# Finite State Machines and String Matching

# String Matching

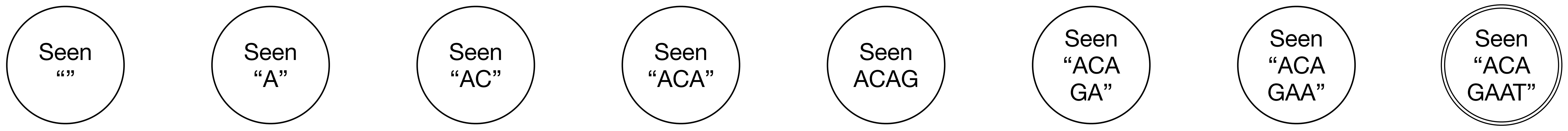
- Idea:
    - We look at each character in the pattern exactly once
    - We use a finite state machine to “remember” what we have matched
    - We create the finite state machine by pre-processing the pattern

# String Matching

- Example:
- Pattern is

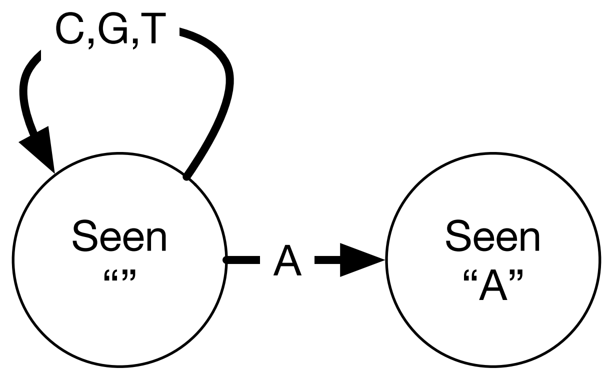
| | | | | | | |
|---|---|---|---|---|---|---|
| A | C | A | G | A | A | T |

- States corresponds to prefixes of the string



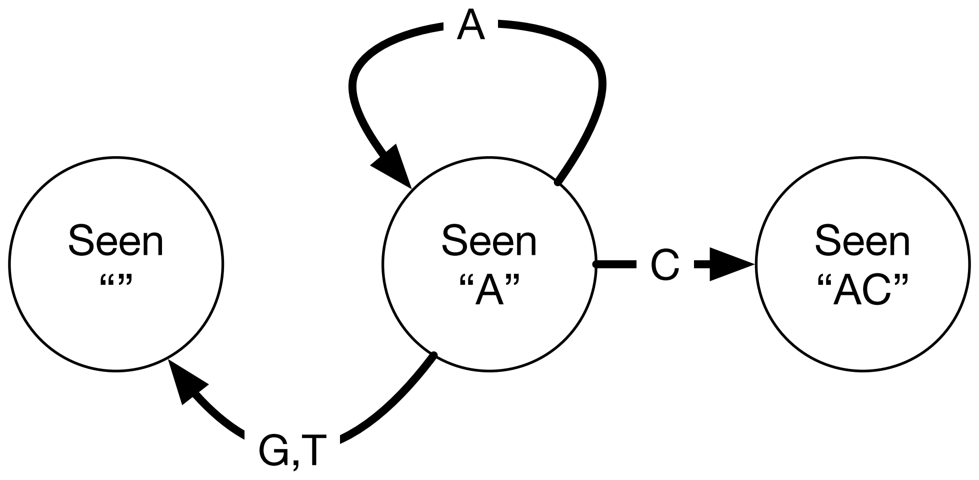
# String Matching

- State 0 corresponds to nothing currently matched
  - When we match an “A”, we move to the next State
  - Otherwise, we go back to the current state



