CS310M: Automata Theory (Minor)

Topic 3: Nondeterministic Finite Finite State Automata

Paritosh Pandya

Indian Institute of Technology, Bombay
Course URL: https://cse.iitb.ac.in/∼pandya58/CS310M/automata.html

Autumn, 2021

Today's Topics:

- Operations on Languages
- Closure Under Boolean Operations
- Closure Under Catenation and Kleene Star
- Closure Under Homomorphism
- Decision Problems:

Source: HMP Sect 3.2.3. Kozen L6 (end). Kozen L10.

Language Operations

Let $A, A_1, A_2 \subseteq \Sigma^*$. Define

- Union $A_1 \cup A_2$
- Intersection $A_1 \cap A_2$
- Complementation $\sim A = \Sigma^* - A$.
- Catenation $A_1 \cdot A2$.
- Kleene Closure A^*
- Reverse $rev(A)$

Closure of Regular languages under operations

Theorem If A_1, A_2 are regular then $A_1 \cap A_2$ is regular.

Language Operations

Let $A, A_1, A_2 \subseteq \Sigma^*$. Define

- Union $A_1 \cup A_2$
- Intersection $A_1 \cap A_2$
- Complementation $\sim A = \Sigma^* - A$.
- Catenation $A_1 \cdot A2$.
- Kleene Closure A^*
- Reverse $rev(A)$

Closure of Regular languages under operations

Theorem If A_1, A_2 are regular then $A_1 \cap A_2$ is regular.
Proof Method: Given DFA M_1, M_2 s.t. $A_1 = L(M_1)$ and $A_2 = L(M_2)$ we construct DFA M_3 s.t. $L(M_3) = A_1 \cap A_2$. The size $|M_3| = |M_1| \times |M_2|$.

Language Operations

Let $A, A_1, A_2 \subseteq \Sigma^*$. Define

- Union $A_1 \cup A_2$
- Intersection $A_1 \cap A_2$
- Complementation $\sim A = \Sigma^* - A$.
- Catenation $A_1 \cdot A2$.
- Kleene Closure A^*
- Reverse $rev(A)$

Closure of Regular languages under operations

Theorem If A_1, A_2 are regular then $A_1 \cap A_2$ is regular.
Proof Method: Given DFA M_1, M_2 s.t. $A_1 = L(M_1)$ and $A_2 = L(M_2)$ we construct DFA M_3 s.t. $L(M_3) = A_1 \cap A_2$. The size $|M_3| = |M_1| \times |M_2|$.

We showed that regular languages are closed under $\cap$, $\sim$, $\cup$.

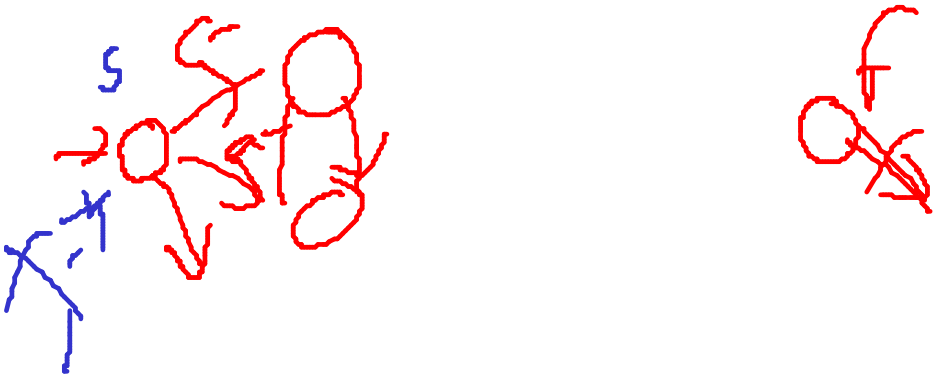
Complexity of closure operations

DFA	DFA	NFA
. Complement	m	2^m ←
. Intersection	$m \times n$	$m \times n$
. Union	$m \times n$	$m + n$

Normalized ϵ-NFA

An ϵ-NFA is normalized if

- It has a single start state and a single final state.
- There are no incoming transitions into start state.
- There are no outgoing transitions from the final state.



