Automata Theory

Why Study Automata?

What the Course is About

Why Study Automata?

- A survey of Stanford grads 5 years out asked which of their courses did they use in their job.
- Basics like intro-programming took the top spots, of course.
- But among optional courses, CS154 stood remarkably high.
  - 3X the score for AI, for example.

How Could That Be?

- Regular expressions are used in many systems.
  - E.g., UNIX a.*b.
  - E.g., DTD's describe XML tags with a RE format like person (name, addr, child*).
- Finite automata model protocols, electronic circuits.

How? – (2)

- Context-free grammars are used to describe the syntax of essentially every programming language.
  - Not to forget their important role in describing natural languages.
- And DTD's taken as a whole, are really CFG's.

How? – (3)

- When developing solutions to real problems, we often confront the limitations of what software can do.
  - *Undecidable* things – no program whatever can do it.
  - *Intractable* things – there are programs, but no fast programs.
- Automata theory gives you the tools.

Other Good Stuff

- We'll learn how to deal formally with discrete systems.
  - Proofs: You never really prove a program correct, but you need to be thinking of why a tricky technique really works.
- We'll gain experience with abstract models and constructions.
  - Models layered software architectures.

Automata Theory – Gateway Drug

- This theory has attracted people of a mathematical bent to CS, to the betterment of all.
  - Ken Thompson – before UNIX was working on compiling regular expressions.
  - Jim Gray – thesis was automata theory before he got into database systems and made fundamental contributions there.

Course Outline

- Regular Languages and their descriptors:
  - Finite automata, nondeterministic finite automata, regular expressions.
  - Algorithms to decide questions about regular languages, e.g., is it empty?
  - Closure properties of regular languages.

Course Outline – (2)

- Context-free languages and their descriptors:
  - Context-free grammars, pushdown automata.
  - Decision and closure properties.

Course Outline – (3)

- Recursive and recursively enumerable languages.
  - Turing machines, decidability of problems.
  - The limit of what can be computed.
- Intractable problems.
  - Problems that (appear to) require exponential time.
  - NP-completeness and beyond.

Finite Automata

What Are They?

Who Needs 'em?

An Example: Scoring in Tennis

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*.

Why Study Finite Automata?

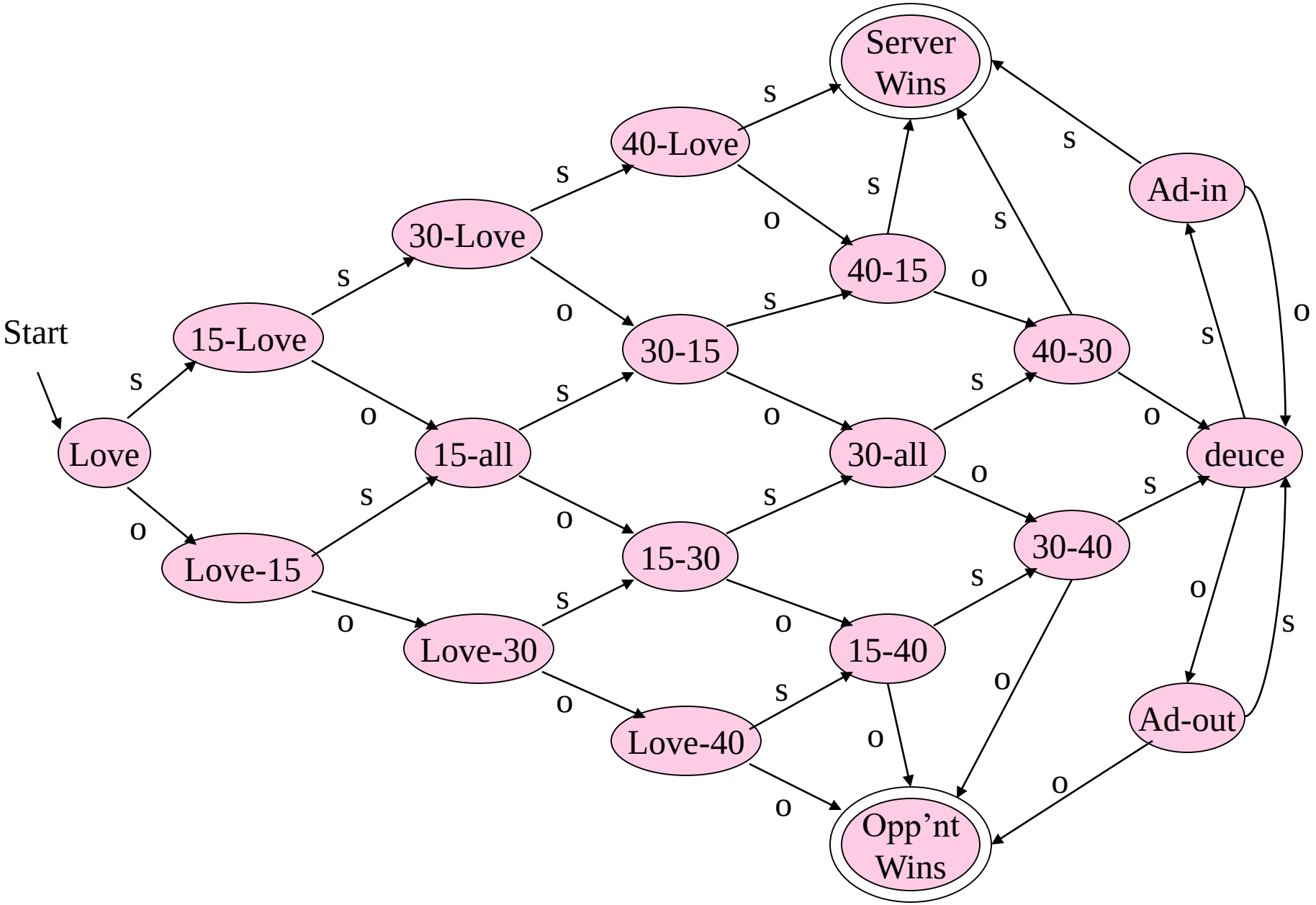
- Used for both design and verification of circuits and communication protocols.
- Used for many text-processing applications.
- An important component of compilers.
- Describes simple patterns of events, etc.

