COMP30026 Models of Computation

Nondeterministic Finite Automata

Bach Le / Anna Kalenkova

Lecture Week 7 Part 2

Semester 2, 2021

This Lecture is Being Recorded

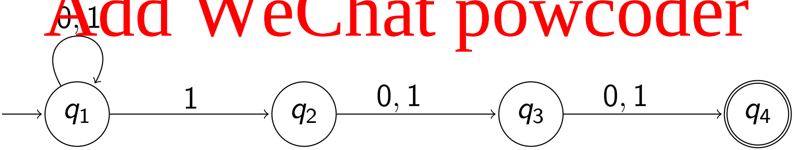
Nondeterminism

The type of machine we have seen so far is called a deterministic finite automaton, or DFA.

We now turn to non-deterministic finite automata, or NFAs.

Here is an NFA that recognises the language

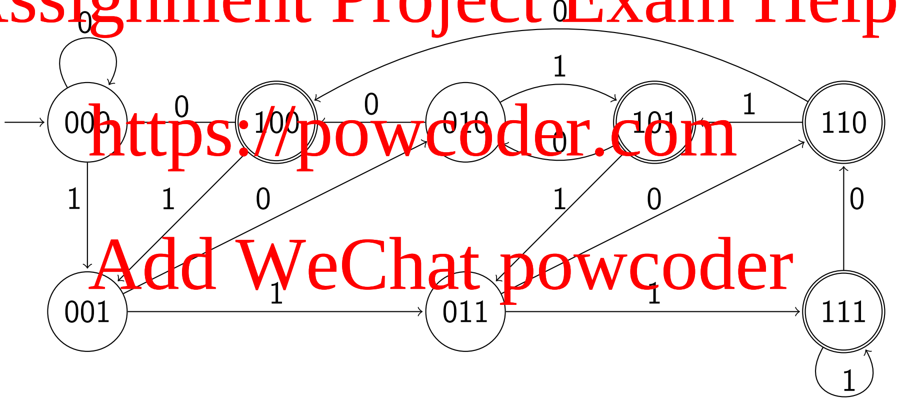
$$\left\{ w \;\middle|\; \begin{array}{l} w \in \{0,1\}^* \text{ has length 3 or more,} \\ \text{and the third last symbol in } w \text{ is } 1 \end{array} \right\}$$



Note: No transitions from q_4, and two possible transitions when we meet a 1 in state q_1.

Nondeterminism

The NFA is more intelligible than a DFA for the same language:



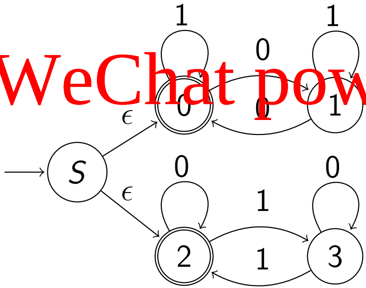
This is the simplest DFA that will do the job!

Epsilon Transitions

NFAs may also be allowed to move from one state to another without consuming input.

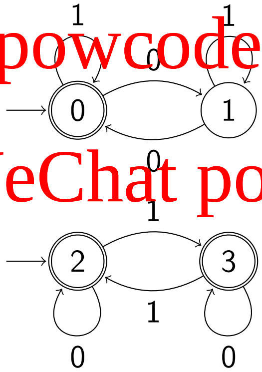
Such a transition is an ϵ transition.

Among other things, this gives us an easy way to construct a machine to recognise the union of two languages:



Epsilon transitions are often useful, but in the previous example we actually did not need them, because an NFA is also allowed to have multiple start states.

This NFA is equivalent to the previous one, but it has only four states:



Formal Definition of NFA

For any alphabet Σ let Σ_ϵ denote $\Sigma \cup \{\epsilon\}$.

