Non-deterministic Finite Automata

ian.mcloughlin@gmit.ie

Non-determinism

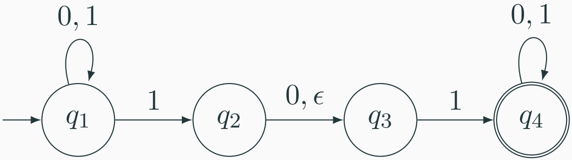
DFAs always have exactly one state to transition to when in any given state and reading any given symbol.

One arrow emerging from each state for each symbol. (Sometimes we use one arrow for two symbols for tidiness.)

Non-deterministic finite automata can have any number of arrows for each state and symbol.

Non-determinism simplifies automata theory, and it can be shown that NFAs and DFAs recognise the same set of languages.

NFA example



Try running the following strings on the automaton.

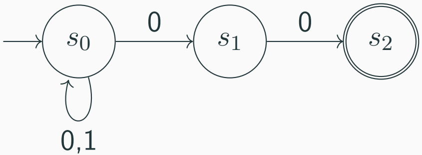
111101, 00001010, 1110, ϵ

Describe in words the strings that the automaton recognises.

Sipser page 48

NFA example

Construct an NFA with alphabet $\{0, 1\}$ to recognise the language w— w ends with 00. Try to do it with only three states.



Sipser Q 1. 7(a)

Non-deterministic Finite Automaton (NFA) definition

A DFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of *states*,
- Σ is a finite set called the *alphabet*,
- δ is the *transition function* $(Q \times \Sigma_\epsilon \to \mathcal{P}(Q))$,
- q_0 is the *start state* $(\in Q)$, and
- F is the set of *accept states* $(\subseteq Q)$.

By Σ_ϵ we mean $\Sigma \cup \{\epsilon\}$. e.g. When $\Sigma = \{0, 1\}$, $\Sigma_\epsilon = \{\epsilon, 0, 1\}$.

Sipser page 35

Powerset example

Take any set, say $A = \{0, 1, 2\}$. Its powerset is the set of all its subsets, and is denoted $\mathcal{P}(A)$.

$$\mathcal{P}(A) = \Big\{ \, \{\} \, , \, \{0\} \, , \, \{1\} \, , \, \{2\} \, , \, \{0, 1\} \, , \, \{0, 2\} \, , \, \{1, 2\} \, , \, \{0, 1, 2\} \, \Big\}$$