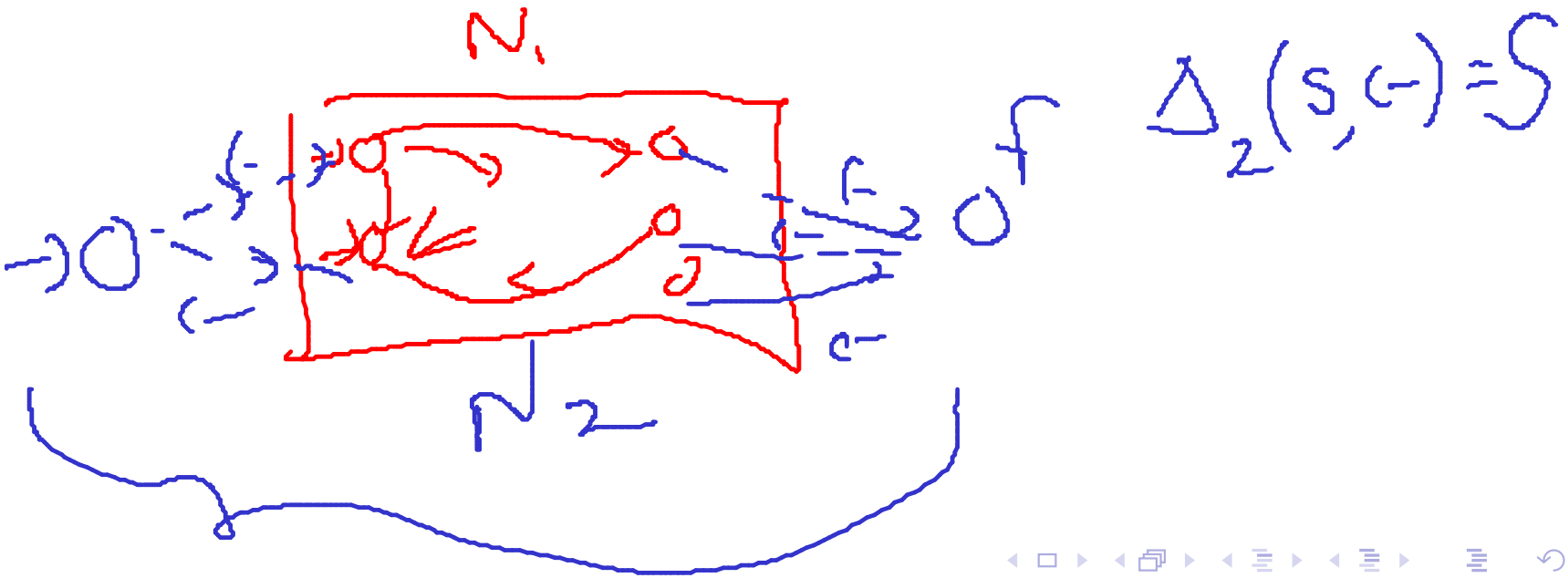
Normalized ϵ-NFA

An ϵ-NFA is normalized if

- It has a single start state and a single final state.
- There are no incoming transitions into start state.
- There are no outgoing transitions from the final state.

Theorem

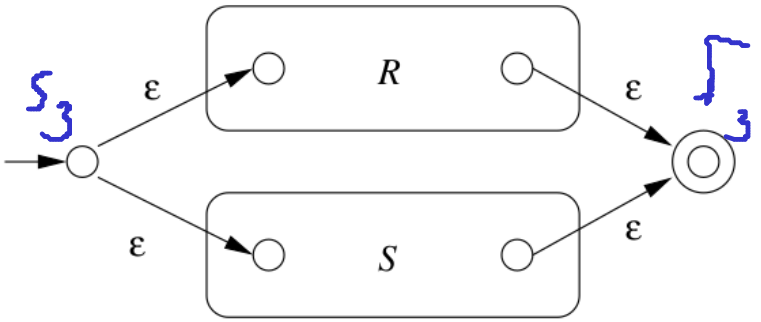
Given any ϵ-NFA $N_1 = (Q, \Sigma, \Delta, S, F)$ we can construct a normalized ϵ-NFA $N_2 = (Q \cup \{s, f\}, \Sigma, \Delta_2, s, f)$ s.t. $L(N_1) = L(N_2)$.



Closure under Union

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ and $N_2 = (Q_2, \Sigma, \Delta_1, s_2, f_2)$ we can construct normalized ϵ-NFA $N_3 = (Q_3, \Sigma, \Delta_3, s_3, f_3)$ s.t. $L(N_3) = L(N_1) \cup L(N_2)$.



