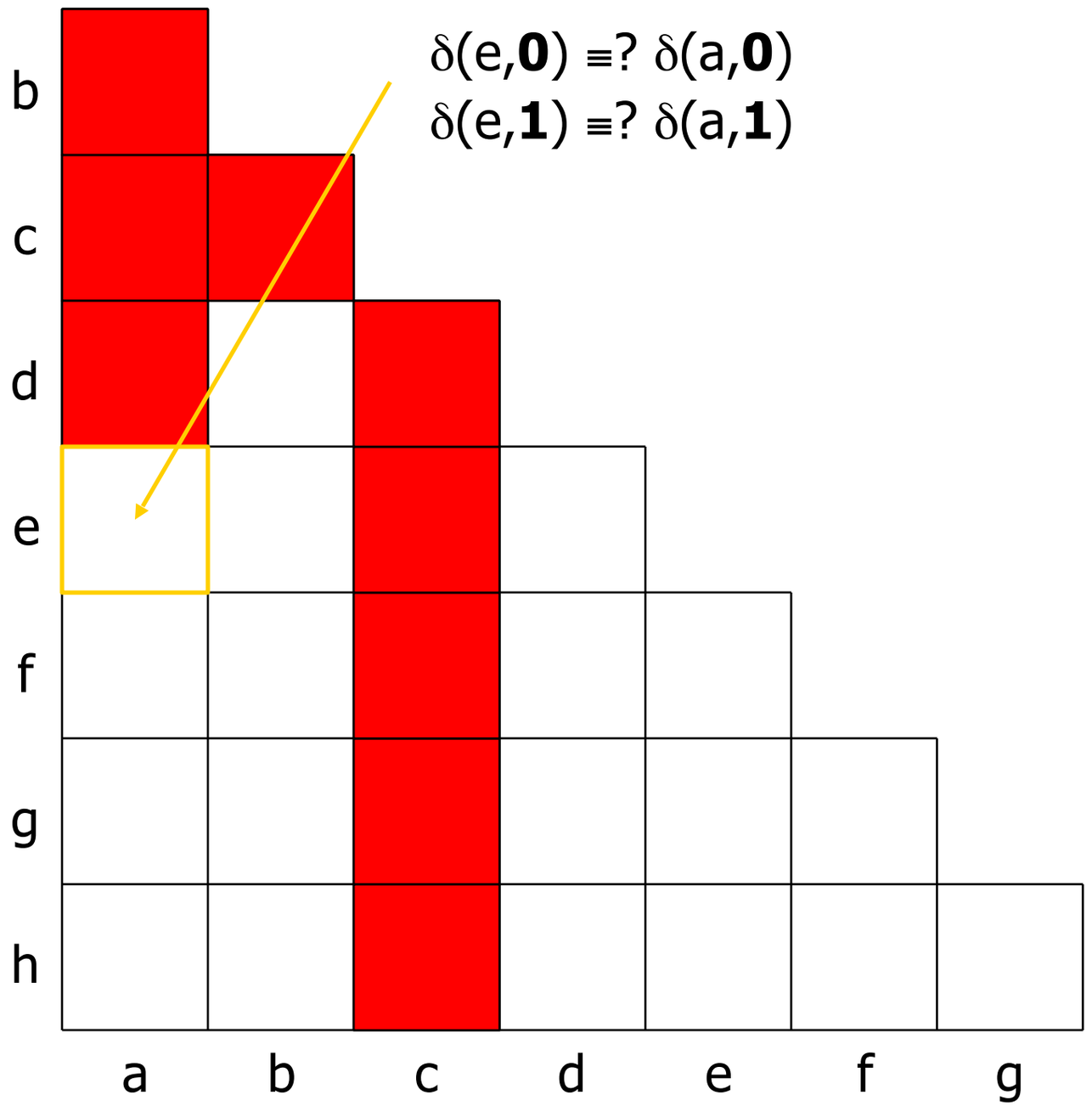

3. For each unmarked pair & symbol, …



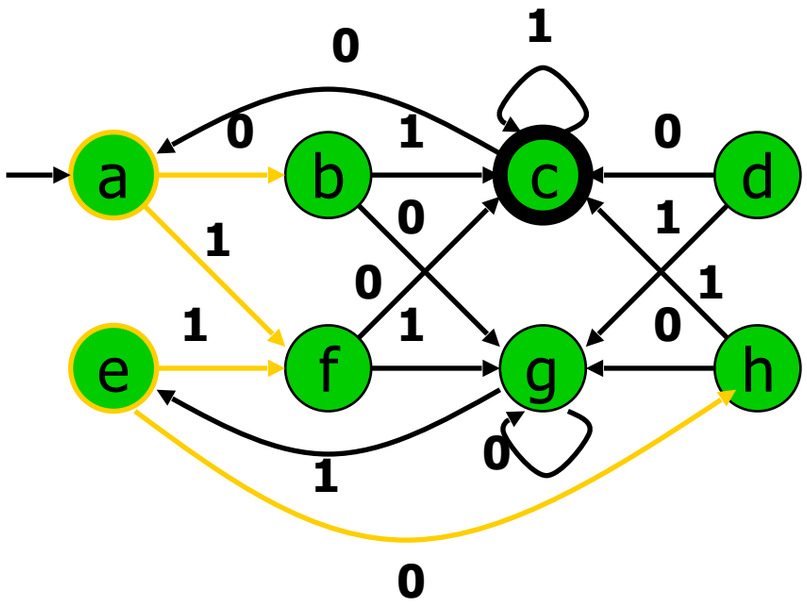
# DFA Minimization: Example



