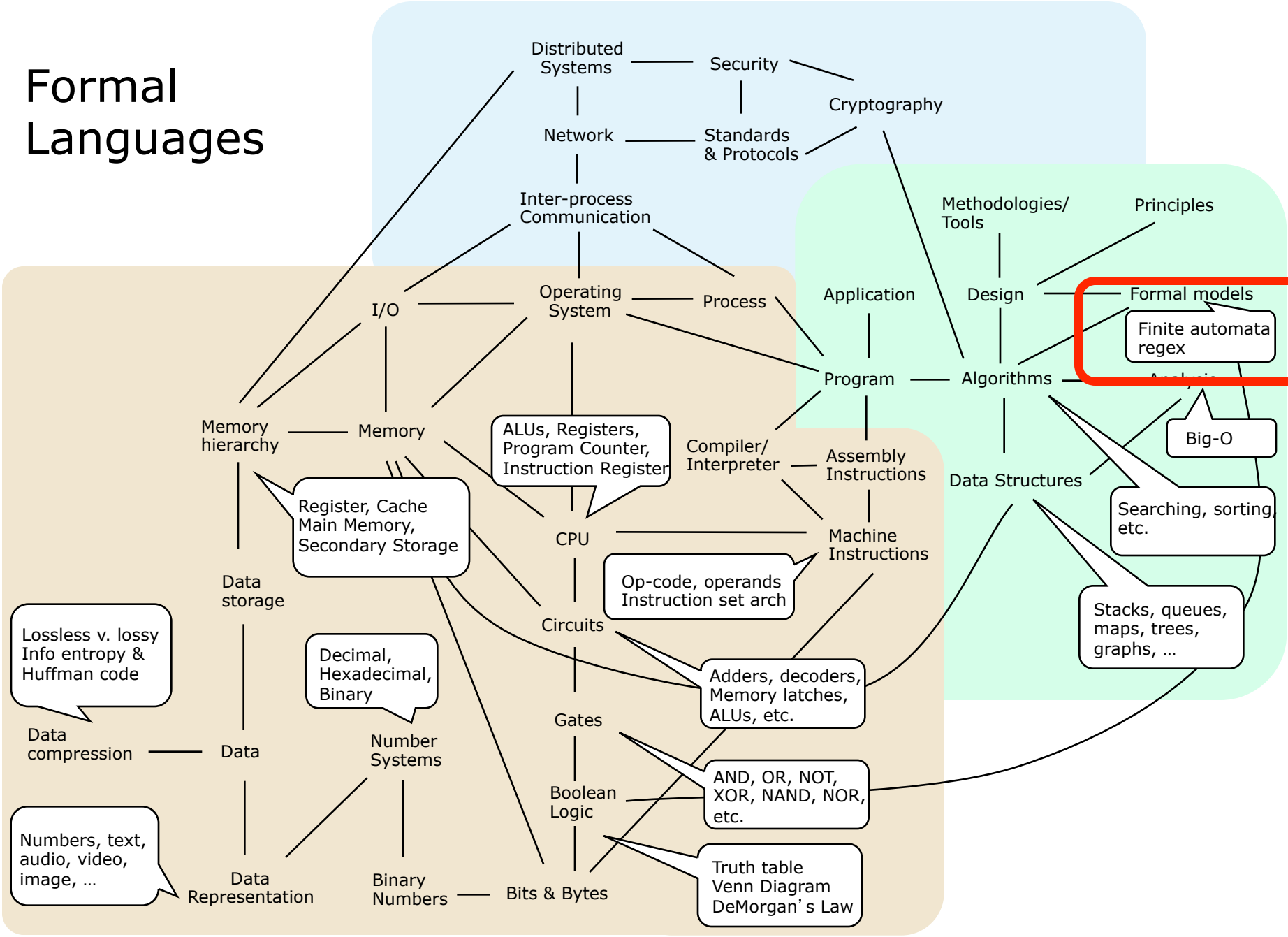
i206: Lecture 16: Finite Automata & Regular Expressions

