Finite Automata

What is a Finite Automaton?

- A formal system.
- Remembers only a finite amount of information.
- Information represented by its *state*.
- State changes in response to *inputs*.
- Rules that tell how the state changes in response to inputs are called *transitions*.

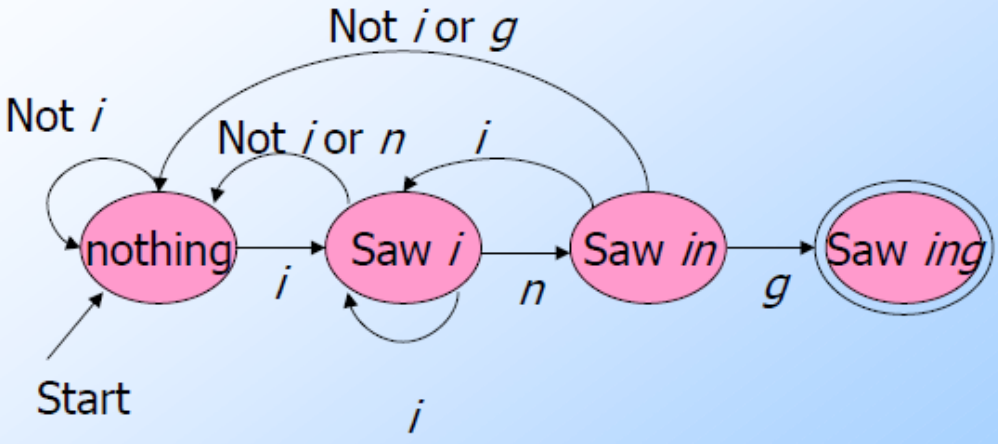
Informal Explanation

- Finite automata are finite collections of states with transition rules that take you from one state to another.
- Original application was sequential switching circuits, where the "state" was the settings of internal bits.
- Today, several kinds of software can be modeled by FA.

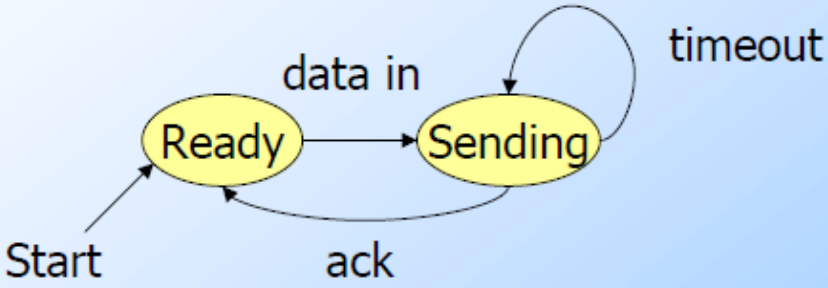
Representing FA

- Simplest representation is often a graph.
  - Nodes = states.
  - Arcs indicate state transitions.
  - Labels on arcs tell what causes the transition.

Example: Recognizing Strings Ending in "ing"