Closure under Union

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ and $N_2 = (Q_2, \Sigma, \Delta_1, s_2, f_2)$ we can construct normalized ϵ-NFA $N_3 = (Q_3, \Sigma, \Delta_3, s_3, f_3)$ s.t. $L(N_3) = L(N_1) \cup L(N_2)$.

- $Q_3 = Q_1 \cup Q_2 \cup \{s_3, f_3\}$ fresh states s_3, f_3.

Closure under Union

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ and $N_2 = (Q_2, \Sigma, \Delta_1, s_2, f_2)$ we can construct normalized ϵ-NFA $N_3 = (Q_3, \Sigma, \Delta_3, s_3, f_3)$ s.t. $L(N_3) = L(N_1) \cup L(N_2)$.

- $Q_3 = Q_1 \cup Q_2 \cup \{s_3, f_3\}$ fresh states s_3, f_3.
- $\Delta_3(s_3, \epsilon) = \{s_1, s_2\}$, and $\forall a \in \Sigma.\ \Delta_3(s_3, a) = \emptyset$.

Closure under Union

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ and $N_2 = (Q_2, \Sigma, \Delta_1, s_2, f_2)$ we can construct normalized ϵ-NFA $N_3 = (Q_3, \Sigma, \Delta_3, s_3, f_3)$ s.t. $L(N_3) = L(N_1) \cup L(N_2)$.

- $Q_3 = Q_1 \cup Q_2 \cup \{s_3, f_3\}$ fresh states s_3, f_3.
- $\Delta_3(s_3, \epsilon) = \{s_1, s_2\}$, and $\forall a \in \Sigma.\ \ \Delta_3(s_3, a) = \emptyset$.
- $\forall s \in Q_1.\ \ \Delta_3(s, a) = \Delta_1(s, a)$ and
  $\forall s \in Q_2.\ \ \Delta_3(s, a) = \Delta_2(s, a)$.

Closure under Union

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ and $N_2 = (Q_2, \Sigma, \Delta_1, s_2, f_2)$ we can construct normalized ϵ-NFA $N_3 = (Q_3, \Sigma, \Delta_3, s_3, f_3)$ s.t. $L(N_3) = L(N_1) \cup L(N_2)$.

- $Q_3 = Q_1 \cup Q_2 \cup \{s_3, f_3\}$ fresh states s_3, f_3.
- $\Delta_3(s_3, \epsilon) = \{s_1, s_2\}$, and $\forall a \in \Sigma.\ \Delta_3(s_3, a) = \emptyset$.
- $\forall s \in Q_1.\ \Delta_3(s, a) = \Delta_1(s, a)$ and
  $\forall s \in Q_2.\ \Delta_3(s, a) = \Delta_2(s, a)$.
- $\forall s \in Q_1 - \{f_1\}.\ \Delta_3(s, \epsilon) = \Delta_1(s, \epsilon)$, and
  $\forall s \in Q_2 - \{f_2\}.\ \Delta_3(s, \epsilon) = \Delta_2(s, \epsilon)$.

Closure under Union

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ and $N_2 = (Q_2, \Sigma, \Delta_1, s_2, f_2)$ we can construct normalized ϵ-NFA $N_3 = (Q_3, \Sigma, \Delta_3, s_3, f_3)$ s.t. $L(N_3) = L(N_1) \cup L(N_2)$.

- $Q_3 = Q_1 \cup Q_2 \cup \{s_3, f_3\}$ fresh states s_3, f_3.
- $\Delta_3(s_3, \epsilon) = \{s_1, s_2\}$, and $\forall a \in \Sigma.\ \Delta_3(s_3, a) = \emptyset$.
- $\forall s \in Q_1.\ \Delta_3(s, a) = \Delta_1(s, a)$ and
  $\forall s \in Q_2.\ \Delta_3(s, a) = \Delta_2(s, a)$.
- $\forall s \in Q_1 - \{f_1\}.\ \Delta_3(s, \epsilon) = \Delta_1(s, \epsilon)$, and
  $\forall s \in Q_2 - \{f_2\}.\ \Delta_3(s, \epsilon) = \Delta_2(s, \epsilon)$.
- $\Delta_3(f_1, \epsilon) = \Delta_1(f_1, \epsilon) \cup \{f_3\}$ and
  $\Delta_3(f_2, \epsilon) = \Delta_2(f_2, \epsilon) \cup \{f_3\}$