Tapan Parikh

Spring 2013

Some slides courtesy Marti Hearst, John Chuang and others

Formal Languages



Theories of Computation

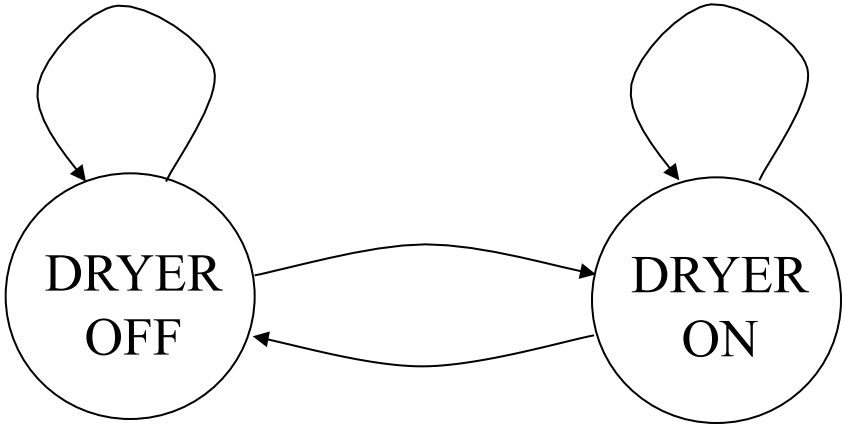
- What are the fundamental capabilities and limitations of computers?
  - Complexity theory: what makes some problems computationally hard and others easy?
  - Automata theory: definitions and properties of mathematical models of computation
  - Computability theory: what makes some problems solvable and others unsolvable?

Finite Automata

- Also known as finite state automata (FSA) or finite state machine (FSM)
- A simple mathematical model of a computer
- Applications: hardware design, compiler design, text processing

