Finite State Machines

Dr. Carmen Bruni `cbruni@uwaterloo.ca`

David R. Cheriton School of Computer Science
University of Waterloo
`https://cs.uwaterloo.ca/~cbruni/`
With thanks to Troy Vasiga `troy.vasiga@uwaterloo.ca`

Wednesday May 15th, 2024

A Puzzle - The Wolf Cabbage and Goat problem

You can play this on the *River Tests* app or at the link below (change cabbage to carrot):
`https://www.transum.org/software/River_Crossing/`

Solution

What does this have to do with computer science?

Let's try to represent this using a diagram.

Finite State Machines

- You've just created your first finite state machine!
- But what does this have to do with computer science?
- ... and what really is a finite state machine?

Finite State Machines

Finite state machines model...

- ...what we know about the world
- ...what changes in our world with input

Examples of Finite State Machines

Finite State Machine Definition

- A finite state machine consists of:
  - A finite set of states
  - A finite set of characters belonging to an **alphabet**
  - Transitions between states using the elements of the alphabet
  - A starting state
  - An accepting state (could be 0, 1, or many such states)

What is a state?

A description of what we know:

What is Input? First Alphabets!

An alphabet is a finite set of symbols

- Alphabets usually denoted by Σ (subscripted if there are multiple alphabets).
- Examples:
  - $\Sigma_1 = \{0, 1\}$
  - $\Sigma_2 = \{a, b, ..., z\}$
  - $\Sigma_3 = \{happy, t,$  $\}$

What is Input?

Consists of elements from our alphabet. Comprise our transitions.
Let $\Sigma = \{$bad mark, stub toe, free pie, clap your hands$\}$.

What is a Starting State?

Where to begin

Final Accepting States

How do we know what is accepted/good?

Words

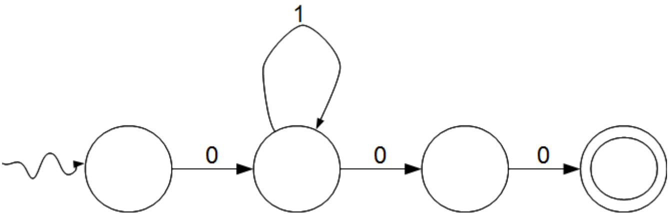
A **word** consists of a finite sequence of symbols from our finite alphabet.

Special word λ is the *empty word*.

Recall: $\Sigma_1 = \{0, 1\}$, $\Sigma_2 = \{a, b, ..., z\}$, $\Sigma_3 = \{happy, t,$  $\}$

Example

What words do the following FSM accept over the alphabet $\Sigma = \{0, 1\}$? Can we characterize these words nicely?



Regular Expressions

We can express the collection of words accepted by the machine on the previous slide as 01^*00.

Regular Expressions Examples

$(ab)^*$

$(a|b)^*$

Another Example

Write an FSM that accepts all words over $\Sigma = \{a, b, c\}$ that contain the word *abc* inside.

DFA

NFA

λ NFA

What about a FSM that accepts words of the form $((a^*b^*)|(ab)^*)$?

Reminder: Concatenation has precedence before alternation - similar to how multiplication has precedence over addition.
Hint: Think of λ as the glue that glues machines together!

λ NFA Solution

Summary

- Finite state machines helps us to model the world.
- Consist of states and transitions between them.
- Collections of words accepted by a DFA/NFA/λ-NFA can be written using a regular expression (This isn't obvious but is true!)