Catenation

Catenation of Languages: Given languages A, B over Σ, catenation $A.B$ is defined as

$$A \cdot B = \{x \cdot y \mid x \in A \text{ and } y \in B\}$$

Example Let $A = \{a, ab\}$ and $B = \{b, ba\}$ Then,
$A \cdot B = \{ab, aba, abb, abba\}$
$B \cdot A = \{ba, bab, baa, baab\}$

Catenation

Catenation of Languages: Given languages A, B over Σ, catenation $A.B$ is defined as

$$A \cdot B = \{x \cdot y \mid x \in A \text{ and } y \in B\}$$

Example Let $A = \{a, ab\}$ and $B = b, ba$. Then,
$A \cdot B = \{ab,\ aba,\ abb,\ abba\}$
$B \cdot A = \{ba,\ bab,\ baa,\ baab\}$

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ and $N_2 = (Q_2, \Sigma, \Delta_1, s_2, f_2)$ we can construct normalized ϵ-NFA $N_3 = (Q_1 \cup Q_2, \Sigma, \Delta_3, s_1, f_2)$ s.t. $L(N_3) = L(N_1) \cdot L(N_2)$.

Catenation

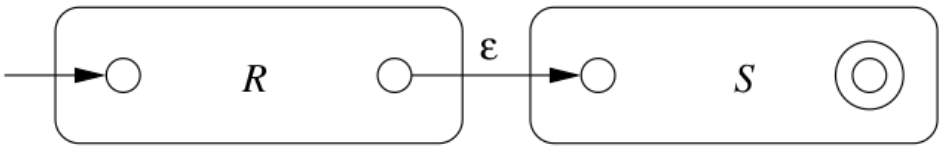
Catenation of Languages: Given languages A, B over Σ, catenation $A.B$ is defined as

$$A \cdot B = \{x \cdot y \mid x \in A \text{ and } y \in B\}$$

Example Let $A = \{a, ab\}$ and $B = b, ba$. Then,
$A \cdot B = \{ab,\ aba,\ abb,\ abba\}$
$B \cdot A = \{ba,\ bab,\ baa,\ baab\}$

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ and $N_2 = (Q_2, \Sigma, \Delta_1, s_2, f_2)$ we can construct normalized ϵ-NFA $N_3 = (Q_1 \cup Q_2, \Sigma, \Delta_3, s_1, f_2)$ s.t. $L(N_3) = L(N_1) \cdot L(N_2)$.



Kleene Closure

$NP = (\{A, The, An\} \cup \{\epsilon\}) \cdot (Adjective)^* \cdot Noun$

The Cat

The black Cat

The Nasty black Cat

Kleene Closure

Power

Let $A \subseteq \Sigma^*$. Define $A^0 = \{\epsilon\}$ and $A^{n+1} = A \cdot A^n$.

$$
\begin{aligned}
\{ab, aab\}^0 &= \{\epsilon\}, \\
\{ab, aab\}^1 &= \{ab, aab\}, \\
\{ab, aab\}^2 &= \{abab, abaab, aabab, aabaab\}, \\
\{ab, aab\}^3 &= \{ababab, ababaab, abaabab, aababab, \\
&\qquad abaabaab, aababaab, aabaabab, aabaabaab\}.
\end{aligned}
$$

Kleene Closure

Power

Let $A \subseteq \Sigma^*$. Define $A^0 = \{\epsilon\}$ and $A^{n+1} = A \cdot A^n$.

$$
\begin{aligned}
\{ab, aab\}^0 &= \{\epsilon\}, \\
\{ab, aab\}^1 &= \{ab, aab\}, \\
\{ab, aab\}^2 &= \{abab, abaab, aabab, aabaab\}, \\
\{ab, aab\}^3 &= \{ababab, ababaab, abaabab, aababab, \\
&\qquad abaabaab, aababaab, aabaabab, aabaabaab\}.
\end{aligned}
$$

Kleene Closure: Given $A \subseteq \Sigma^*$ define language A^*

$$
\begin{aligned}
A^* &= \bigcup_{n \geq 0} A^n \\
&= A^0 \cup A^1 \cup A^2 \cup \ldots
\end{aligned}
$$

Example: $A = \{ab\}$ then $A^* = \{(ab)^n \mid n \geq 0\}$
$= \{\epsilon, ab, abab, ababab, \ldots\}$.