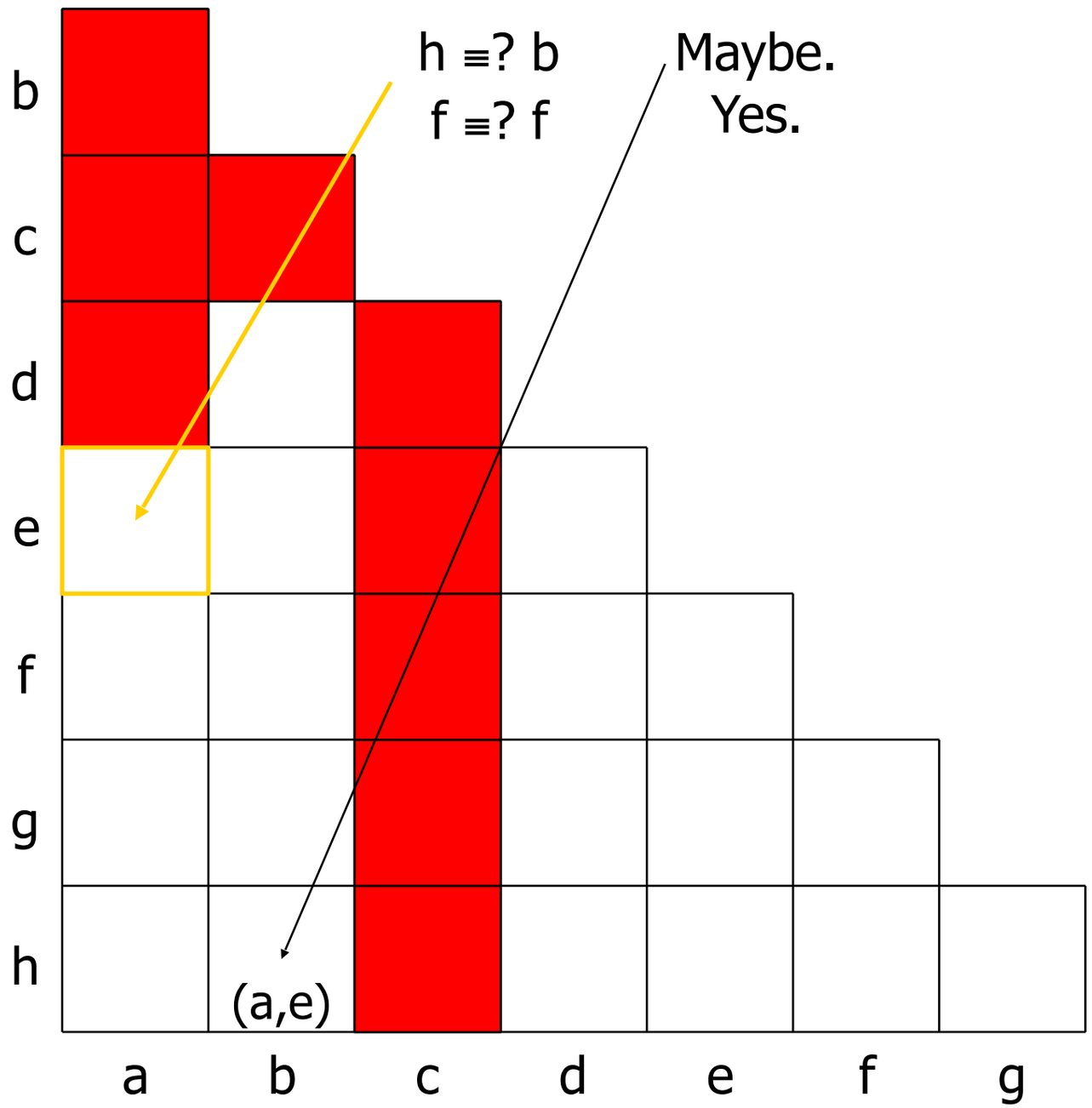

3. For each unmarked pair & symbol, …



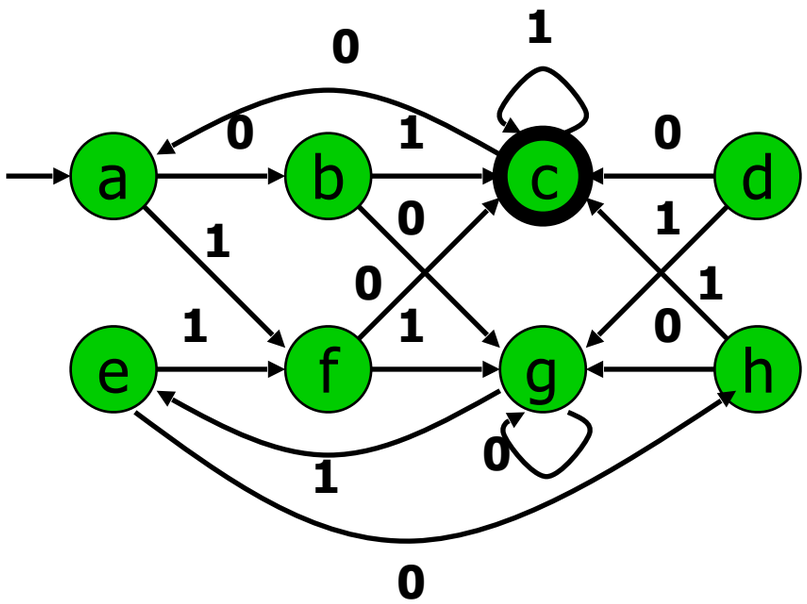
# DFA Minimization: Example



