Chapter 1: Regular Languages

Section 1.1 Finite automata (DFA's)
Section 1.2 Non-deterministic finite automata (NFA's)
Section 1.3 Regular Expressions - languages -
Section 1.4 Proving certain languages are not regular

Real story 1: DFA's & NFA's are very simple algorithms

Real story 2: grep, egrep (extended grep), looking
in many long text files, searching for the occurrence of
certain substrings, Unix/Linux builds NFA or DFA

Finite automata:
Consider some simple languages, e.g.

\[
\text{DIV\_BY\_2} = \{ s \in \{0,1\}^* | \text{s represents an integer divisible by 2} \}
\]

leading 0's not OK 🙄
empty string, \( \varepsilon \), not OK 😞

0, 2, 4, 117396, 12 \( \in \text{DIV\_BY\_2} \)
13, 121, 012, \( \varepsilon \) \( \notin \text{DIV\_BY\_2} \)

Algorithm: Is 1234567809312 in \text{DIV\_BY\_2}?
Intuitively: build a simple, left-to-right scan of

1 2 3 4 5 6 7 8 0 9 3 1 2

Finite automaton has: finite set of states, \( Q \), involves finite alphabet, \( \Sigma \).

Example:

DIVED BY 2:
- reject \( \varepsilon \)
- reject anything else
- accept 0
- otherwise, \( S_1, \ldots, S_k \)
  - accept if \( S_k = 0, 2, 4, 6, 8 \)
  - reject if \( S_k = 1, 3, 5, 7, 9 \)

Start

Special cases:
- \( \varepsilon \): reject
- \( 0 \): accept
- \( S_2, S_3 \): reject
Formally: states \( Q \), \( \Sigma \), alphabet, \( \delta: Q \times \Sigma \rightarrow Q \)

\( q_0 = \) initial state

\( \delta(q, \sigma) = \) the state that you move to from \( q \) seeing the symbol \( \sigma \)

"accepting state" = "final states" \( F \)

Formally: a deterministic finite automaton is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\)

\[ \text{DIV\_BY\_3} = \{ s \in \{0, 1, \ldots, 9\}^* \mid s \text{ represents an integer divisible by 3} \} \]

\[ \text{PRIMES} = \{ s \in \{0, 1, \ldots, 9\}^* \mid s \text{ represents a prime number} \} \]

We say that a language is regular if there is a DFA recognizes the languages

E.g. \( \text{DIV\_BY\_2} \), \( \text{DIV\_BY\_3} \), are regular

\( \text{SQUARES} \), \( \text{PRIMES} \), is not regular

To show regular, need to build a DFA that recognizes
Dw_Bi-3: DFA: idea: see if what we have so for is:

1) is divisible by 3
2) mod 3 is 1
3) mod 3 is 2

something here

$q_0$

mod 3 is 0

$9 \mod 3, 0$

$5, 1, 4, 7$

mod 3 is 1

$2, 5, 8$

$9 \mod 3, 2$

$0, 3, 6, 9$

$2, 7$

$q_0 \rightarrow 9 \mod 3, 0$

$q_0 \rightarrow 3$

$q_0 \rightarrow \{q_0\}$ where we find a 3