$\Sigma = \{0,1\}$

$L = \{w \in \Sigma^* \mid \#_1(w) \text{ is } 1\}$

$\Sigma^* \cdot 1 \cdot \Sigma^*$

Kleen Closure (2)

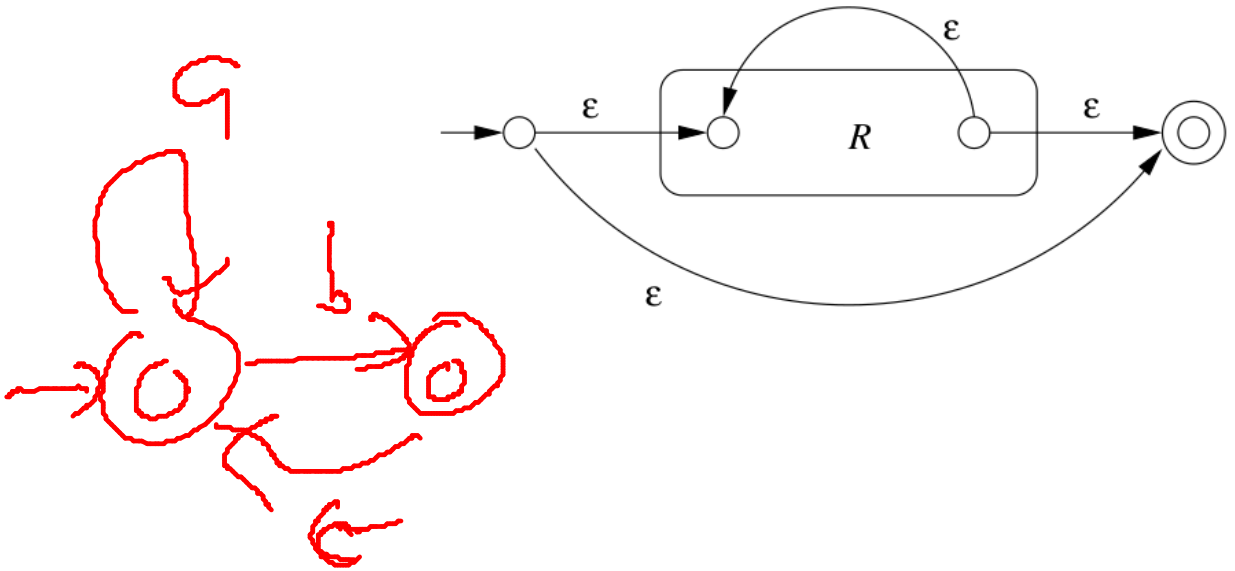
Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ we can construct normalized ϵ-NFA $N_2 = (Q_1 \cup \{s_2, t_2\}, \Sigma, \Delta_2, s_2, f_2)$ s.t. $L(N_2) = (L(N_1))^*$.

Kleen Closure (2)

Theorem

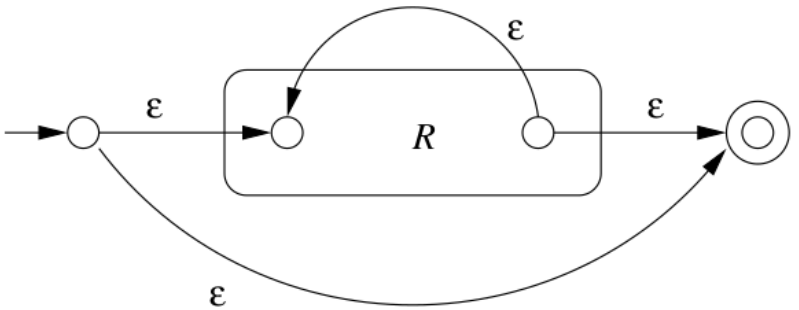
Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ we can construct normalized ϵ-NFA $N_2 = (Q_1 \cup \{s_2, t_2\}, \Sigma, \Delta_2, s_2, f_2)$ s.t. $L(N_2) = (L(N_1))^*$.



Kleen Closure (2)

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ we can construct normalized ϵ-NFA $N_2 = (Q_1 \cup \{s_2, t_2\}, \Sigma, \Delta_2, s_2, f_2)$ s.t. $L(N_2) = (L(N_1))^*$.