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

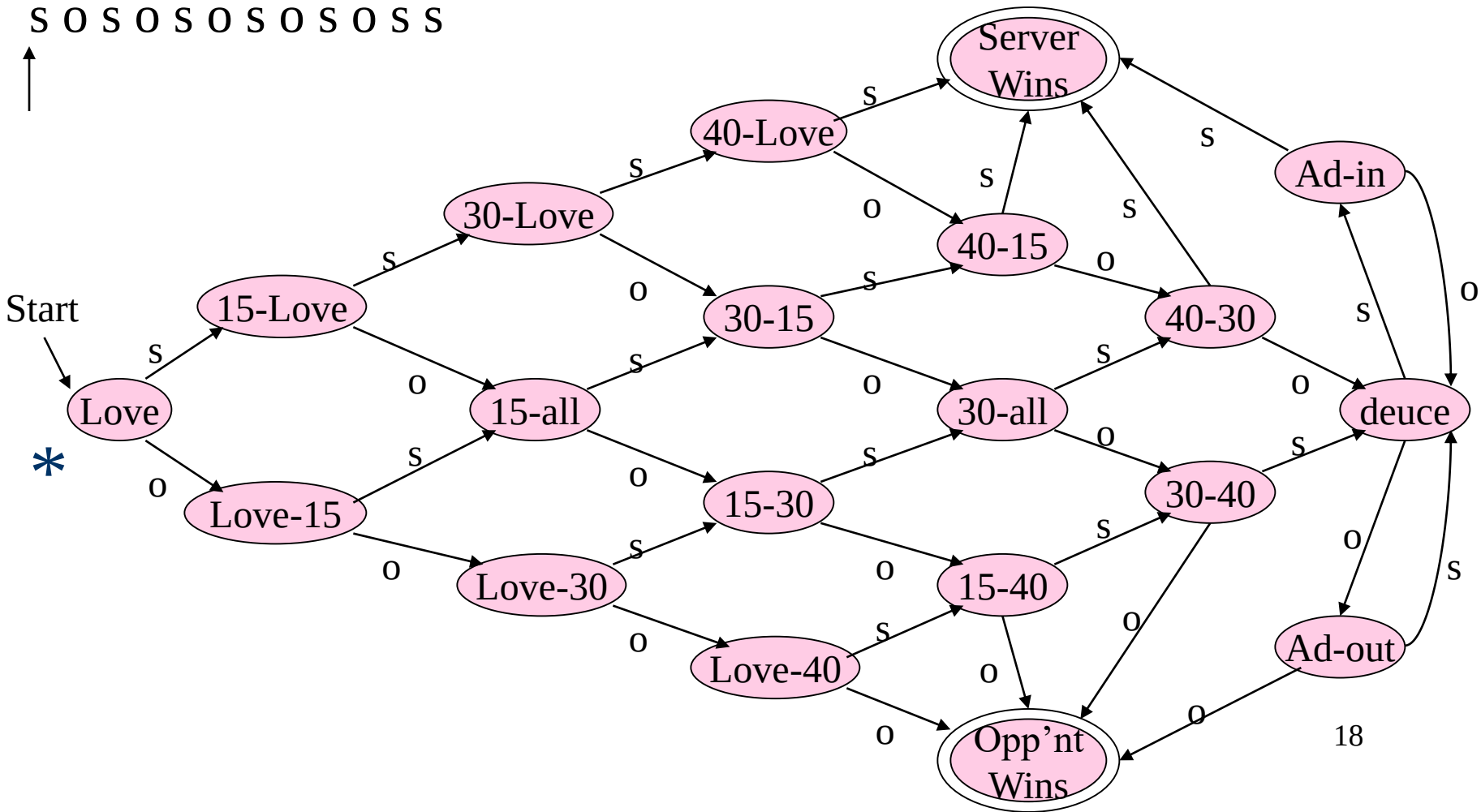
- One person serves throughout.
- To win, you must score at least 4 points.
- You also must win by at least 2 points.
- Inputs are s = “server wins point” and o = “opponent wins point.”


Start
Love
15-Love
30-Love
40-Love
Server Wins
Love-15
15-all
30-15
40-15
Ad-in
Love-30
15-30
30-all
40-30
deuce
Love-40
15-40
30-40
Ad-out
Opp'nt Wins
s
o

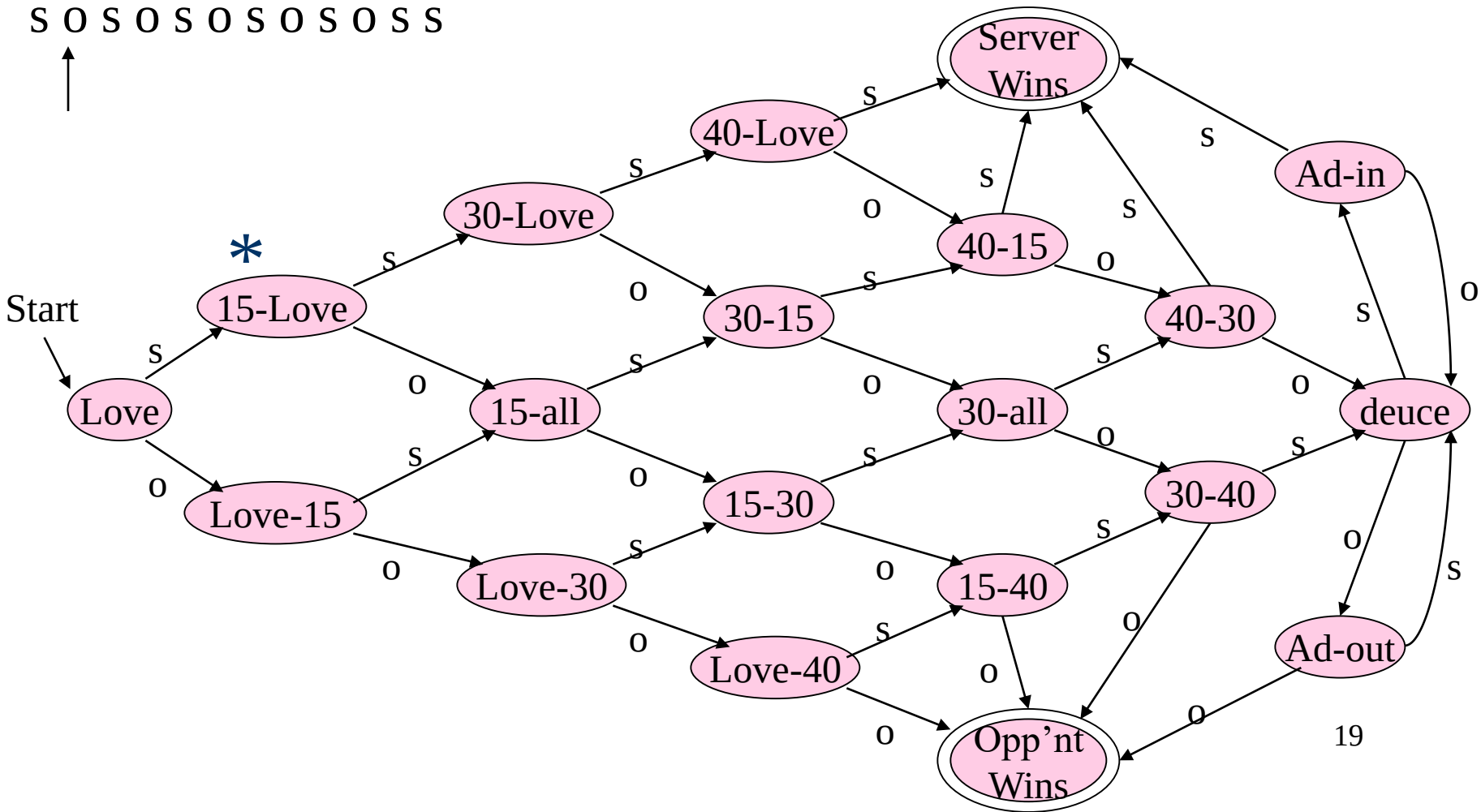
Acceptance of Inputs

- 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.