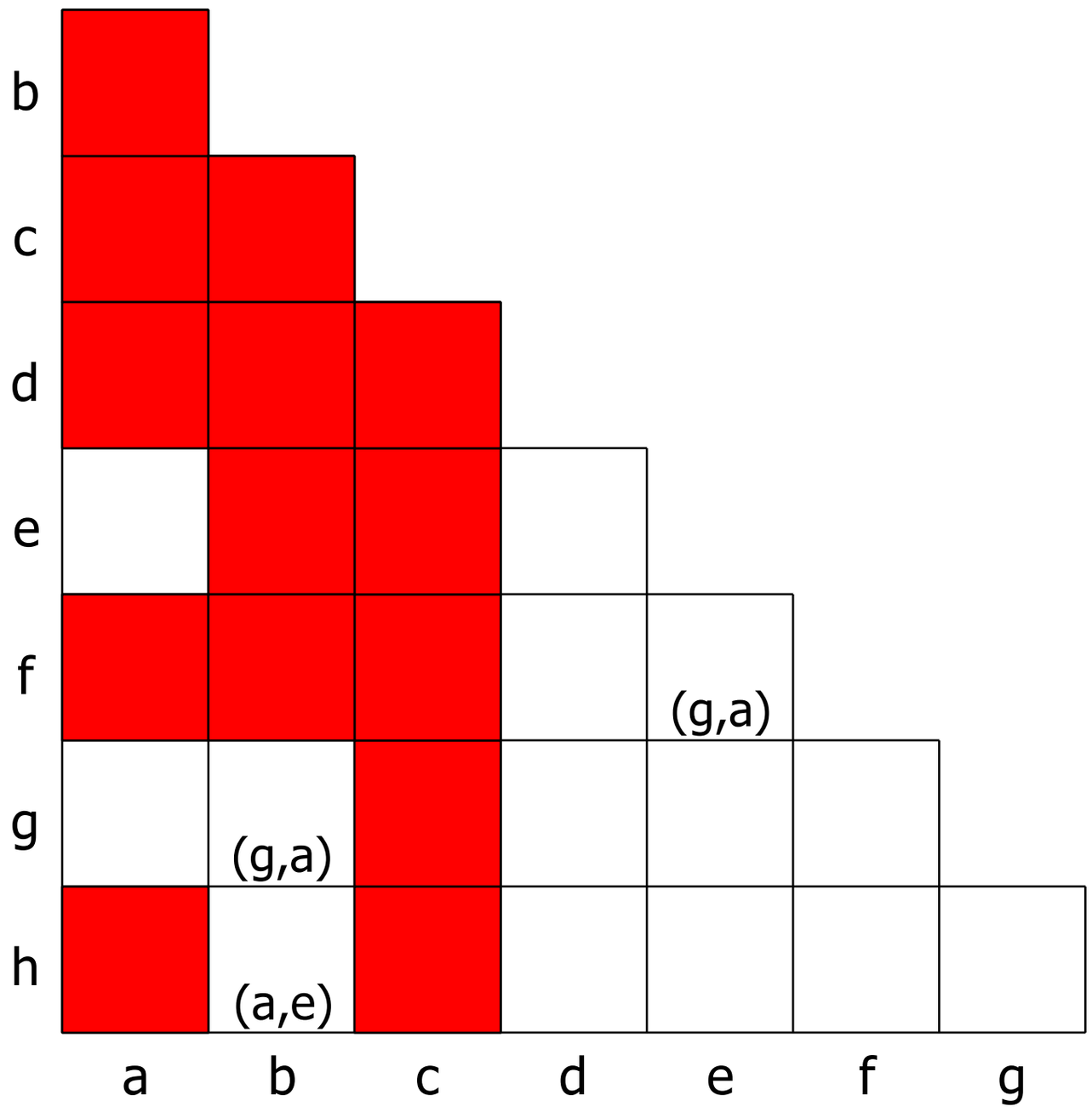

3. For each unmarked pair & symbol, …



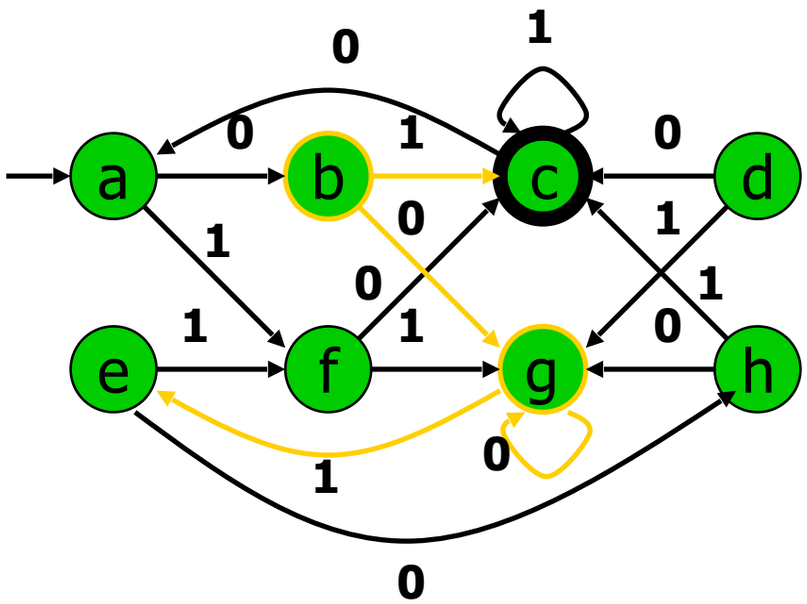
# DFA Minimization: Example