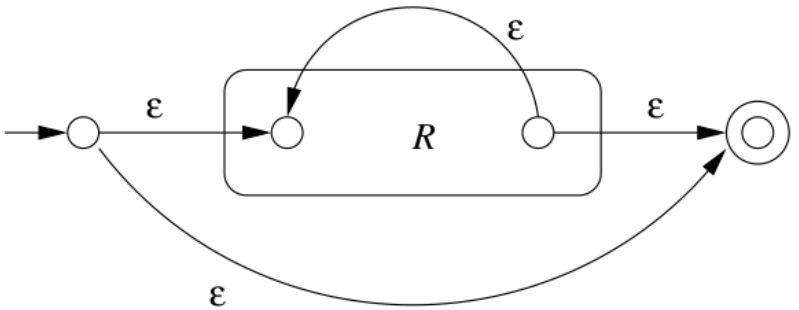


- $Q_2 = Q_1 \cup \{s_2, f_2\}$ fresh states s_2, f_2.

Kleen Closure (2)

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ we can construct normalized ϵ-NFA $N_2 = (Q_1 \cup \{s_2, t_2\}, \Sigma, \Delta_2, s_2, f_2)$ s.t. $L(N_2) = (L(N_1))^*$.

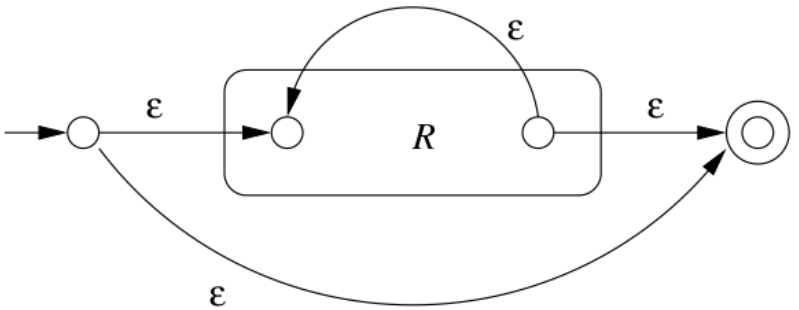


- $Q_2 = Q_1 \cup \{s_2, f_2\}$ fresh states s_2, f_2.
- $\Delta_2(s_2, \epsilon) = \{s_1, t_2\}$, and $\forall a \in \Sigma.\ \ \Delta_2(s_2, a) = \emptyset$.

Kleen Closure (2)

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ we can construct normalized ϵ-NFA $N_2 = (Q_1 \cup \{s_2, t_2\}, \Sigma, \Delta_2, s_2, f_2)$ s.t. $L(N_2) = (L(N_1))^*$.

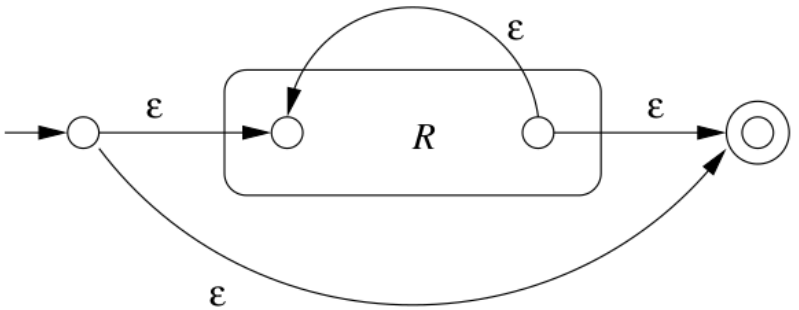


- $Q_2 = Q_1 \cup \{s_2, f_2\}$ fresh states s_2, f_2.
- $\Delta_2(s_2, \epsilon) = \{s_1, t_2\}$, and $\forall a \in \Sigma.\ \ \Delta_2(s_2, a) = \emptyset$.
- $\forall s \in Q_1.\ \ \Delta_2(s, a) = \Delta_1(s, a)$ and $\forall s \in Q_1 - \{f_1\}.\ \ \Delta_2(s, \epsilon) = \Delta_1(s, \epsilon)$.

Kleen Closure (2)

Theorem

Given normalized ϵ-NFA $N_1 = (Q_1, \Sigma, \Delta_1, s_1, f_1)$ we can construct normalized ϵ-NFA $N_2 = (Q_1 \cup \{s_2, t_2\}, \Sigma, \Delta_2, s_2, f_2)$ s.t. $L(N_2) = (L(N_1))^*$.



