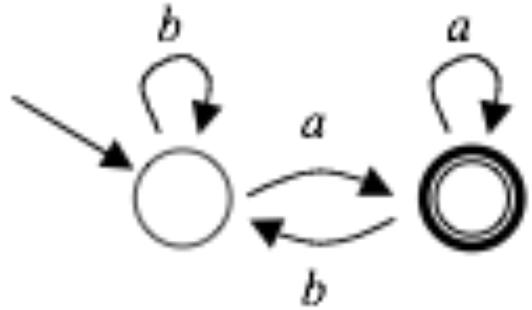
Deterministic Finite Automata (DFAs)

3/29/24

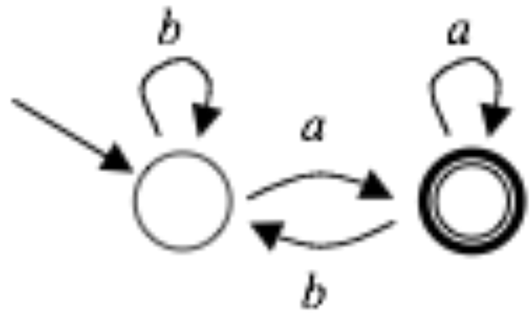
(Using slides adapted from the book)

Deterministic Finite Automata (DFA)



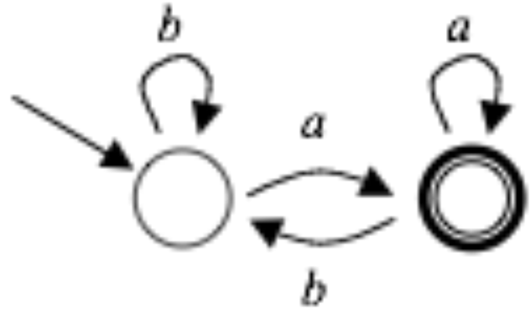
- ”Machine” to decide whether to accept a string
  - *States*: Circles; where the computation can be
  - *Start state* (w/ unlabeled incoming arrow) is where it begins
  - Machine reads the characters one at a time and follows corresponding arrows (*transitions*)
  - At end of input, accepts if in double-circled state (an *accepting state*) and rejects otherwise (i.e. if in *non-accepting state*)

Deterministic Finite Automata (DFA)



What strings does this machine accept?

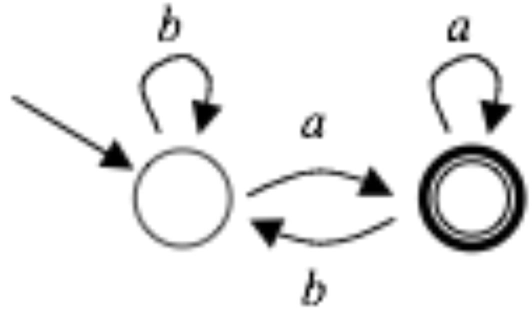
Deterministic Finite Automata (DFA)



- This DFA accepts $\{xa \mid x \in \{a,b\}^*\}$

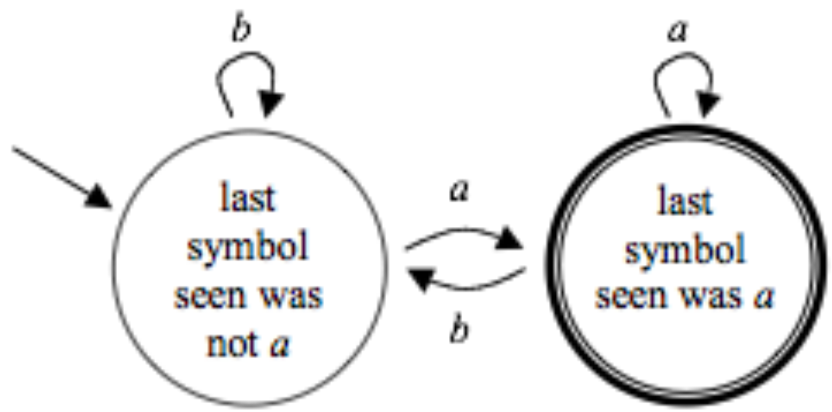
  "words ending in a"

