Finite Automata

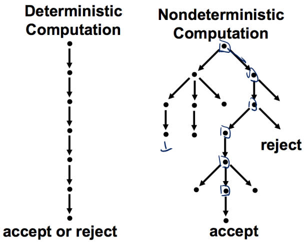
- Deterministic Finite Automata
- Languages decided by a DFA – Regular Languages
- Closure Properties of regular languages
- Non-Deterministic Finite Automata, DFA= NFA
- Regular Expression: Computation as Description
- DFA=NFA=RegExp, Generalized NFA
- Non-Regular Languages, The Pumping Lemma
- Minimizing DFA

IMDAD ULLAH KHAN

Equivalence of DFA and NFA

Equivalence of DFA and NFA

Ways to think about non-determinism



Parallel computation (with certain restriction) and accepting when one of the node succeeds

Or tree of all possible walks from a start state branching according to symbols on edges and accepting if any leaf node is a final state

Or computing with guessing capability to choose the next state (at certain states) and verifying the right choice

Does verified guessing capability of NFA increases its power over DFA's?

Are NFA and DFA equal in computational power?

Equivalence of DFA and NFA

NFA and DFA are equal in computational power!

Every DFA is an NFA, so NFA's are at least as powerful as DFA

Regular Languages $=$ DFA-Recognizable Langauges $\subseteq$ NFA-Recognizable Languages

For every NFA N, there is a DFA M such that $L(M) = L(N)$

Every NFA can be perfectly simulated by some DFA

Regular Languages $=$ DFA-Recognizable Langauges $=$ NFA-Recognizable Languages

Equivalence of DFA and NFA

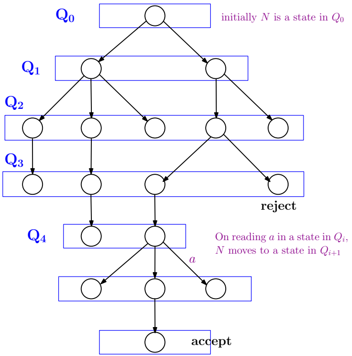
Every NFA can be perfectly simulated by some DFA

For every NFA N, there is a DFA M such that $L(M) = L(N)$

$N = (Q, \Sigma, Q_0, \delta, F) \longrightarrow$ DFA $M = (Q', \Sigma, q_0, \delta', F')$ recognizing $L(N)$

Intuitively,

- M runs all possible threads of N in parallel
- Maintains the set of states where all threads can be after each step



Number and lengths of paths depend on given input, cannot maintain that

Equivalence of DFA and NFA

Every NFA can be perfectly simulated by some DFA

For every NFA N, there is a DFA M such that $L(M) = L(N)$

$N = (Q, \Sigma, Q_0, \delta, F) \longrightarrow$ DFA $M = (Q', \Sigma, q_0, \delta', F')$ recognizing $L(N)$

- $Q' = \mathcal{P}(Q)$ $\triangleright$ exponential in $|Q|$ but finite
- $\Sigma = \Sigma$
- $q_0 = Q_0$
- $F' = \{R \in Q' : R$ contain some state in F i.e. $R \cap F \neq \emptyset\}$
- $\delta : Q' \times \Sigma \mapsto Q'$
  For $R \subseteq Q$ and $\sigma \in \Sigma$, $\delta'(R, \sigma) = \bigcup_{q \in R} \delta(q, \sigma)$

Does it take into account ϵ-transitions?

Equivalence of DFA and NFA

Every NFA can be perfectly simulated by some DFA

For every NFA N, there is a DFA M such that $L(M) = L(N)$

$N = (Q, \Sigma, Q_0, \delta, F) \longrightarrow$ DFA $M = (Q', \Sigma, q_0, \delta', F')$ recognizing $L(N)$

- $Q' = \mathcal{P}(Q)$ ▷ exponential in $|Q|$ but finite
- $\Sigma = \Sigma$
- $q_0 = E\Big(Q_0\Big)$ ▷ **Subset Construction**
- $F' = \{R \in Q' : R$ contain some state in F i.e. $R \cap F \neq \emptyset\}$
- $\delta : Q' \times \Sigma \mapsto Q'$

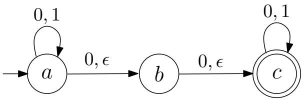
  For $R \subseteq Q$ and $\sigma \in \Sigma$, $\delta'(R, \sigma) = \bigcup_{q \in R} E\Big(\delta(q, \sigma)\Big)$,

$E\big(R\big) = \{r \in Q : r$ is reachable from some $q \in R$ in ≥ 0 ϵ-transitions$\}$

▷ ϵ-closure of the set R

ϵ-closure of the set R

$E(R) = \{r \in Q : r$ is reachable from some $q \in R$ in ≥ 0 ϵ-transitions$\}$



$$
\begin{aligned}
E(\{a\}) &= \{a, b, c\} \\
E(\{b\}) &= \{b, c\} \\
E(\{c\}) &= \{c\} \\
E(\{b, c\}) &= E(\{b\}) \cup E(\{c\})
\end{aligned}
$$

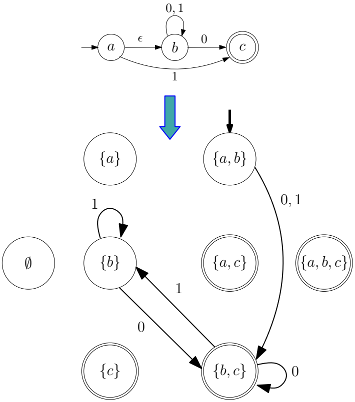
Simulating NFA by DFA

For every NFA N, there is a DFA M such that $L(M) = L(N)$

$N = (Q, \Sigma, Q_0, \delta, F) \longrightarrow$ DFA $M = (Q', \Sigma, q_0, \delta', F')$ recognizing $L(N)$

- $Q' = \mathcal{P}(Q)$
- $\Sigma = \Sigma$
- $q_0 = E\Big(Q_0\Big)$
- $F' = \{R \in Q' : R \cap F \neq \emptyset\}$
- $\delta : Q' \times \Sigma \mapsto Q'$ For $R \subseteq Q$ and $\sigma \in \Sigma$,
  $\delta'(R, \sigma) = \bigcup_{q \in R} E\big(\delta(q, \sigma)\big)$

$E(R) = \{q \in Q :$
q reachable from R in $\geq 0\, \epsilon$-transition$\}$



For every NFA N, there is a DFA M such that $L(M) = L(N)$

$N = (Q, \Sigma, Q_0, \delta, F) \longrightarrow$ DFA $M = (Q', \Sigma, q_0, \delta', F')$ recognizing $L(N)$

The process is termed as Subset Construction