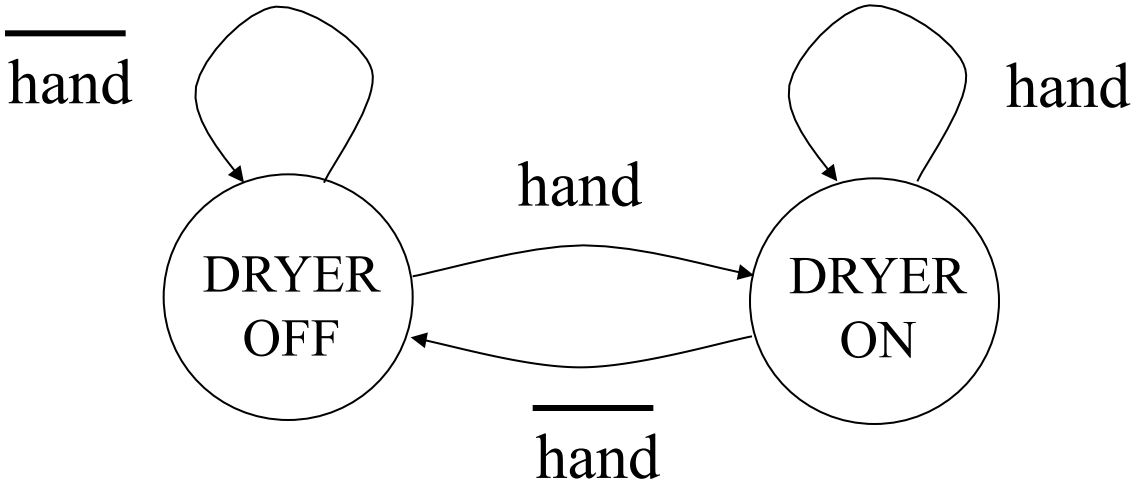
A First Example

- Touch-less hand-dryer



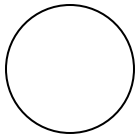
A First Example

- Touch-less hand-dryer

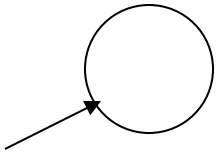


Finite Automata State Graphs

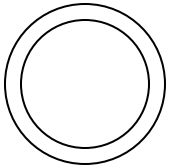
- A state



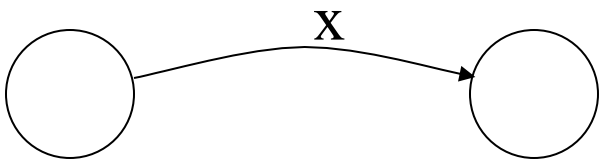
- The start state



- An accepting state



- A transition

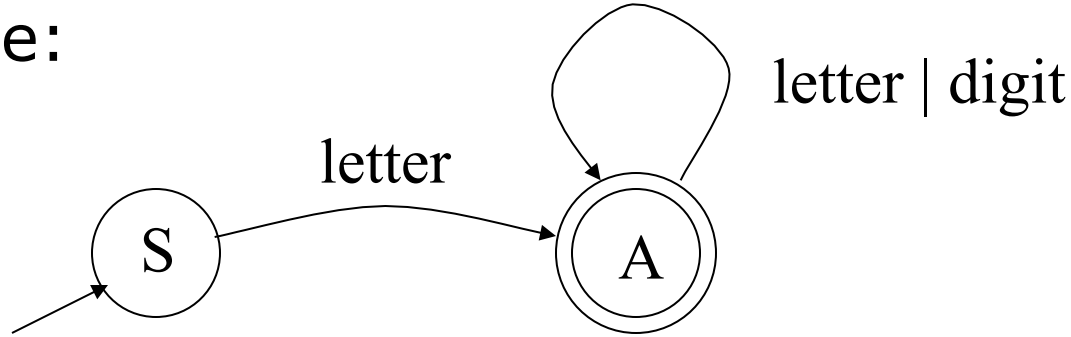


Finite Automata

- Transition: $s_1 \rightarrow^x s_2$
  - In state s_1 on input "x" go to state s_2
- If end of input
  - If in accepting state => *accept*
  - Else => *reject*
- If no transition possible => reject

Language of a FA

- Language of finite automaton M: set of all strings accepted by M
- Example:

