Nondeterministic Finite Automata - Lecture 4

James Marshall

Definition - Computation by an NFA

An NFA $N = (Q, \Sigma, \delta, q_0, F)$ accepts string $w \in \Sigma^*$ if and only if $w = y_1 y_2 ... y_m$ where $y_i \in \Sigma_\varepsilon$, and a sequence of states $r_0, r_1, r_2 ... r_m$ exists in Q satisfying:

1. $r_0 = q_0$
2. $r_{i+1} \in \delta (r_i, y_{i+1})$ for all $i = 0, 1, ..., m - 1$
3. $r_m \in F$

(the definition for computation by a DFA is very similar.
Q: what would be the only two differences?)

Nondeterministic finite automata have the same computational power as deterministic finite automata

Theorem

Every nondeterministic finite automaton has an equivalent deterministic finite automaton

Proof (by construction)

For an NFA $N = (Q, \Sigma, \delta, q_0, F)$ recognising language A, construct a DFA $M = (Q', \Sigma, \delta', q'_0, F')$ that also recognises A as follows:

1. $Q' = \mathcal{P}(Q)$
2. $\delta' (R, a) = \bigcup_{r \in R} \delta (r, a)$
3. $q'_0 = \{q_0\}$
4. $F' = \{R \in Q' \mid R \text{ contains an accept state of } N\}$

Corollary

A language is regular if and only if it is recognised by an NFA.

An example of an NFA (from the last lecture):

...and its equivalent DFA (**N.B.** Construction of DFA *M* only works if NFA *N* has no ε transitions. The definition of *M*'s transition function (2) and start state (3) need to be extended, see Sipser p.56):

Theorem

The class of regular languages is closed under the union operator

Proof Sketch (by construction using NFAs)

Theorem

The class of regular languages is closed under the star operator

Proof Sketch (by construction using NFAs)