Finite Automata & Regular Expressions

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

https://xkcd.com/208/

What is a Computer?

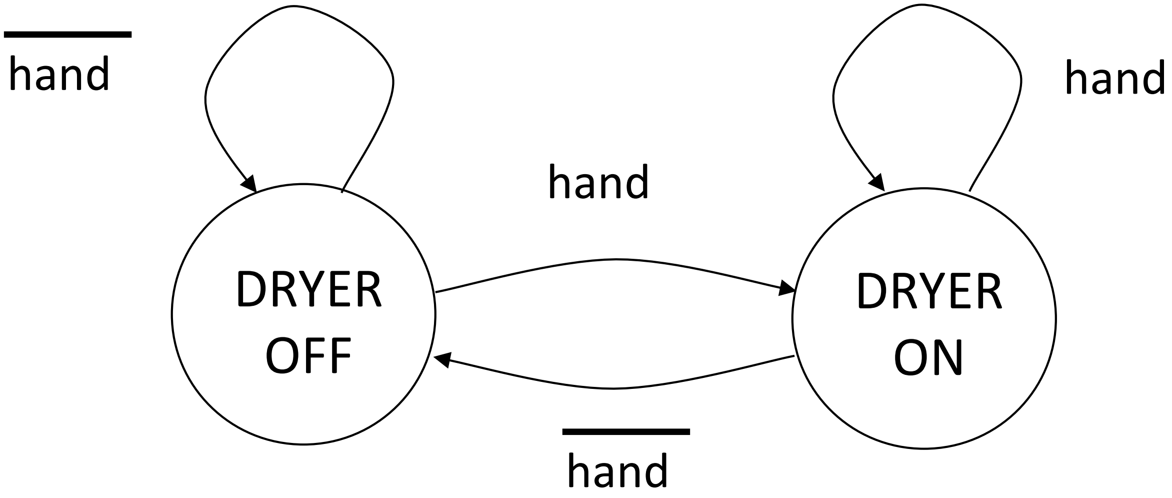
- Theory of computation:
  - What are the fundamental capabilities and limitations of computers?
  - Complexity theory: what makes some problems computationally hard and others easy?
  - Computability theory: what makes some problems solvable and others unsolvable?
  - Automata theory: definitions and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

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

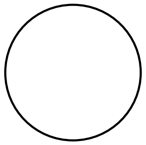
A First Example

- Touch-less hand-dryer

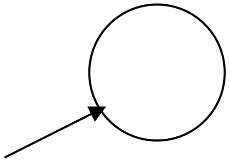


Finite Automata State Graphs

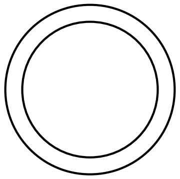
- *A state*



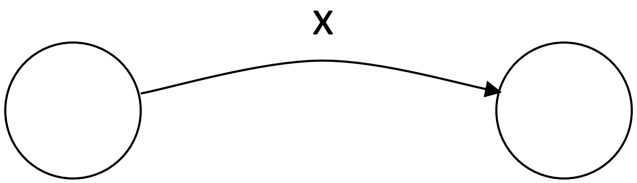
- The start state



- *An accepting state*



- *A transition*

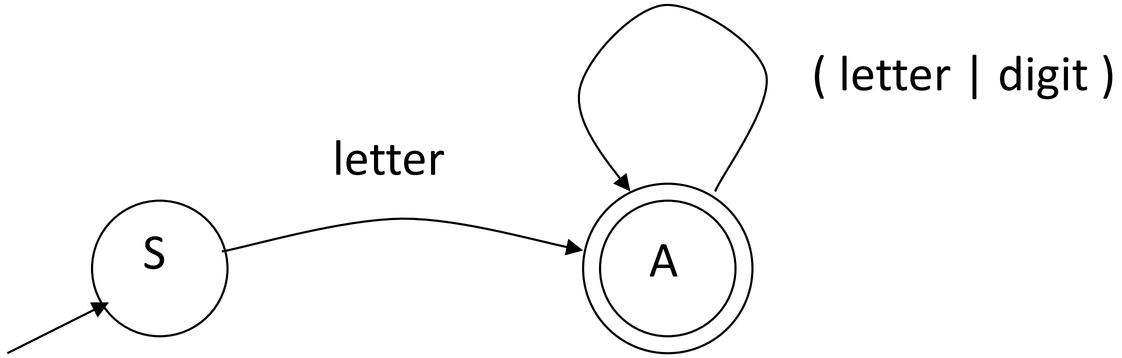


Finite Automata

- Transition: $s_1 \rightarrow^{x} s_2$
  - In state s_1 on input “x” go to state s_2
- At 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:

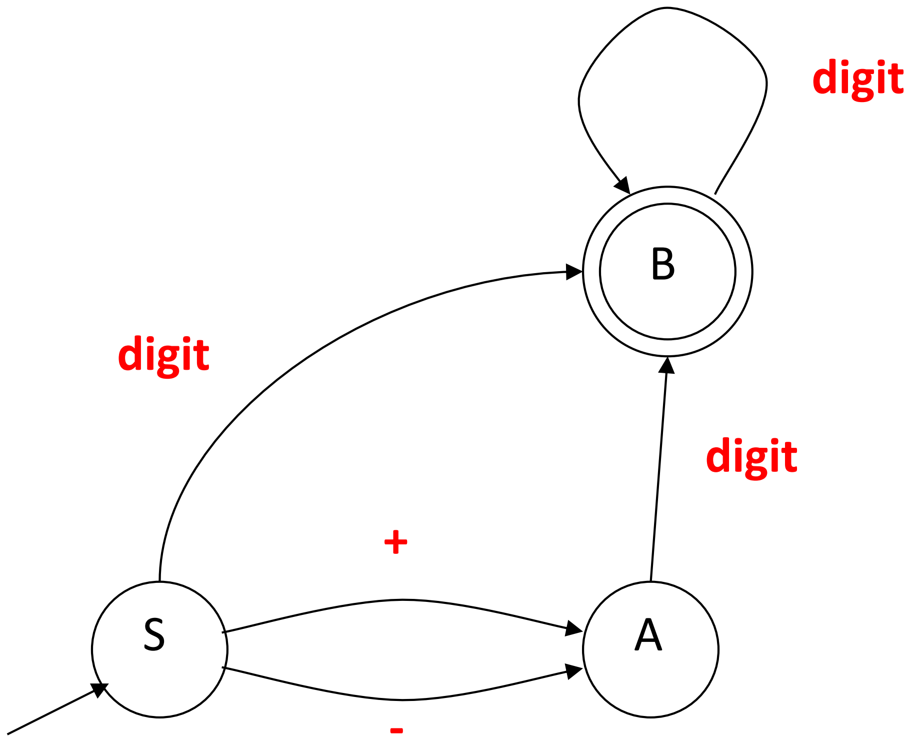


- Which of the following are in the language?
  - x, tmp2, 123, a?, 2apples

- A language is called a <u>regular language</u> if it is recognized by some finite automaton

Example

- What is the language of this FA?



Regular Expressions

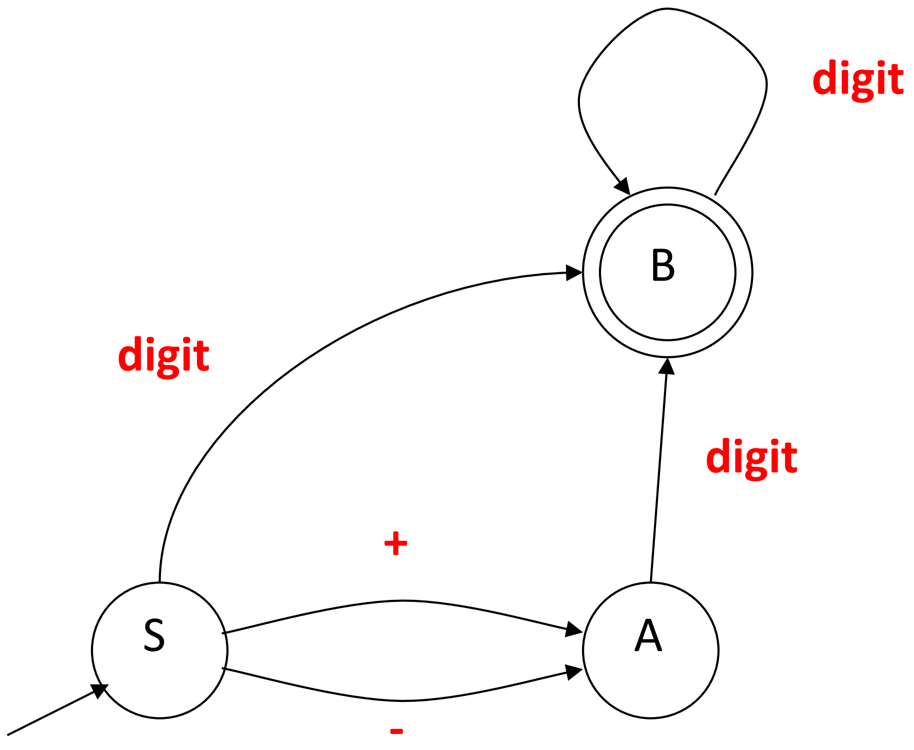
- Regular expressions (regex) are used to describe regular languages

