

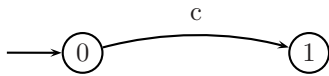
Finite-State Automata

Linguistics 384

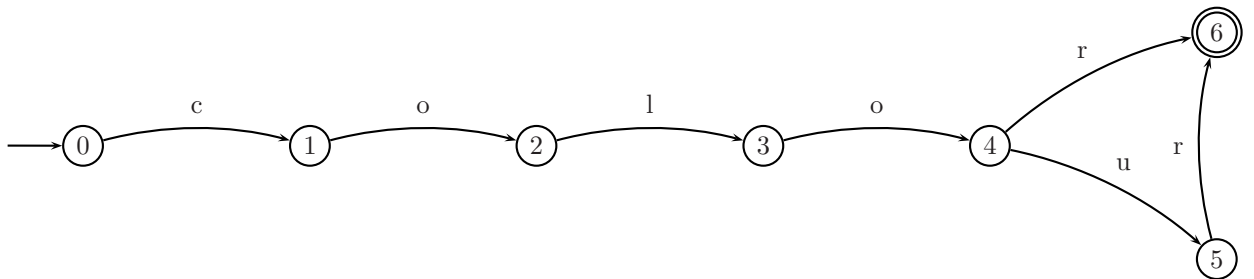
Spring 2004

A **finite-state automaton** (FSA)—plural = automata—is a series of **states** and **transitions** between states. (Also called a **finite-state machine**.)

- By moving through the FSA, you try to match an input string.
- The idea is that you can move from state to state, when conditions on the arcs are met.
e.g. Move from state 0 to state 1 if I encounter the letter *c*. (Otherwise, do nothing.)

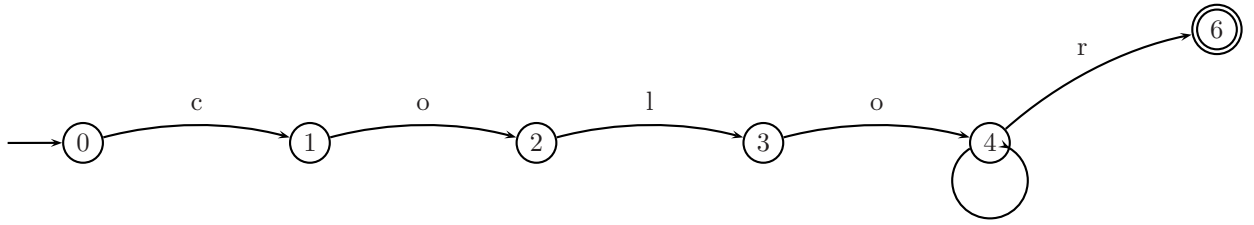


We represent states with **nodes** (circles) and transitions with **arcs** (arrows) in a picture. (Note that the numbers on the nodes are just for our convenience; they don't really “mean” anything.)

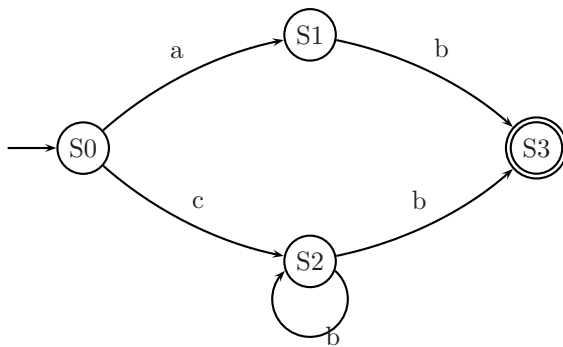


The FSA above matches (or generates) the string *color* or the string *colour*. The double lines around node 6 means that this is a **final state** = can end at this state. i.e. If you match *colo*, you are not done. (you have to get to a double circle or it fails.)

Now, the following FSA is similar, but subtly different. How is it different?



What does the following FSA do?



Note:

- You have to have a single start node, but you can have more than one end node.
- The nodes don't necessarily have to go in order from left to right.
- FSAs are what are used to match regular expressions. Logically, they are equivalent.

Now draw three FSAs which recognize the following regular expressions:

- aem
- aer*m
- a(b|p)er*m