Example: Protocol for Sending Data

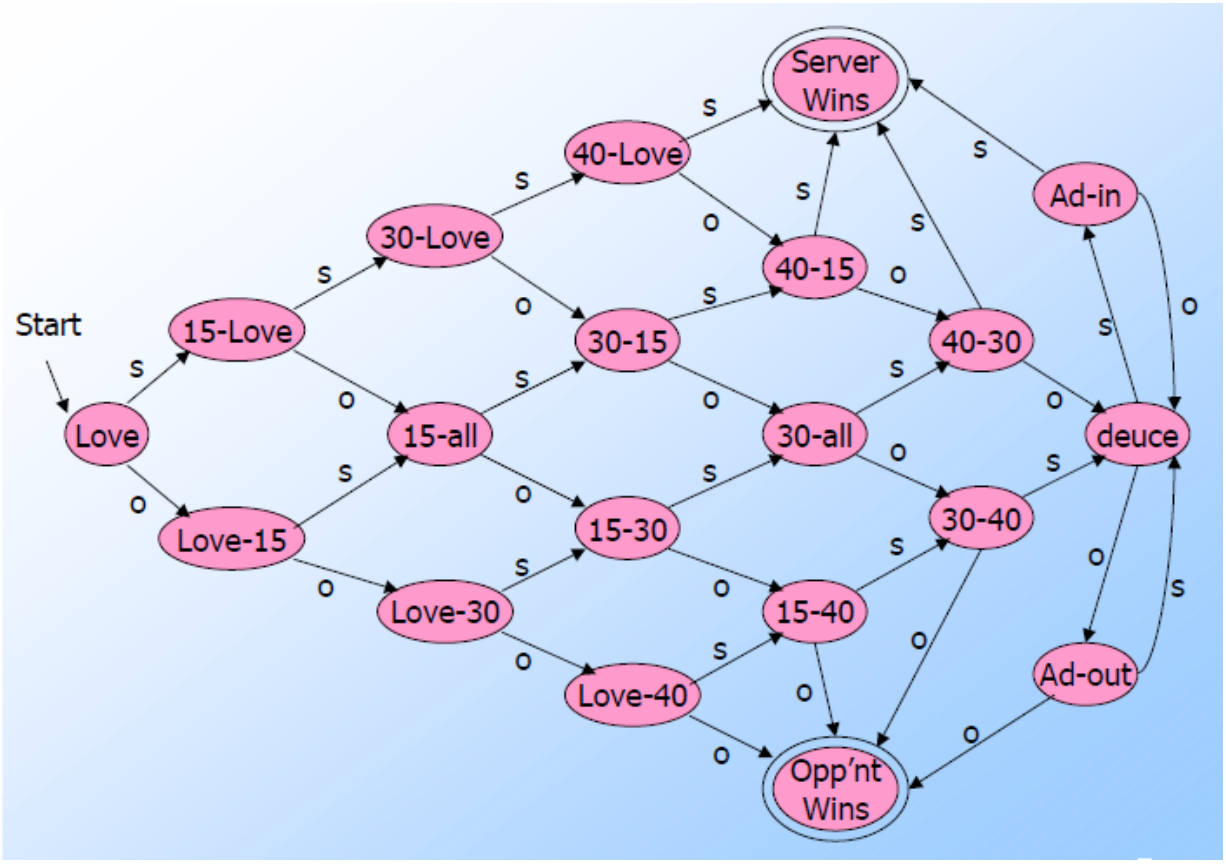


Tennis

- Like ping-pong, except you are very tiny and stand on the table.
- *Match* = 3-5 sets.
- *Set* = 6 or more games.

Scoring a Game

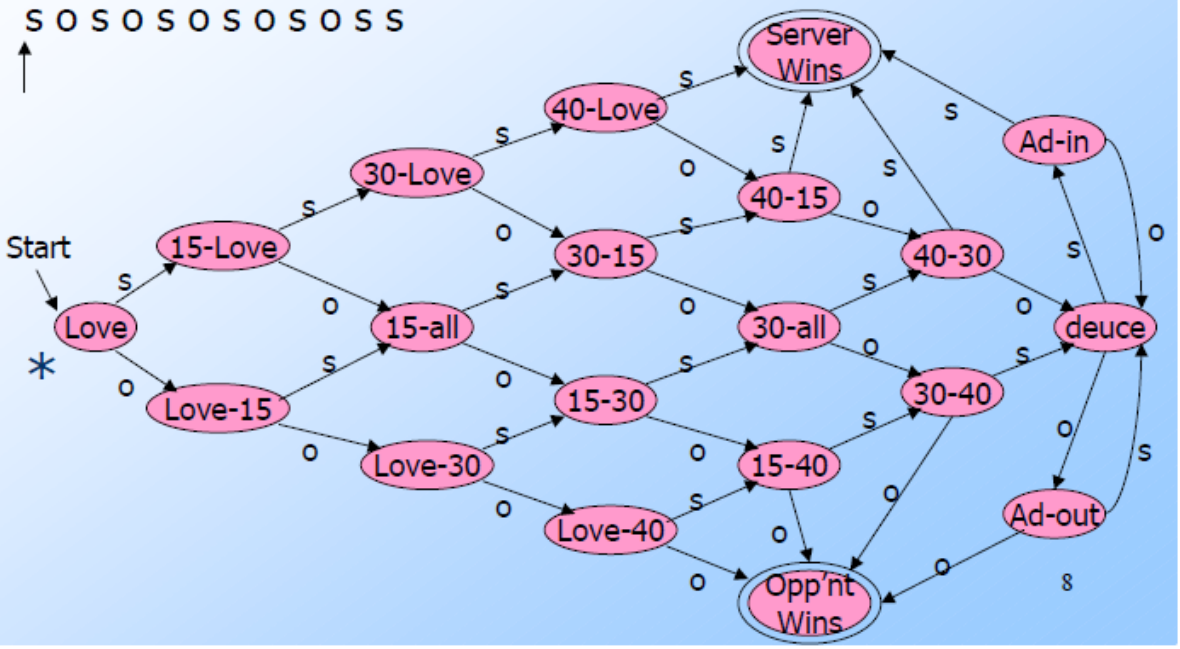
- One person serves throughout.
- To win, you must score at least 4 points.
- You also must win by at least 2 points.
- Inputs are s = "server wins point" and o = "opponent wins point."



Acceptance of Inputs

- Given a sequence of inputs (*input string*), start in the start state and follow the transition from each symbol in turn.
- Input is *accepted* if you wind up in a final (accepting) state after all inputs have been read.

Example: Processing a String



Language of an Automaton

- The set of strings accepted by an automaton A is the *language* of A.
- Denoted L(A).
- Different sets of final states -> different languages.
- Example: As designed, L(Tennis) = strings that determine the winner.