Deterministic Finite Automata (DFA)



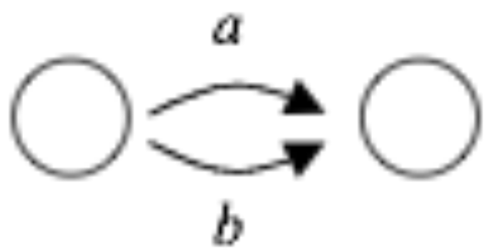
- This DFA accepts $\{xa \mid x \in \{a,b\}^*\}$

  "words ending in a"
- Labels can be added, but they have no effect, like program comments:

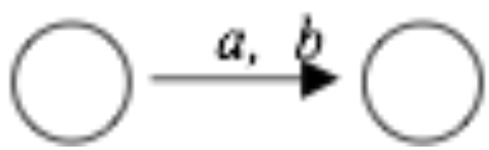


A DFA Convention

- We don't draw multiple arrows with the same source and destination states:



- Instead, we draw one arrow with a list of symbols:



DFAs Define Languages

- Given any string over Σ, a DFA can read the string and follow its state-to-state transitions
- At the end of the string, if it is in an accepting state, we say it accepts the string
- Otherwise it rejects
- The language defined by a DFA is the set of strings in Σ^* that it accepts. To identify this:
  - Show every word in the language is accepted
  - Show every accepted string is in the language

A Classic Riddle

- A man must cross river with wolf, goat and cabbage
- Has rowboat w/ room for man plus <u>one</u> possession
- If left alone together:
  - Wolf eats goat
  - Goat eats cabbage

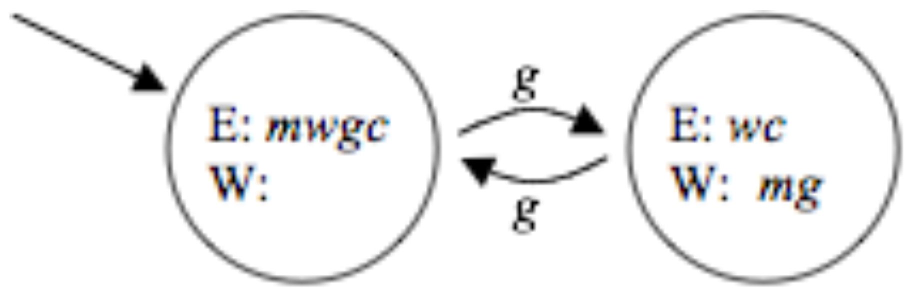
How can the man cross without loss?

Solutions As Strings

- Four moves can be encoded as four symbols:
  - Man crosses with wolf (*w*)
  - Man crosses with goat (*g*)
  - Man crosses with cabbage (*c*)
  - Man crosses with nothing (*n*)
- Then sequence of moves is a string, such as *gnwgcng*:
  - First cross with *g*oat, then cross back with *n*othing, then cross with *w*olf, …