- $Q_2 = Q_1 \cup \{s_2, f_2\}$ fresh states s_2, f_2.
- $\Delta_2(s_2, \epsilon) = \{s_1, t_2\}$, and $\forall a \in \Sigma.\ \ \Delta_2(s_2, a) = \emptyset$.
- $\forall s \in Q_1.\ \ \Delta_2(s, a) = \Delta_1(s, a)$ and $\forall s \in Q_1 - \{f_1\}.\ \ \Delta_2(s, \epsilon) = \Delta_1(s, \epsilon)$.
- $\Delta_2(f_1, \epsilon) = \Delta_1(f_1, \epsilon) \cup \{s_1, f_2\}$.

Reverse of a Language

- For a word x let $rev(x)$ denote its reverse. E.g. $rev(abab) = baba$.

Reverse of a Language

- For a word x let $rev(x)$ denote its reverse. E.g. $rev(abab) = baba$.
- For $A \subseteq \Sigma^*$, let $rev(A) = \{rev(x) \mid x \in A\}$.

Reverse of a Language

- For a word x let $rev(x)$ denote its reverse. E.g. $rev(abab) = baba$.
- For $A \subseteq \Sigma^*$, let $rev(A) = \{rev(x) \mid x \in A\}$.
- Theorem: Regular languages are closed under reverse.

Reverse of a Language

- For a word x let $rev(x)$ denote its reverse. E.g. $rev(abab) = baba$.
- For $A \subseteq \Sigma^*$, let $rev(A) = \{rev(x) \mid x \in A\}$.
- Theorem: Regular languages are closed under reverse.

Construction of reverse automaton: Given ϵ-NFA $N_1 = (Q, \Sigma, \epsilon, \Delta_1, S, F)$ we obtain $N_2 = (Q, \Sigma, \epsilon, \delta_2, F, S)$ where we reverse every transition of Δ_1 to get Δ_2.

$$(p \xrightarrow{\alpha} q) \in \delta_2 \quad \text{iff} \quad (q \xrightarrow{\alpha} p) \in \delta_1$$

Why does N_2 give the reverse language?

Homomorphism

Homomorphism is a map $h : \Sigma^* \to \Gamma^*$ with the property

$$\forall u, v \in \Sigma^*. \quad h(u \cdot v) \;=\; h(u) \cdot h(v).$$

Consequences:

- $h(\epsilon) = \epsilon$.
- Giving $h(a)$ for $a \in \Sigma$ uniquely determines h.

$h(q_1 \ldots q_n) = h(q_1) h(q_2) \ldots h(q_n)$

Homomorphism

Homomorphism is a map $h : \Sigma^* \to \Gamma^*$ with the property

$$\forall u, v \in \Sigma^*.\ \ h(u \cdot v) \ = \ h(u) \cdot h(v).$$

Consequences:

- $h(\epsilon) = \epsilon$.
- Giving $h(a)$ for $a \in \Sigma$ uniquely determines h.

Theorem

If $h : \Sigma^* \to \Gamma^*$ is a homomorphism and $B \subseteq \Gamma^*$ is regular then $h^{-1}(B) \ = \ \{x \in \Sigma^* \ \mid \ h(x) \in B\}$ is also regular.

Homomorphism

Homomorphism is a map $h : \Sigma^* \to \Gamma^*$ with the property

$$\forall u, v \in \Sigma^*.\ \ h(u \cdot v) \ = \ h(u) \cdot h(v).$$

Consequences:

- $h(\epsilon) = \epsilon$.
- Giving $h(a)$ for $a \in \Sigma$ uniquely determines h.

Theorem

If $h : \Sigma^* \to \Gamma^*$ is a homomorphism and $B \subseteq \Gamma^*$ is regular then $h^{-1}(B) \ = \ \{x \in \Sigma^* \ \mid \ h(x) \in B\}$ is also regular.

Proof method

Given DFA $BB = (Q, \Gamma, \delta, q_0, F)$ for B construct DFA AA for $h^{-1}(B)$ as follows.

$$AA \ \stackrel{\text{def}}{=} \ (Q, \Sigma, \delta', q_0, F) \text{ with } \delta'(q, a) \ = \ \hat{\delta}(q, h(a)).$$

Homomorphism (2)

Theorem

If $h : \Sigma^* \to \Gamma^*$ is a homomorphism and $A \subseteq \Sigma^*$ is regular then $h(A) = \{h(x) \mid x \in A\}$ is also regular.