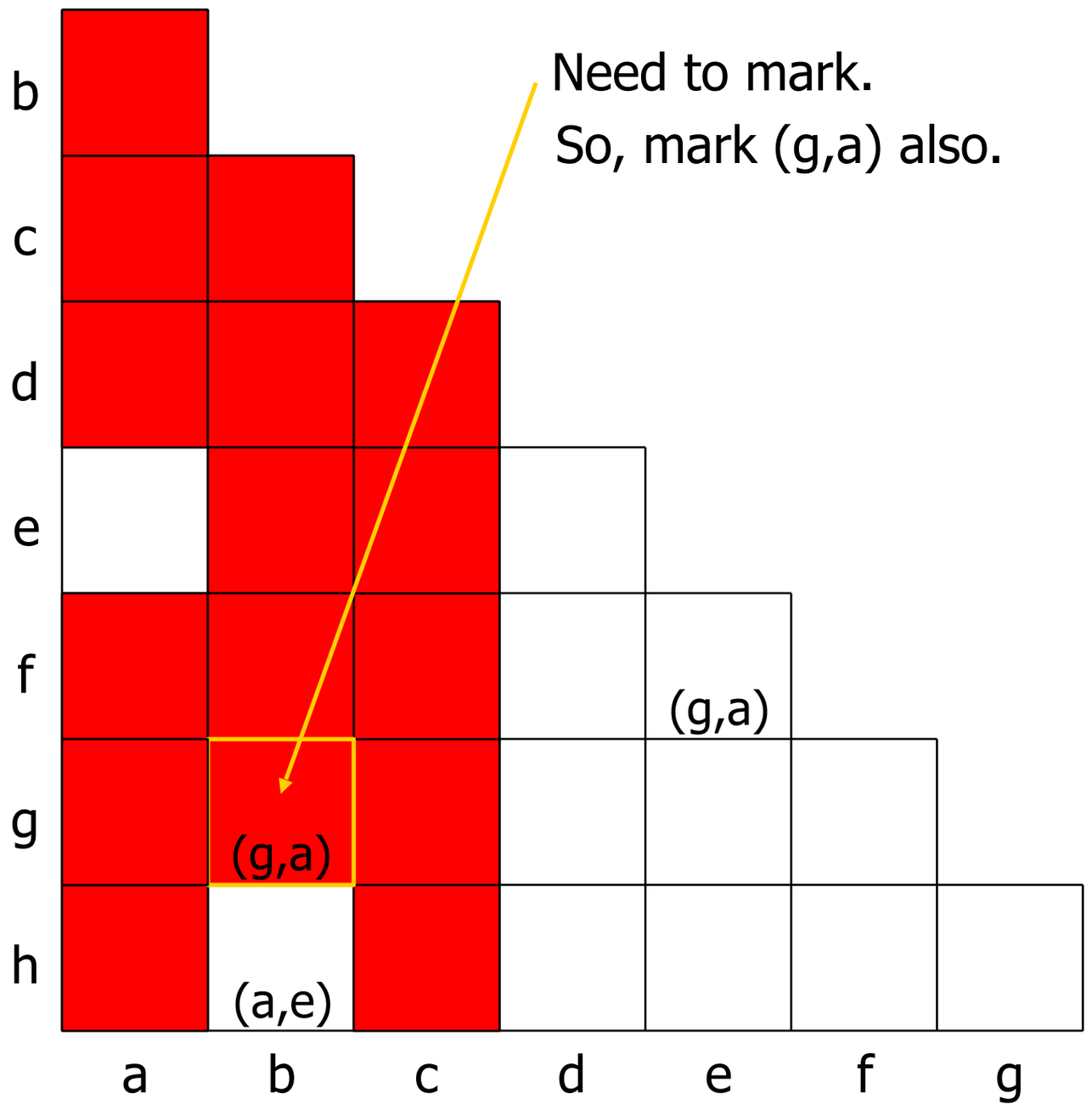

3. For each unmarked pair & symbol, …



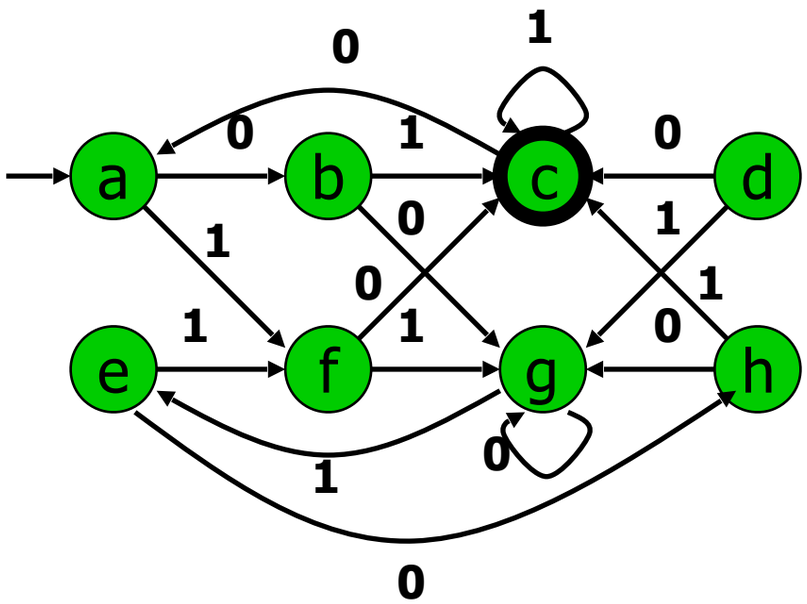
# DFA Minimization: Example