An NFA is a 5-tuple $(Q, \Sigma, \delta, I, F)$, where

- Q is a finite set of states,
- Σ is a finite alphabet,
- $\delta : Q \times \Sigma_\epsilon \to \mathcal{P}(Q)$ is the transition function,
- $I \subseteq Q$ are the start states, and
- $F \subseteq Q$ are the accept states.

Note that, unlike a DFA, an NFA can have several “start” states—it can start executing from any one of those.

NFA Acceptance and Recognition, Formally

The definition of what it means for an NFA N to accept a string says that it has to be possible to make the necessary transitions.

Let $N = (Q, \Sigma, \delta, I, F)$ be an NFA and let $w = v_1 v_2 \cdots v_n$ where each v_i is a member of Σ_ϵ.

N accepts w iff there is a sequence of states $r_0, r_1, \ldots, r_n$, with each $r_i \in Q$, such that

1. $r_0 \in I$
2. $r_{i+1} \in \delta(r_i, v_{i+1})$ for $i = 0, \ldots, n-1$
3. $r_n \in F$

N recognises language A iff $A = \{w \mid N \text{ accepts } w\}$.

DFAs vs NFAs

The class of languages recognised by NFAs is exactly the class of regular languages.

Theorem: Every NFA has an equivalent DFA.

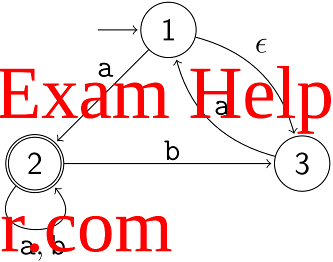
The proof rests on the so-called subset construction.

Given NFA N, we construct DFA M, each of whose states corresponds to a set of N-states.

If N has k states then M may have up to 2^k states (but it will often have far fewer than that).

DFAs vs NFAs

Consider the NFA on the right. We can systematically construct an equivalent DFA from the NFA.



The DFA's start state is $\{1,3\}$.
From $\{1,3\}$, a takes us to $\{1,2,3\}$.
From $\{1,2,3\}$, a takes us back to $\{1,2,3\}$, b takes us to $\{2,3\}$.

Any state S which contains an accept state from the NFA will be an accept state for the DFA. Here we mark accept states with a star.

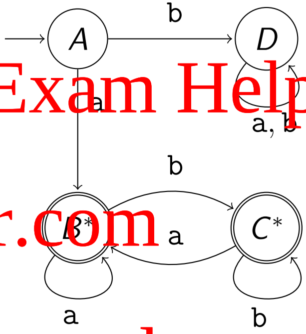
	a	b
A={1,3}	B*	–
B*={1,2,3}	B*	C*
C*={2,3}	B*	C*

DFAs vs NFAs

Here is the equivalent DFA that we derive.

Any state S which contains an accept state from the NFA (in this case the NFA has just one, namely state 2) becomes an accept state for the DFA.

We add (dead) state D that corresponds to the empty set.



	a	b
A={1,3}	B*	D
B*={1,2,3}	B*	C*
C*={2,3}	B*	C*
D=∅	D	D

More Formally . . .

Let $N = (Q, \Sigma, \delta, I, F)$. Let $\xrightarrow{*}_\epsilon$ be the reflexive transitive closure of $\rightarrow_\epsilon$, which in turn is defined by $s \rightarrow_\epsilon s'$ iff $s' \in \delta(s, \epsilon)$.

Let $E(S)$ be the "ϵ closure" of $S \subseteq Q$, that is, S together with all states reachable from states in S, using only ϵ steps:

$$E(S) = \bigcup_{s \in S} \{s' \in Q \mid s \xrightarrow{*}_\epsilon s'\}$$

We construct $M = (\mathcal{P}(Q), \Sigma, \delta', q_0, F')$ as follows:

- $q_0 = E(I)$
- $\delta'(S, v) = \bigcup_{s \in S} E(\delta(s, v))$
- $F' = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$

Note: This construction may include unreachable states.