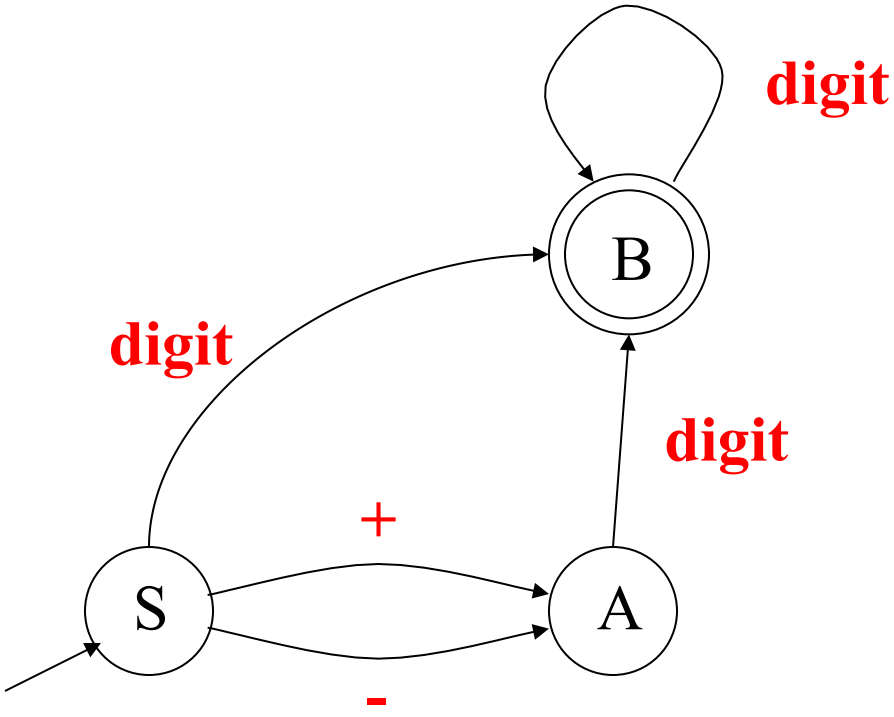


- Which of the following are in the language?
  - x, tmp2, 123, a?, 2apples

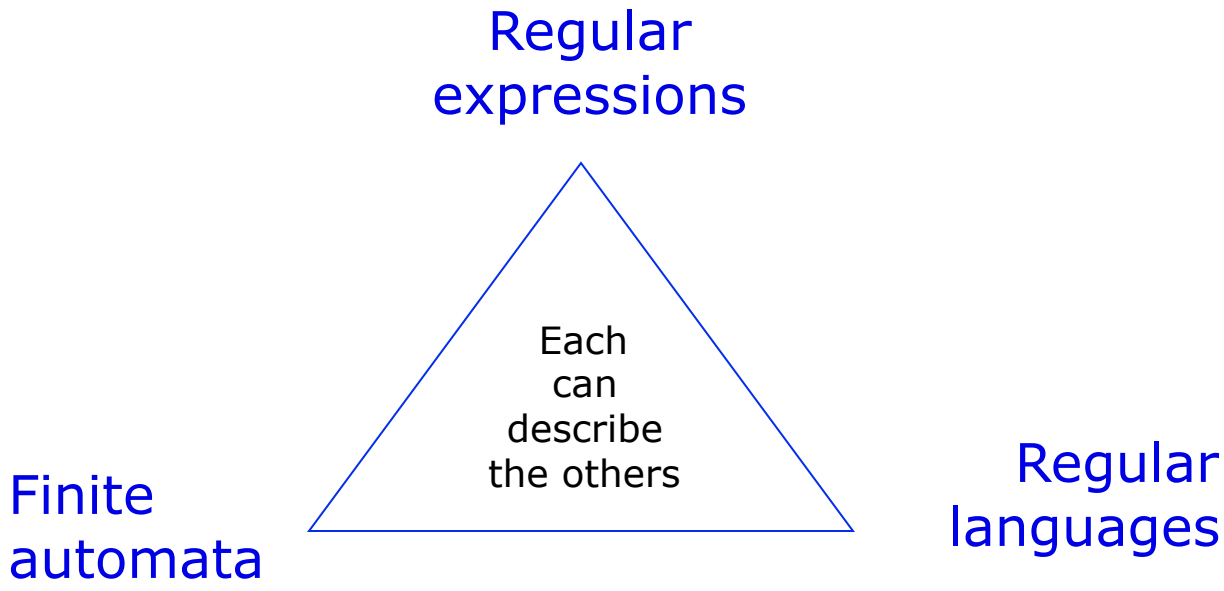
- A language is called a regular language if it is recognized by some finite automaton

Example

- What is the language of this FA?



Three Equivalent Representations



Theorem:

For every regular expression, there is a deterministic finite-state automaton that defines the same language, and vice versa.


WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.
OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!
BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!
IT'S HOPELESS!
EVERYBODY STAND BACK.
I KNOW REGULAR EXPRESSIONS.
PERL!
TAP TAP

Regex Rules

- ? Zero or one occurrences of the preceding character/regex

woodchucks?	"how much wood does a woodchuck chuck?"
behaviou?r	"behaviour is the British spelling of behavior"

- * Zero or more occurrences of the preceding character/regex

baa*	ba, baa, baaa, baaaa …
ba*	b, ba, baa, baaa, baaaa …
[ab]*	ε, a, b, ab, ba, baaa, aaabbb, …
[0-9][0-9]*	any positive integer, or zero
cat.*cat	A string where "cat" appears twice anywhere