Homomorphism (2)

Theorem

If $h : \Sigma^* \to \Gamma^*$ is a homomorphism and $A \subseteq \Sigma^*$ is regular then $h(A) = \{h(x) \mid x \in A\}$ is also regular.

Proof method: Given regular epxression re for A, obtain $h(re)$ by substituting every letter a by $h(a)$. Then $L(h(re)) = h(A)$.

Homomorphism (2)

Theorem

If $h : \Sigma^* \to \Gamma^*$ is a homomorphism and $A \subseteq \Sigma^*$ is regular then $h(A) = \{h(x) \mid x \in A\}$ is also regular.
Proof method: Given regular epxression re for A, obtain $h(re)$ by substituting every letter a by $h(a)$. Then $L(h(re)) = h(A)$.

Theorem (projection) Given $NFA(A_1)$ over Σ and surjection $h : \Sigma \to \Gamma$, we can construct $NFA(A_2)$ over Γ s.t. $L(A_2) = h(L(A_1))$.
Proof Method: Substitute label a by $h(a)$ in each transition of A_1 to get A_2. Note that this can make DFA into NFA. We have $|A_2| = |A_1|$.

Homomorphism (2)

Theorem

If $h : \Sigma^* \to \Gamma^*$ is a homomorphism and $A \subseteq \Sigma^*$ is regular then $h(A) = \{h(x) \mid x \in A\}$ is also regular.
Proof method: Given regular epxression re for A, obtain $h(re)$ by substituting every letter a by $h(a)$. Then $L(h(re)) = h(A)$.

Theorem (projection) Given $NFA(A_1)$ over Σ and surjection $h : \Sigma \to \Gamma$, we can construct $NFA(A_2)$ over Γ s.t. $L(A_2) = h(L(A_1))$.
Proof Method: Substitute label a by $h(a)$ in each transition of A_1 to get A_2. Note that this can make DFA into NFA. We have $|A_2| = |A_1|$.

Example: Given regular A over $\{0, 1\}$ the language $hamming_3(A)$ is regular.

Decision Problems

- Membership Given NFA N and $x \in \Sigma^*$ determine if $x \in L(N)$?

Decision Problems

- Membership Given NFA N and $x \in \Sigma^*$ determine if $x \in L(N)$?
- Non-emptiness Given NFA determine if $L(N) \neq \emptyset$.

Decision Problems

- Membership Given NFA N and $x \in \Sigma^*$ determine if $x \in L(N)$?
- Non-emptiness Given NFA determine if $L(N) \neq \emptyset$.
- Language Equivalence Given NFA N_1 and N_2 determine if $L(N_1) = L(N_2)$.
  We can reduce this to checking of:
  $L(N_1) \subseteq L(N_2) \wedge L(N_2) \subseteq L(N_1)$.

Decision Problems

- Membership Given NFA N and $x \in \Sigma^*$ determine if $x \in L(N)$?
- Non-emptiness Given NFA determine if $L(N) \neq \emptyset$.
- Language Equivalence Given NFA N_1 and N_2 determine if $L(N_1) = L(N_2)$.
  We can reduce this to checking of:
  $L(N_1) \subseteq L(N_2) \wedge L(N_2) \subseteq L(N_1)$.
- Language Containment Given NFA N_1 and N_2, determine if $L(N_1) \subseteq L(N_2)$.

Membership

Given NFA N and $x \in \Sigma^*$ determine if $x \in L(N)$?

Method

- Initialize $Frontier = S$, set of start states.
- Scan the word left to right one letter at a time, doing the following:

  For the current letter a, let $Frontier = \hat{\Delta}(Frontier, a)$
- Return "accepted" if $Frontier \cap F \neq \emptyset$.

Complexity: $O(|x| \cdot |S|)$

Non-emptiness

Given NFA determine if $L(N) \neq \emptyset$.

Method: Search for a path from a start state to a final state. If path found then "non-empty".

Use Breadth-first search (BFS), or Depth-first search (DFS)

Language Containment

Given NFA N_1 and N_2, determine if $L(N_1) \subseteq L(N_2)$.
Method: Use

$$A \subseteq B \quad \text{iff} \quad (A \cap (\sim B)) = \emptyset$$

Compute Automaton $Product(N_1, Compl(N_2))$. Check if this is non-empty.