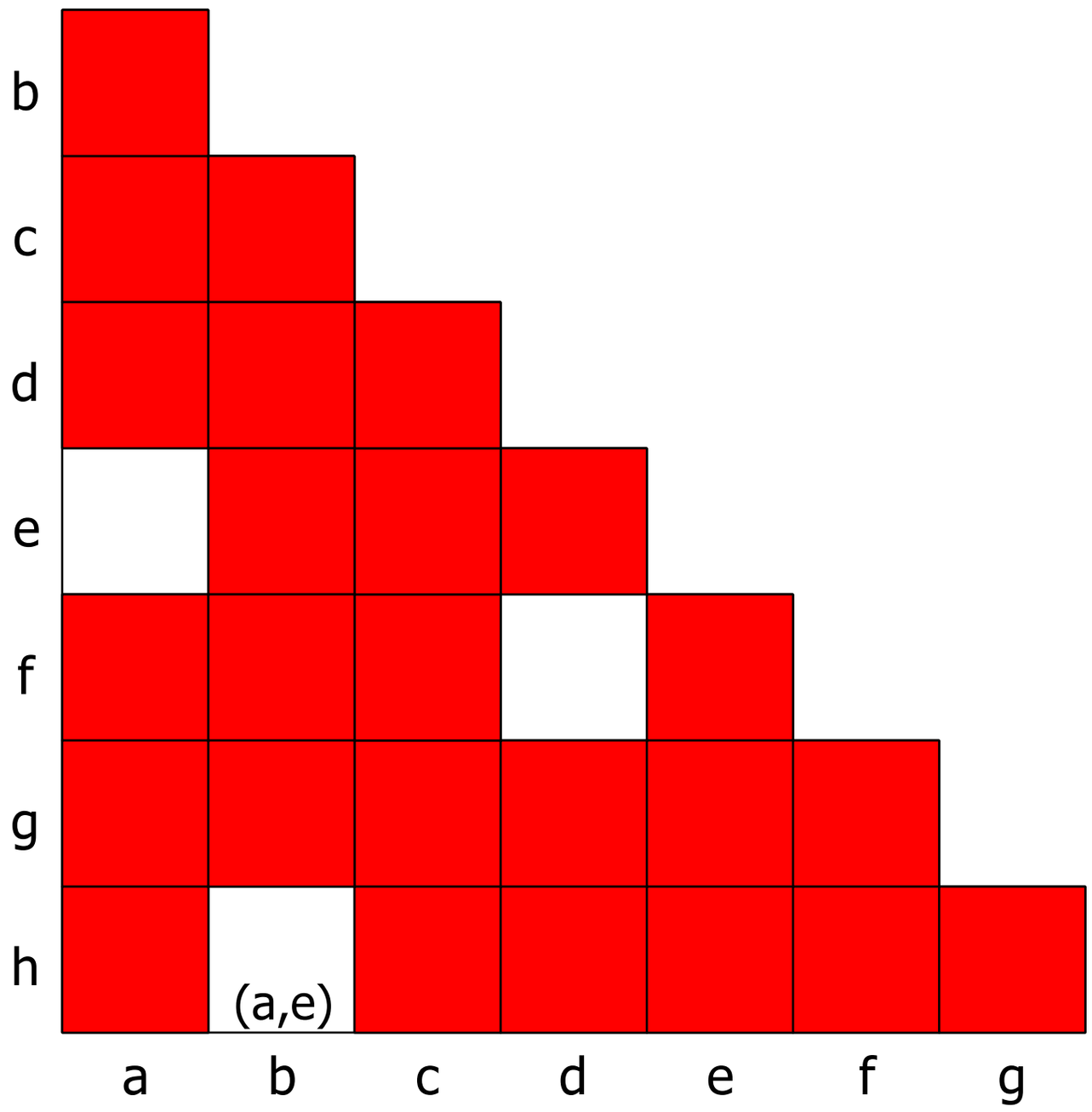

3. For each unmarked pair & symbol, …



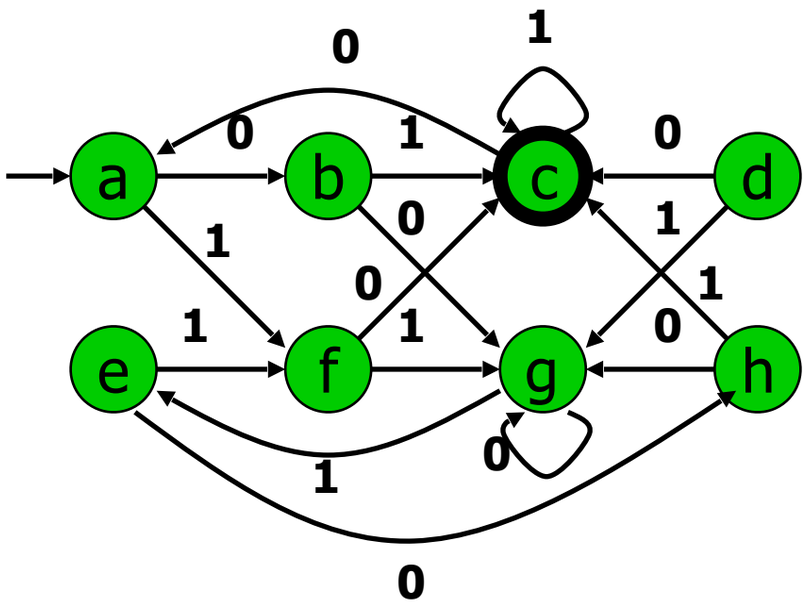
# DFA Minimization: Example