- + One or more occurrences of the preceding character/regex

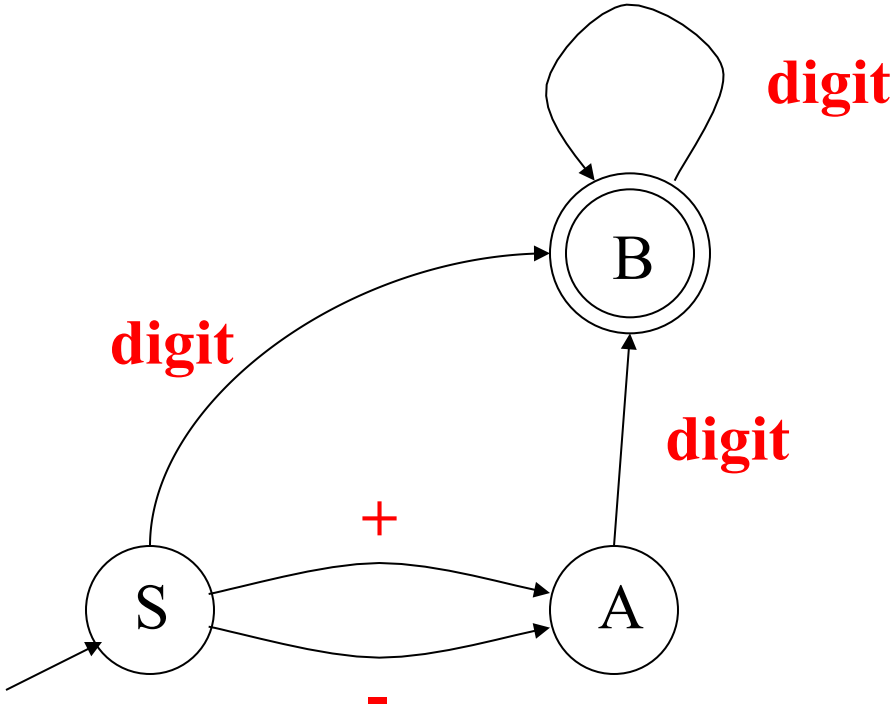
ba+	ba, baa, baaa, baaaa …

- {n} Exactly n occurrences of the preceding character/regex

ba{3}	baaa

Example

- What is the language of this FA?
- Regular expression: (\+|-)?[0-9]+



Regex Rules

- . matches any character
- * is greedy:

  <.*> Home

- Lazy (non-greedy) quantifier:

  <.*?> Home

Regex Rules

- [] Disjunction (Union)

[wW]ood	“how much wood does a Woodchuck chuck?”
[abcd]*	“you are a good programmer”
[A-Za-z0-9]	(any letter or digit)
[A-Za-z]*	(any letter sequence)

- | Disjunction (Union)

(cats?|dogs?)+	“It’s raining cats and a dog.”

- ( ) Grouping

(gupp(y|ies))*	“His guppy is the king of guppies.”

Regex Rules

- ^ $ \b Anchors (start/end of input string; word boundary)

^The	"The cat in the hat."
^The end\.$	"The end."
^The .* end\.$	"The bitter end."
(the)*	"I saw him the other day."
(\bthe\b)*	"I saw him the other day."

- Special rule: when ^ is FIRST WITHIN BRACKETS it means NOT

[^A-Z]* (anything **not** an upper case letter)

Regex Rules

- \ Escape characters

\.	“The + and \ characters are missing.”
\+	“The + and \ characters are missing.”
\\	“The + and \ characters are missing.”

Operator Precedence

Operator	Precedence
()	highest
* + ? {}	
sequences, anchors	
|	lowest

- What is the difference?
  - [a-z][a-z]|[0-9]*
  - [a-z]([a-z]|[0-9])*

Regex for Dollars

- Can you write a regular expression to find all monetary values like $1.50?
- Can you extend it to include commas, like $1,000.50?

Regex for Dollars

- No commas

  \$[0-9]+(\.[0-9][0-9])?

- With commas

  \$[0-9][0-9]?[0-9]?(,[0-9][0-9][0-9])*(\.[0-9][0-9])?

- With or without commas

  \$[0-9][0-9]?[0-9]?((,[0-9][0-9][0-9])*| [0-9]*)
  (\.[0-9][0-9])?

Regex in Python

- `import re`

- `result = re.search(pattern, string)`
- `result = re.findall(pattern, string)`
- `result = re.match(pattern, string)`

- Python documentation on regular expressions
  - http://docs.python.org/release/3.1.3/library/re.html
  - Some useful flags like IGNORECASE, MULTILINE, DOTALL, VERBOSE