CSC236 winter 2020, week 10: Finite automata

Recommended reading: Chapter 7 Vassos course notes

Colin Morris
colin@cs.toronto.edu
http://www.cs.toronto.edu/~colin/236/W20/

March 16, 2020

Important reading

http://www.cs.toronto.edu/~colin/236/W20/last3weeks/

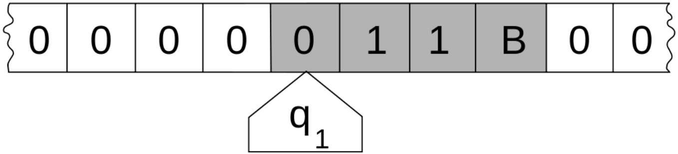
Supplemental reading/viewing

- Vassos course notes, Chapter 7
- David Liu course notes (final chapter, starts pg. 63)
- David Liu also has some excellent YouTube videos on 236 topics

A “purely mechanical process”

It was stated above that ’a function is effectively calculable if its values can be found by some purely mechanical process’. We may take this statement literally, understanding by a purely mechanical process one which could be carried out by a machine. It is possible to give a mathematical description, in a certain normal form, of the structures of these machines.

(Alan Turing, 1939)



Finite state automata

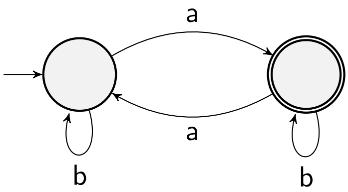
AKA finite state machines, AKA deterministic finite state automata, AKA DFSAs, DFAs...

Example: A machine accepting ODDA $= \{s \in \{a, b\}^* \mid$ s has an odd number of a's$\}$

Finite state automata

AKA finite state machines, AKA deterministic finite state automata, AKA DFSAs, DFAs...

Example: A machine accepting ODDA $= \{s \in \{a, b\}^* \mid \text{s has an odd number of a's}\}$



Anatomy of a state machine

Another example: BIN

Recall from last lecture, BIN, the language of binary numbers (with no redundant leading zeros)

Convention: implicit dead states

When drawing state diagrams, if we don't draw a transition for symbol a from state q, it's assumed to go to a **dead state** (a non-accepting state from which there is no escape).

Your turn: UNIFORM

$\mathrm{UNIFORM} = \{s \in \{a, b\}^* \mid s \text{ consists of non-zero repetitions of a single symbol}\}$.

Formal definition

A DFSA M is a quintuple, $M = (Q, \Sigma, \delta, s, F)$

Q finite set of states

Σ finite alphabet

$\delta : Q \times \Sigma \to Q$ the transition function

$s \in Q$ start state

$F \subseteq Q$ set of accepting states

The transition function, δ

$\delta : Q \times \Sigma \to Q$

$\delta(q_n, a)$ answers the question "where do we go if we're in state q_n and we see the symbol a?"

The *extended* transition function, δ^*

$\delta^* : Q \times \Sigma^* \to Q$.

$\delta^*(q_n, x)$ answers the question "where do we end up if we start from state q_n and process all the symbols in string x?"

A string $x \in \Sigma^*$ is **accepted** by FSA M iff $\delta^*(s, x) \in F$.

$\mathcal{L}(D)$ is the language containing the strings accepted by FSA M.

Proving correctness of automata

cf. 7.3.3 in Vassos notes

WTS: $\mathcal{L}(D) = \text{UNIFORM}$

In other words, WTS the following predicate holds $\forall x \in \{a, b\}^*$:

$P(x)$: D accepts $x \iff x$ is of the form a^k or b^k for some $k > 0$.

Strengthening our predicate

$$P(x) : \delta^*(s, x) = \begin{cases} q_0 & \text{if } x = \varepsilon \\ q_a & \text{if } x = a^k, \text{ for some } k > 0 \\ q_b & \text{if } x = b^k, \text{ for some } k > 0 \\ q_r & \text{otherwise (i.e. x contains a's and b's)} \end{cases}$$

Non-determinism – motivation

Design a DFA that accepts strings in $\{a, b\}^*$ where the second-last symbol is a

The non-deterministic way

From DFA to NFA

Two new features:

1. A state q can have **multiple** transitions when it sees symbol a
   - Instead of mapping to a specific state, δ (and δ^*) map to *sets of states*
2. ε **transitions** — we can have arrows between states labelled with the empty string, ε. These can happen 'spontaneously'

One change to formal definition:

- (previously) $\delta : Q \times \Sigma \to Q$
- (NFA) $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$
  - Where $\mathcal{P}(Q)$ denotes the **powerset** of Q — the set of all subsets. Also written 2^Q.

A use for ε transitions

Let M_1 and M_2 be arbitrary automata. Construct an NFA that accepts $\mathcal{L}(M_1) \cup \mathcal{L}(M_2)$.

Life in a non-deterministic world

An NFA accepts a string x if $F \cap \delta^*(s, x) \neq \emptyset$.
Take your choice of intuition:

- The NFA tries all possible paths at once
- The NFA 'magically' knows the right path to take (if one exists)

(Don't confuse non-determinism with stochasticity. The machine isn't rolling dice.)