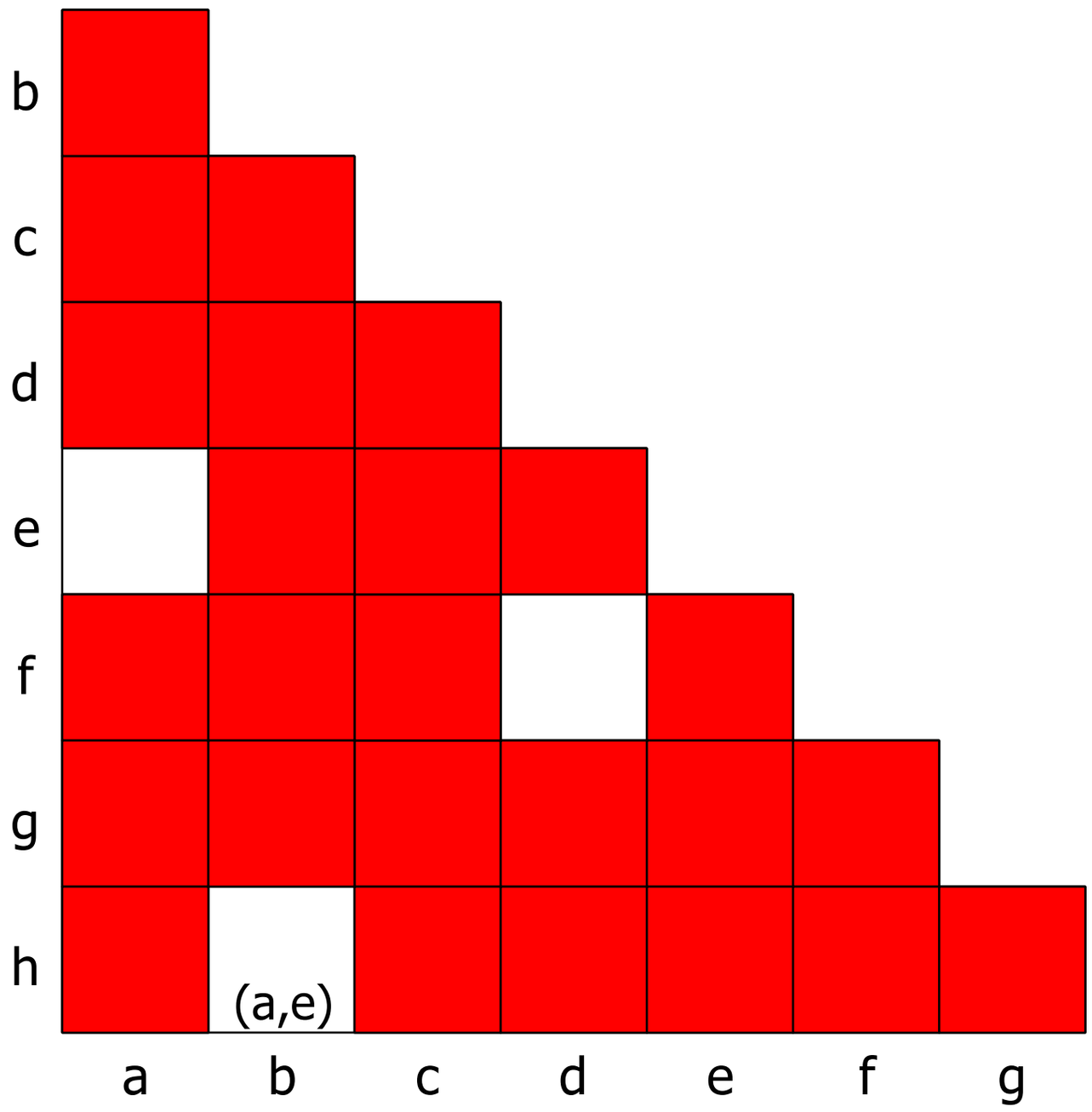



3. For each unmarked pair & symbol, …



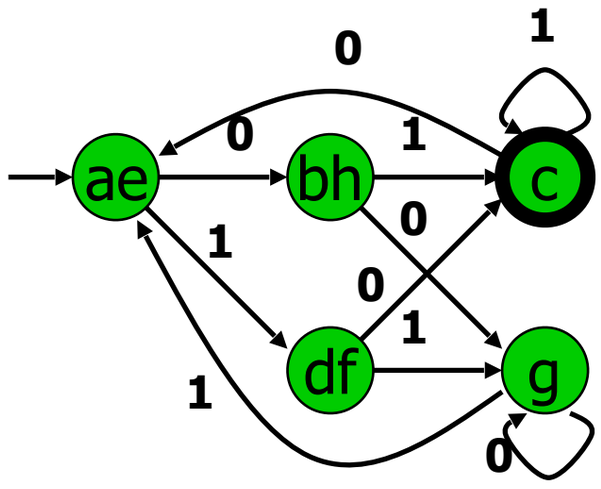
# DFA Minimization: Example