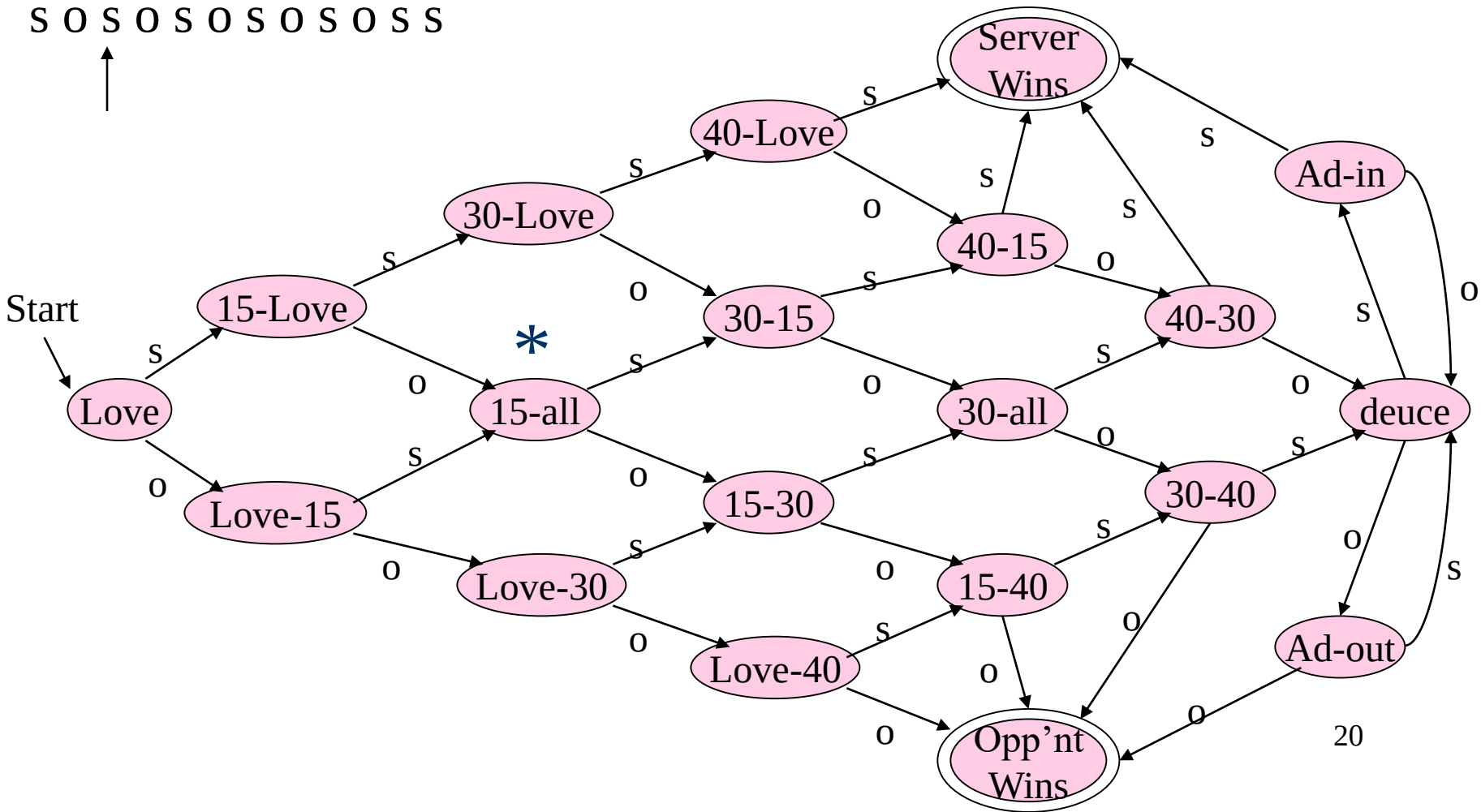
Example: Processing a String



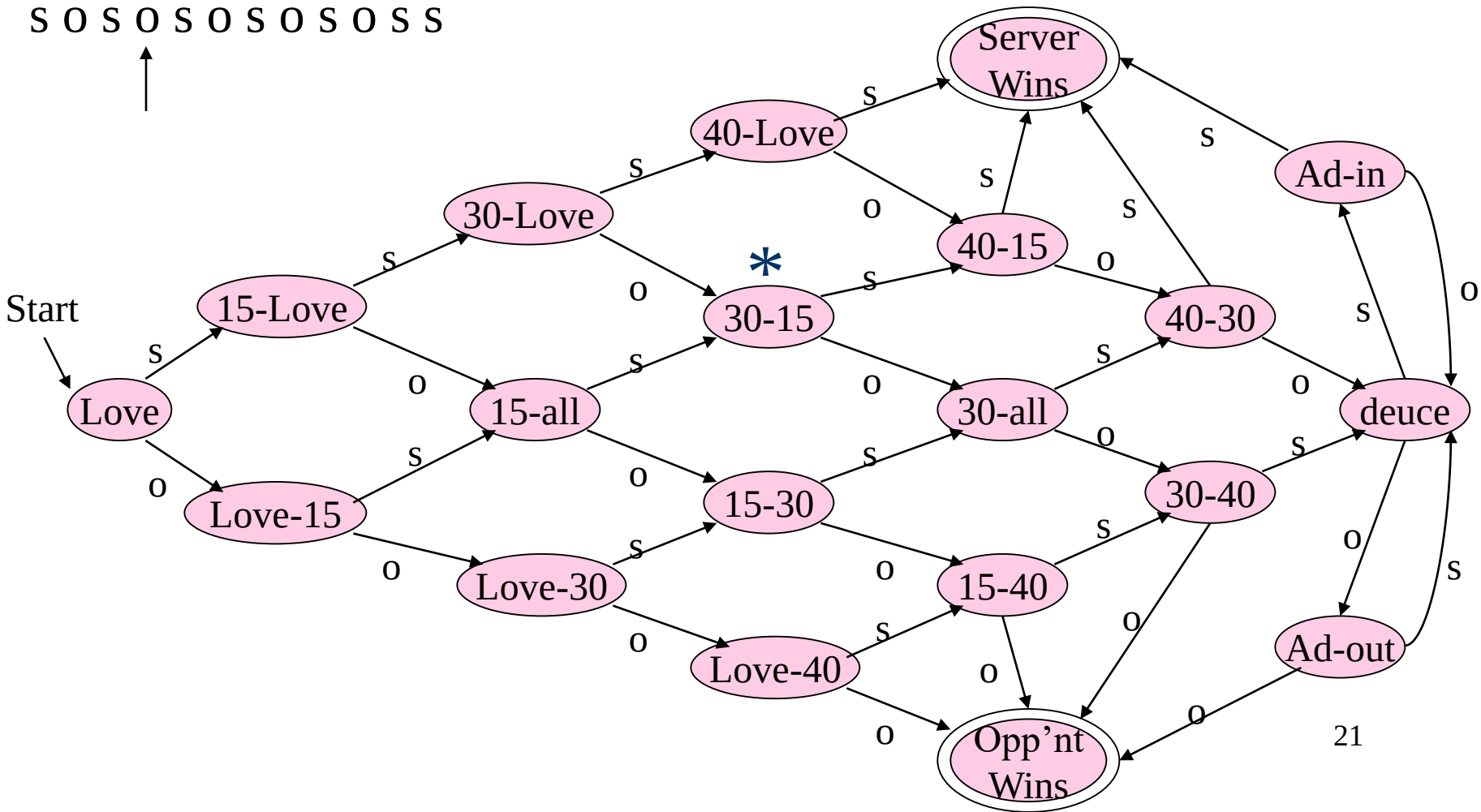
Example: Processing a String



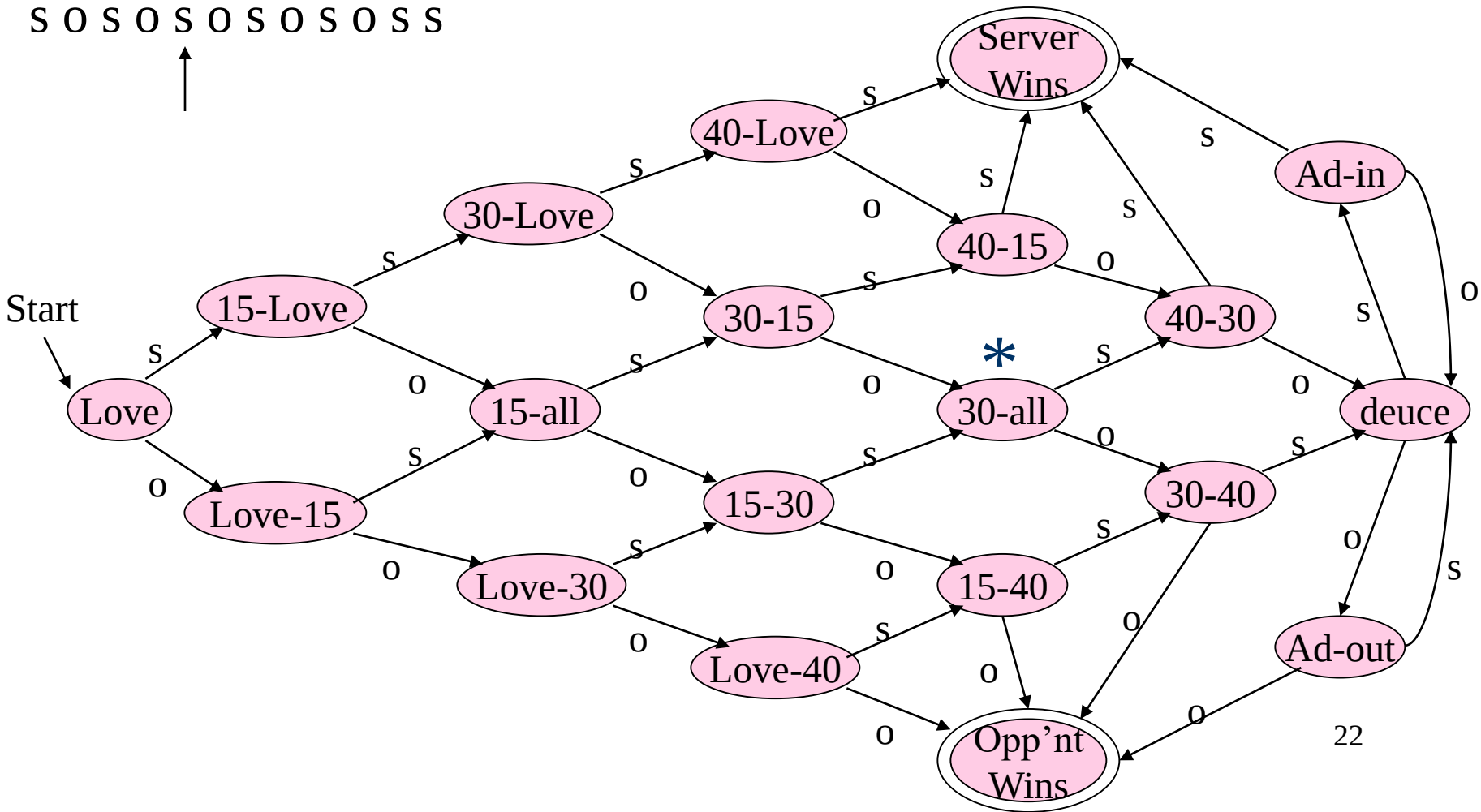