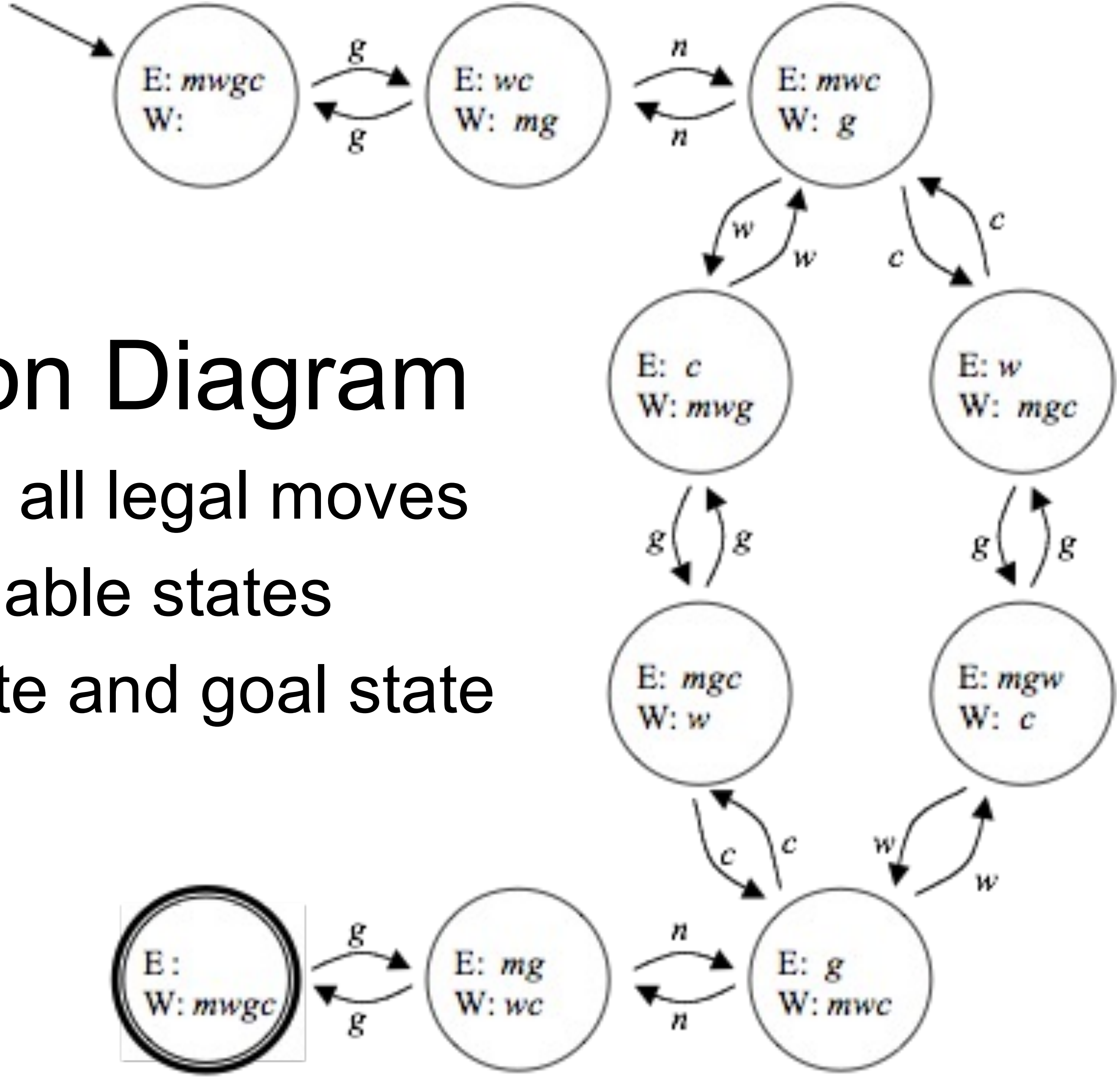
Moves As State Transitions

- Each move takes our puzzle universe from one state to another
- For example, the *g* move is a transition between these two states:



Transition Diagram

- Showing all legal moves
- All reachable states
- Start state and goal state



The Language Of Solutions

- Every path gives some $x \in \{w,g,c,n\}^*$

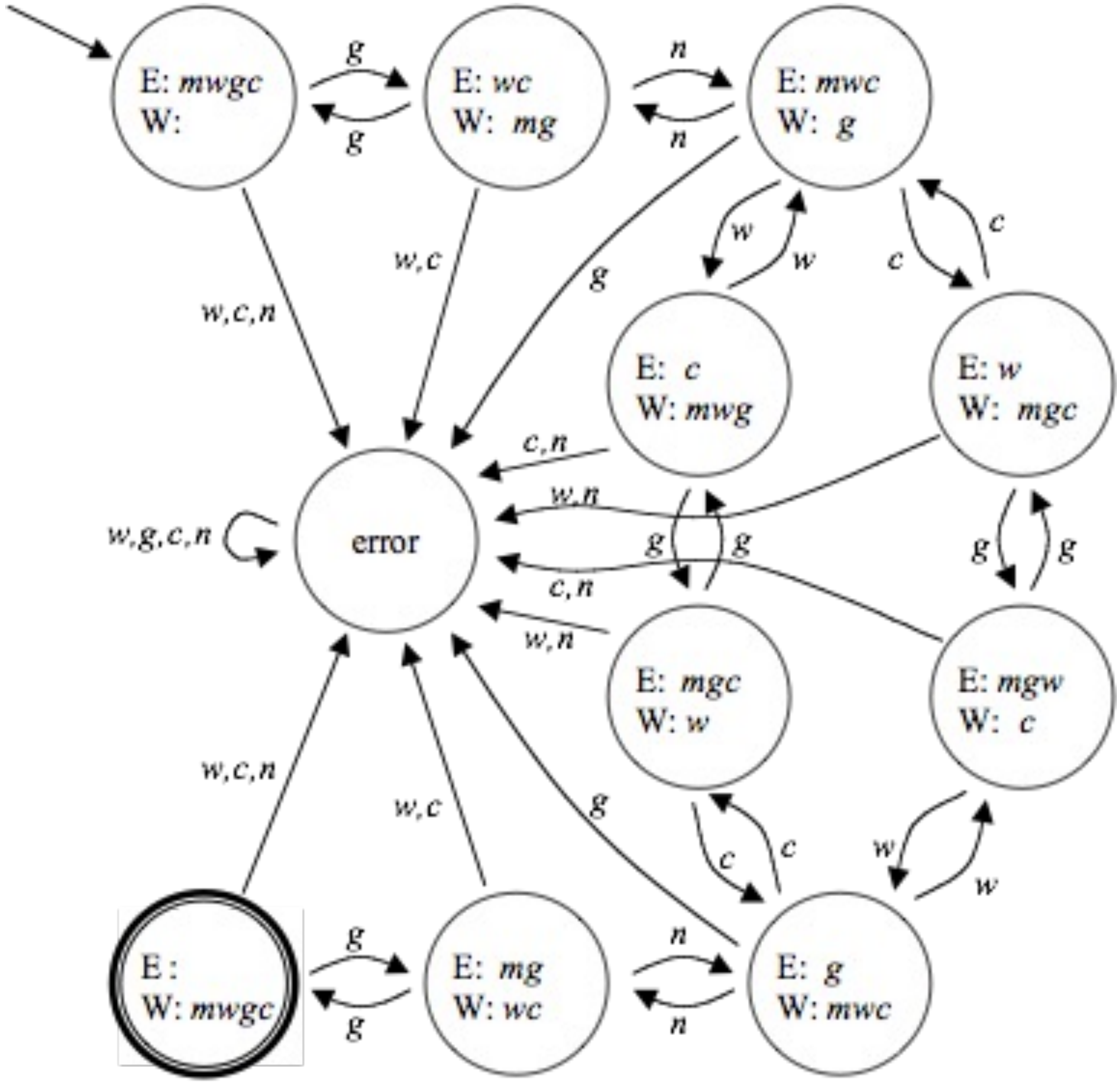
- The diagram defines the language of solutions to the problem:

  $\{x \in \{w,g,c,n\}^* \mid$ starts in start state and ends in goal state$\}$

- This is an infinite language
  - (The two shortest strings are *gnwgcng* and *gncgwng*)

Diagram Gets Stuck

- On many strings that are not solutions, the previous diagram gets stuck
- Automata that never get stuck are easier to work with
- We'll need one additional state to use when an error has been found in a solution


E: mwgc
W:
E: wc
W: mg
E: mwc
W: g
E: c
W: mwg
E: w
W: mgc
error
E: mgc
W: w
E: mgw
W: c
E :
W: mwgc
E: mg
W: wc
E: g
W: mwc
g
n
w
c
w,c,n
w,c
c,n
w,n
w,g,c,n

Complete Specification

- The diagram shows exactly one transition from every state on every symbol in Σ
- It gives a computational procedure for deciding whether a given string is a solution:
  - Start in the start state
  - Make one transition for each symbol in the string
  - If you end in the goal state, accept; if not, reject

Designing a DFA

- Think about how to identify words in the language, looking at input one char at a time
  - What do you have to remember?
- These become the states
- Then add transitions to update “memory”

Work design examples

Application

- Write a program to count "real" lines of Java code
  - Ignore blank lines, lines with only a comment (// or /* … */)
- Assume you can read the input one char at a time