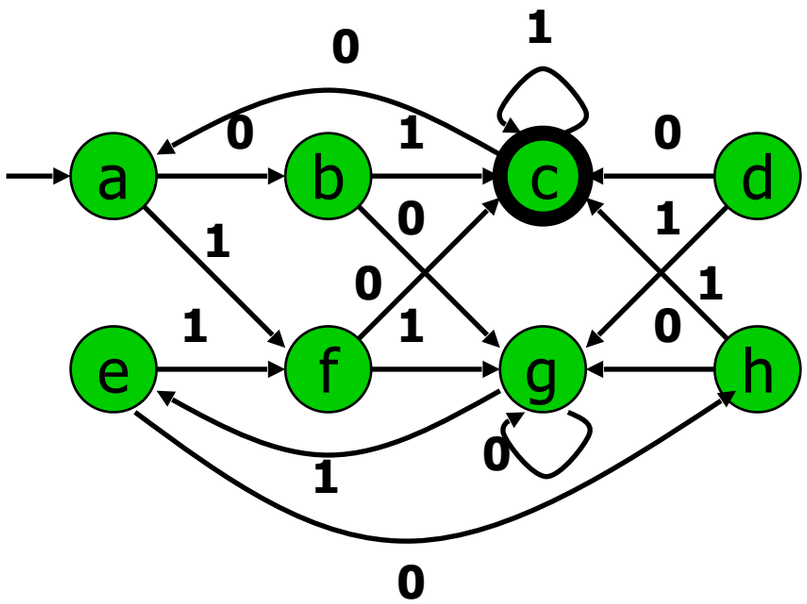



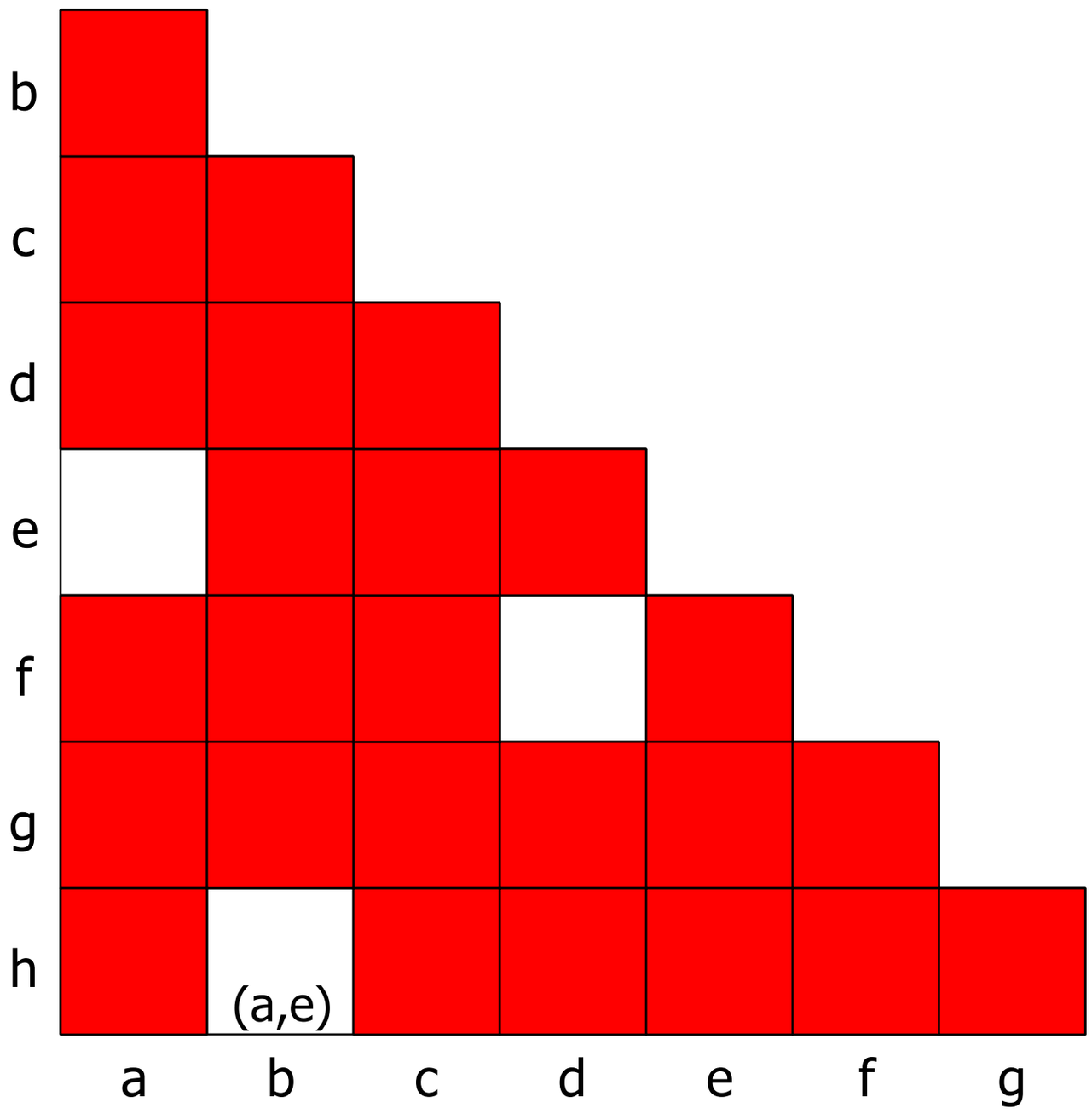

4. Coalesce unmarked pairs of states.

a ≡ e
b ≡ h
d ≡ f



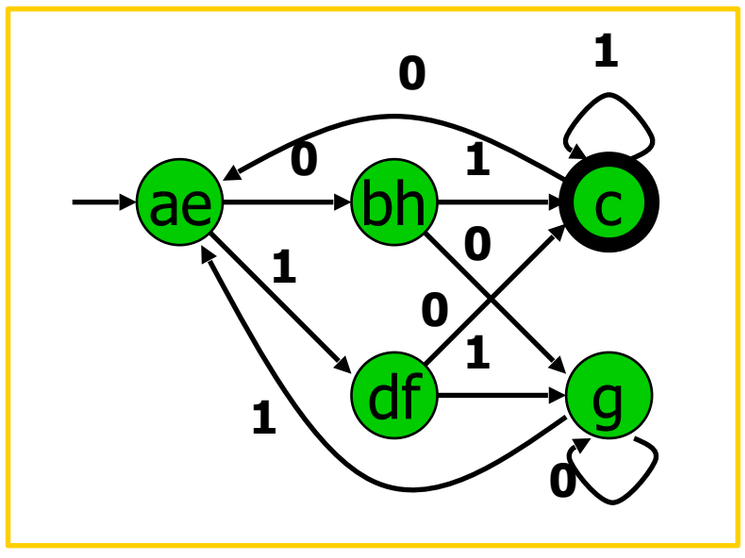
# DFA Minimization: Example




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.

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).