Example: Processing a String



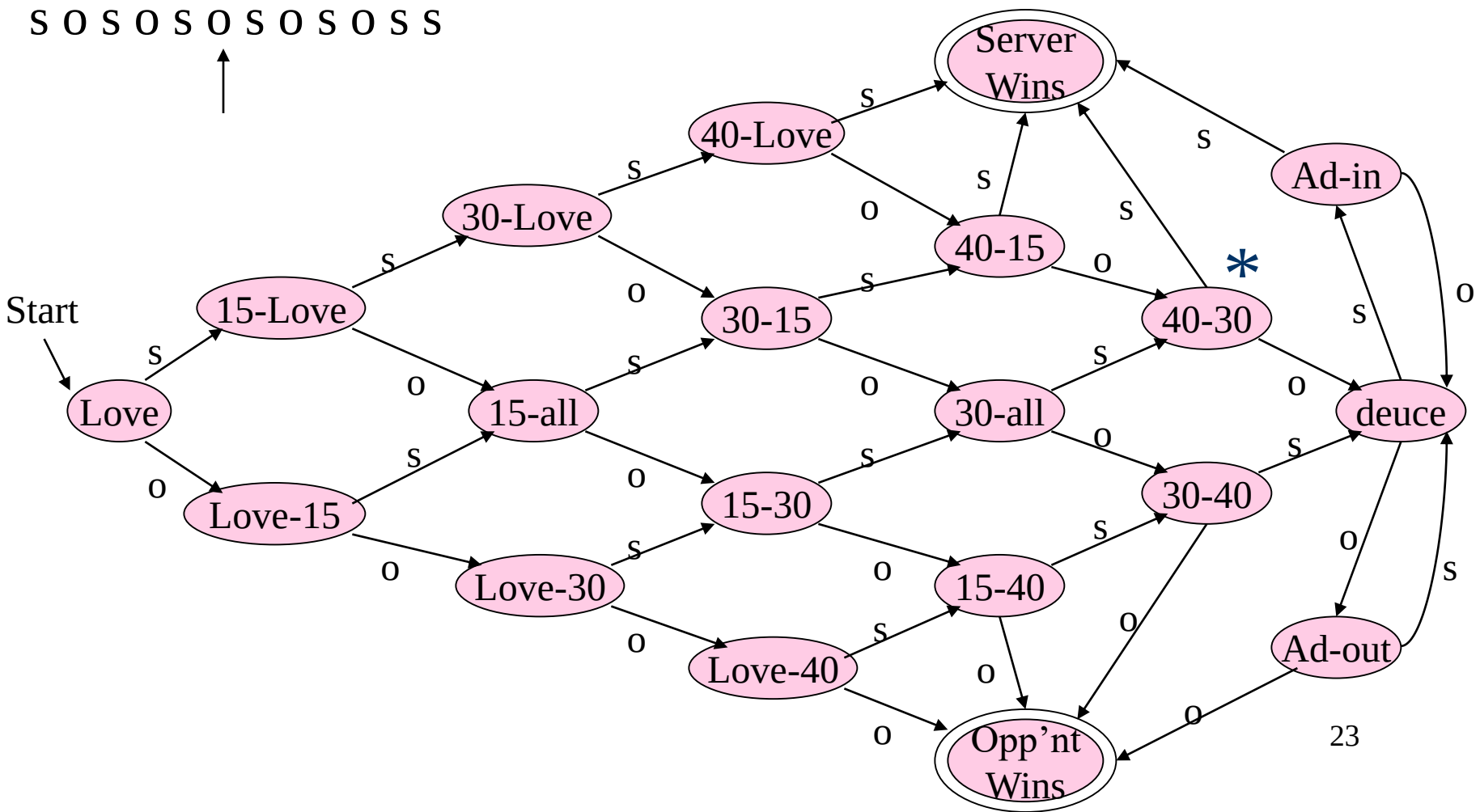
Example: Processing a String



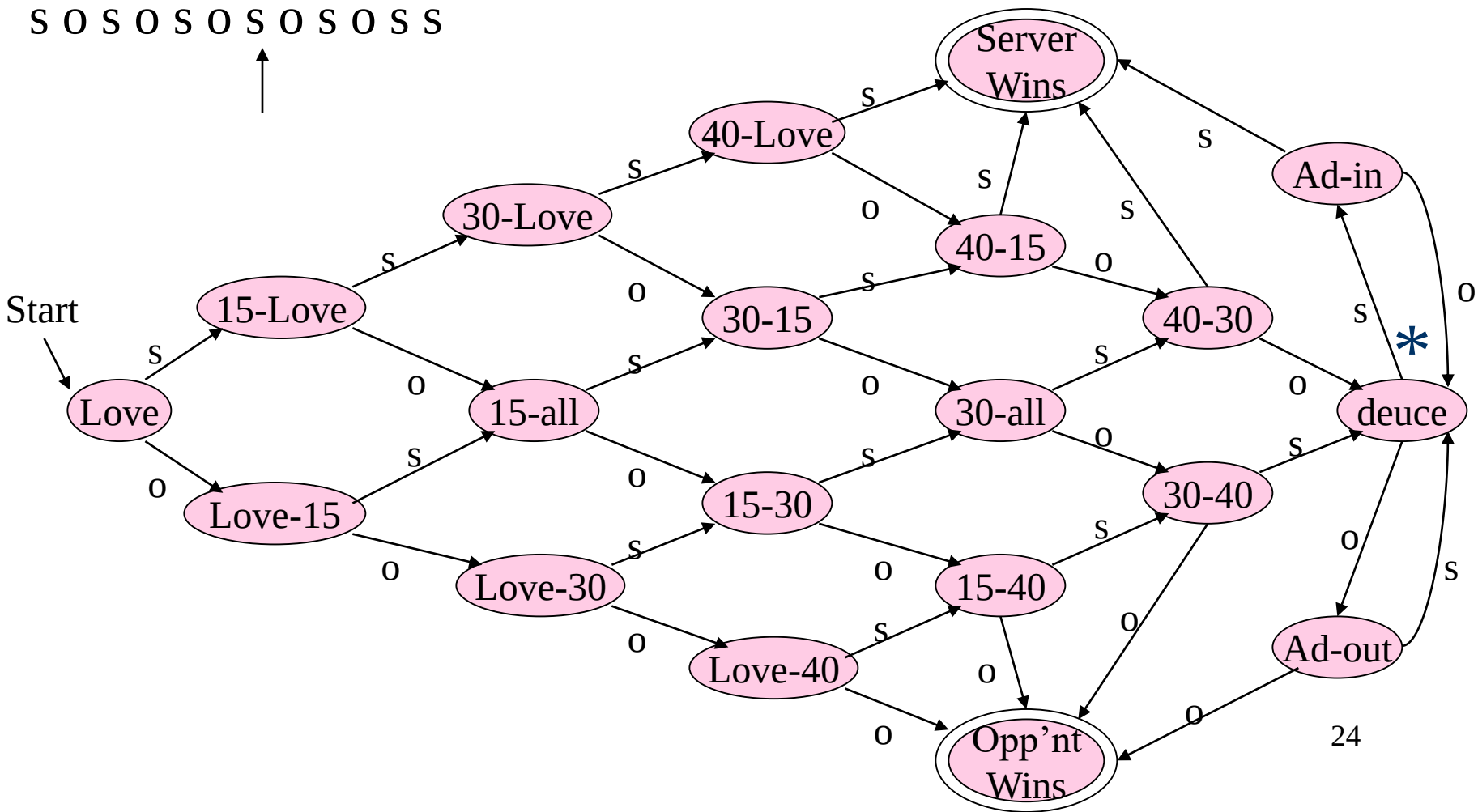