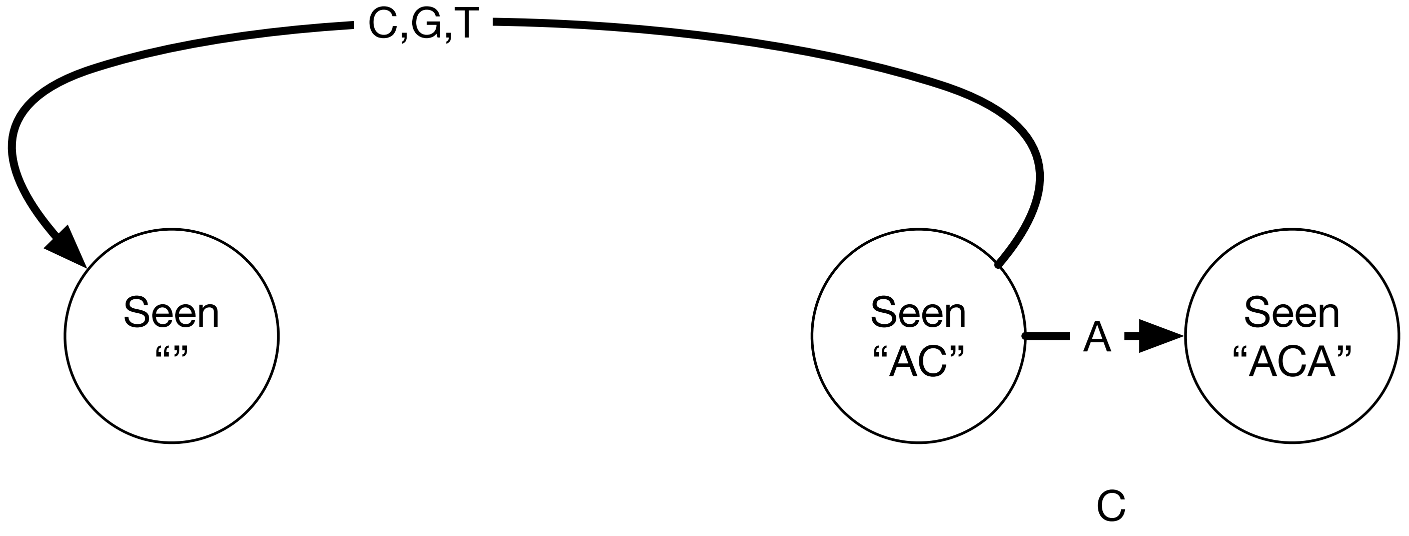
# String Matching

- In State “Seen A”:
  - If we have a “C”, go to next State
  - If we have a G or T, go back
  - If we have an “A”, we have seen an “AA” which has a suffix that partially matches



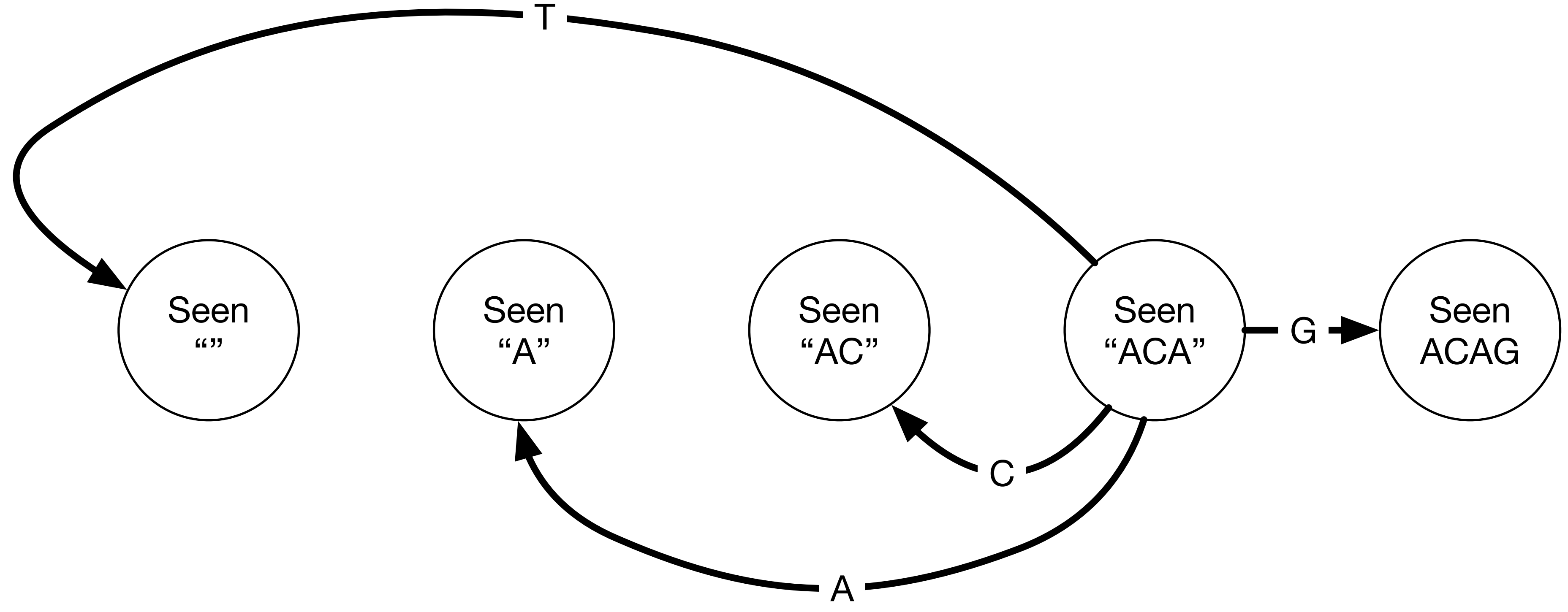
# String Matching

- State “AC”
    - “A” move to “ACA”
    - “C”, “G”, “T” move to “ ”



# String Matching

- State “ACA”
  - “C” matches “AC”
  - “T” matches “”
- State “ACA”
  - “G” matches next state
  - “A” matches “A”



# String Matching

- Your turn:  Quiz