- Arithmetic expression example: (8+2)*3

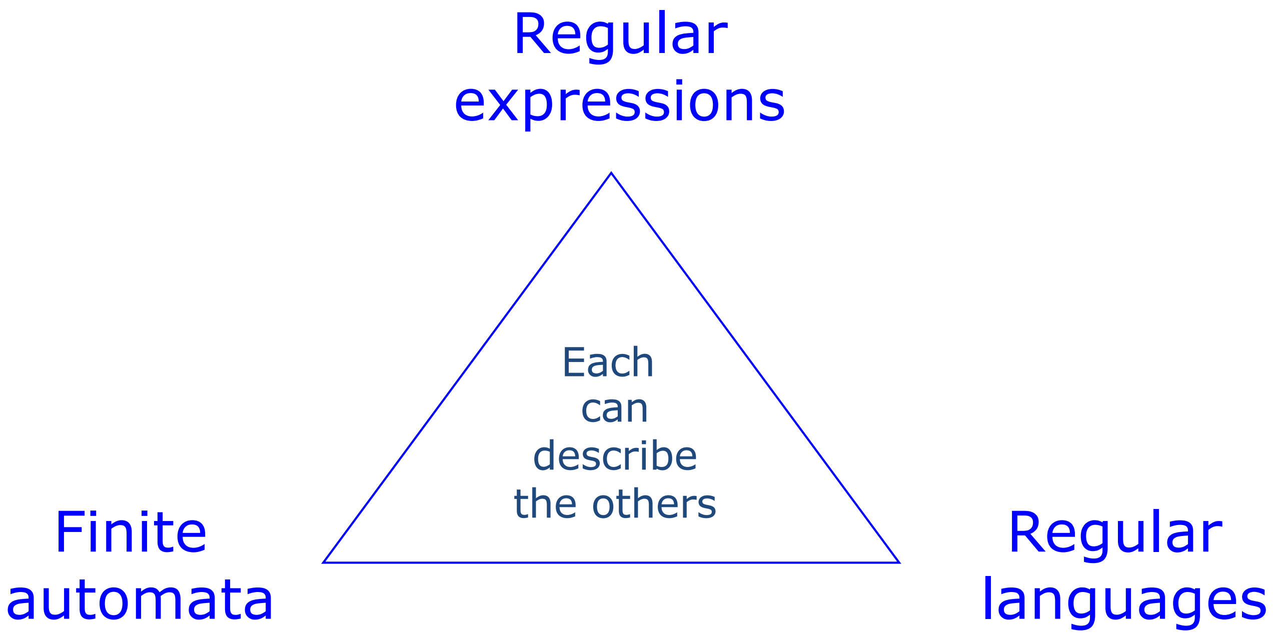
- Regular expression example: (\+|-)?[0-9]+

Example

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



Three Equivalent Representations



Regular expressions

Each can describe the others

Finite automata

Regular languages

Regex Rules

- Goal: match patterns
- String of characters matches the same string

woodchuck	‘how much wood does a woodchuck chuck?’
e	‘you are a programmer’
206	‘INFO206 consists of 206A and 206B.’
!	‘Keep it to yourself!’

- . Wildcard matches any character at that position

p.nt	‘pant, pint, paint, print’

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

Regex Rules

- * is greedy:

  <.*>   <a href=“index.html”>Home</a>

- Lazy (non-greedy) quantifier:

  <.*?>   <a href=“index.html”>Home</a>

Similarly, +? is the non-greedy quantifier for +, and ?? is the non-greedy quantifier for ?

Regex Rules

- [ ] Disjunction (Union)

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

- \| Disjunction (Union)

(cats?|dogs?)+	‘It is 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.’

and so on

Operator Precedence

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

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