Example: Processing a String



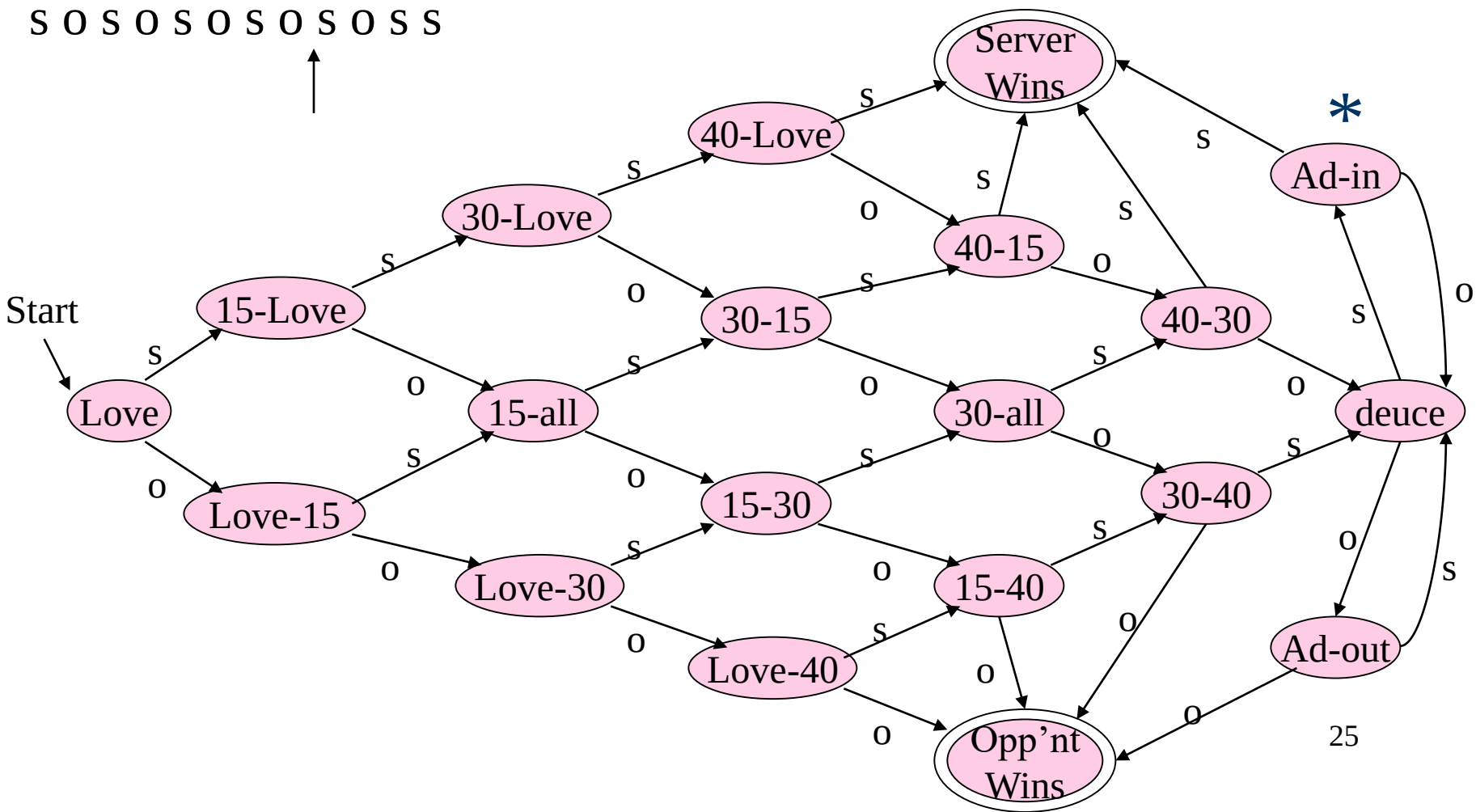
Example: Processing a String



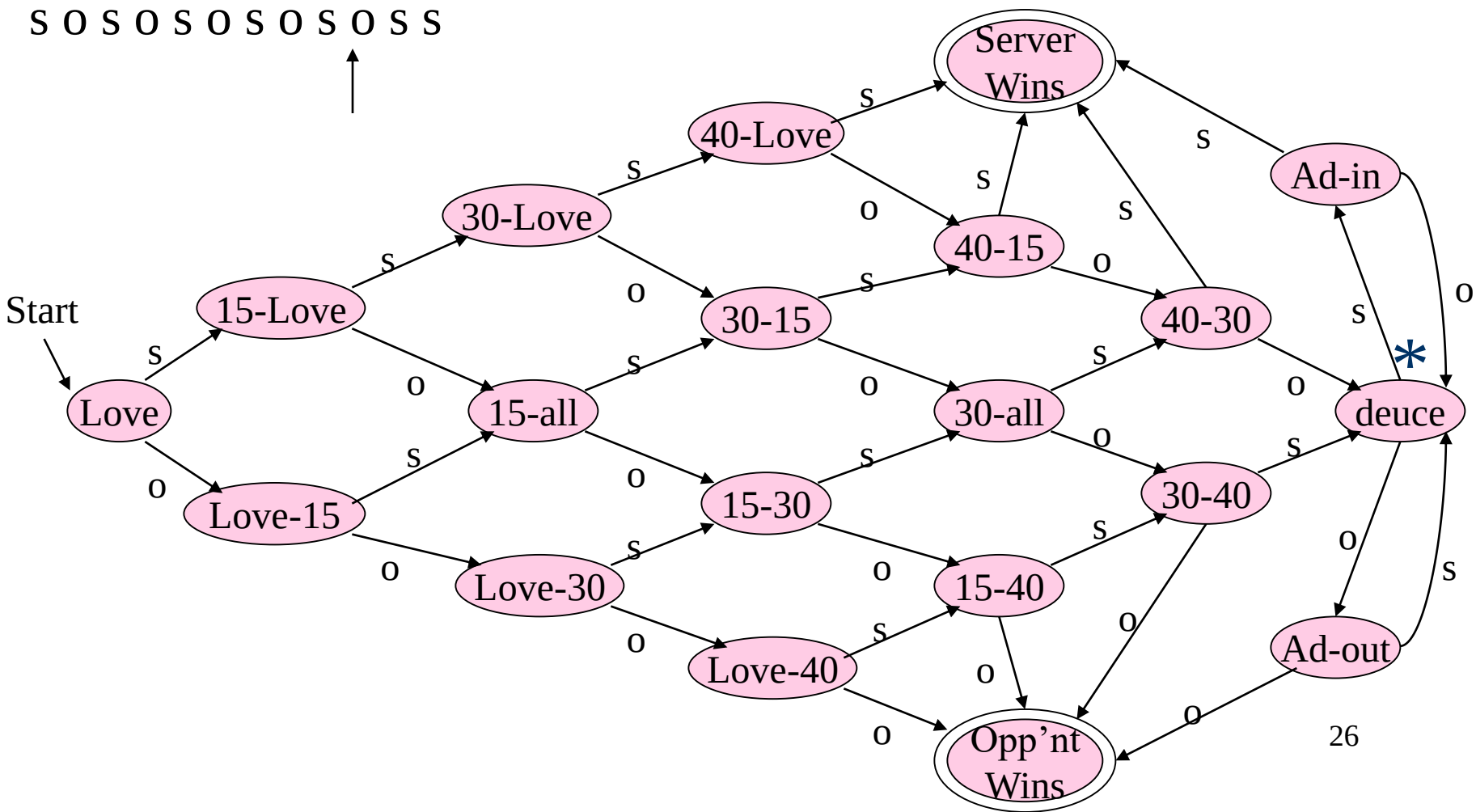
Example: Processing a String



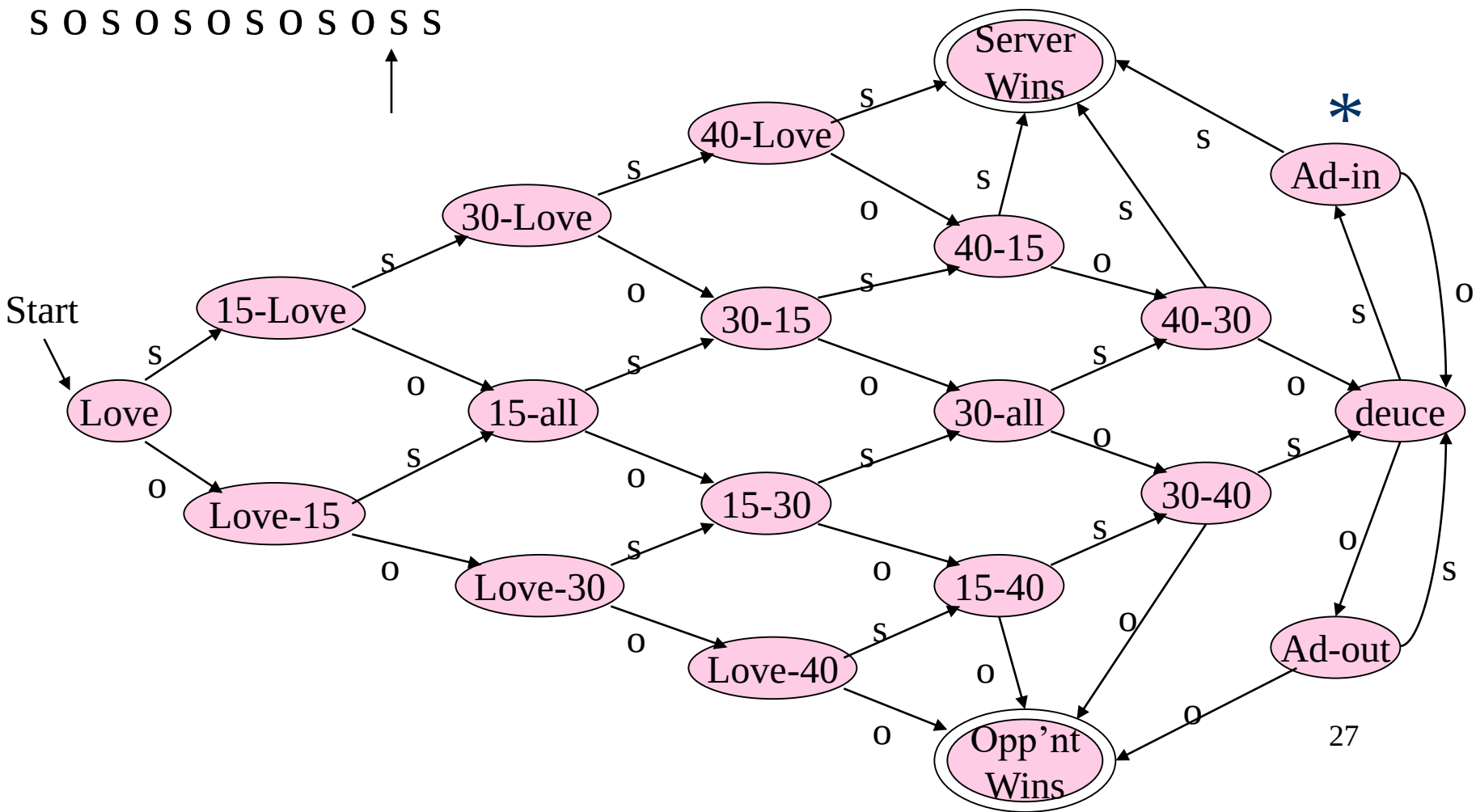
Example: Processing a String



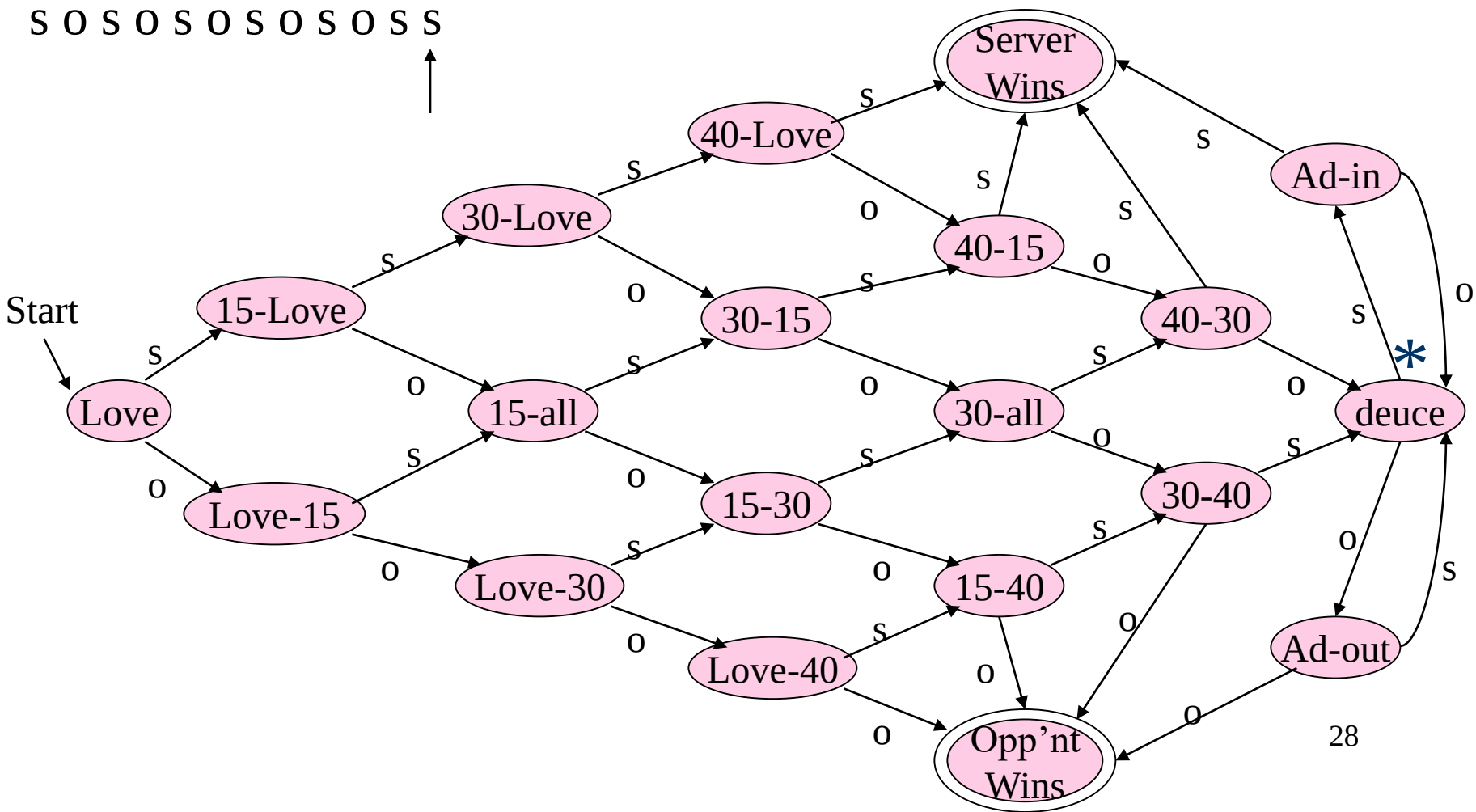
Example: Processing a String



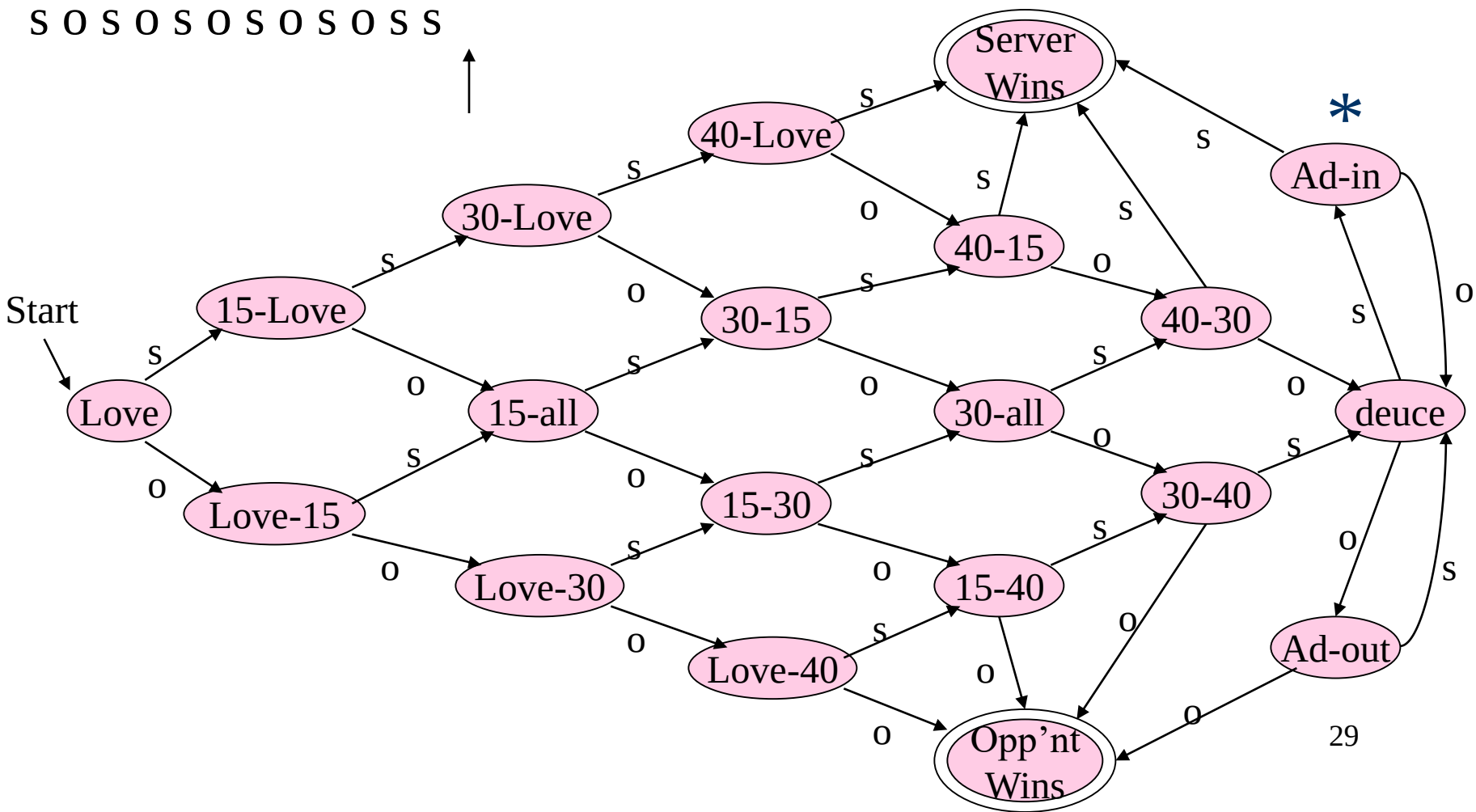
Example: Processing a String



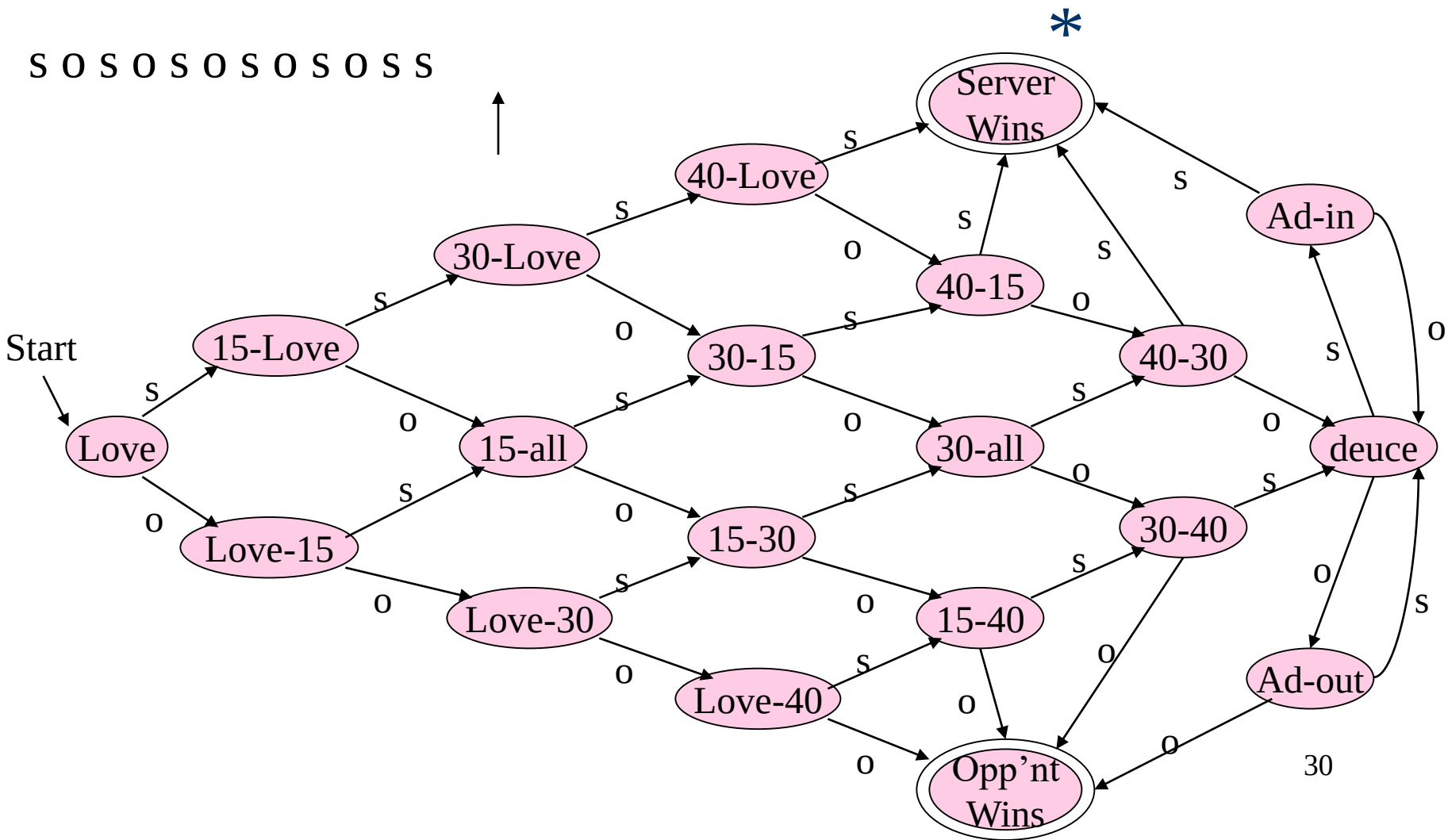
Example: Processing a String



Example: Processing a String



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.

Text (Not Required)

- Hopcroft, Motwani, Ullman, *Automata Theory, Languages, and Computation* 3rd Edition.
- Course covers essentially the entire book.