CIS 770: Formal Language Theory

Pavithra Prabhakar

Kansas State University

Spring 2015

Optimal Algorithms

Best Solutions

If a problem can be solved computationally, is there always a "best" method for solving it?

Optimal Algorithms

Best Solutions

If a problem can be solved computationally, is there always a "best" method for solving it?

Example

Merge Sort and Heap Sort achieve the assymptotically best time possible for sorting (in a certain model).

Optimal Algorithms

Manuel Blum

Best Solutions

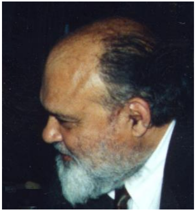
If a problem can be solved computationally, is there always a "best" method for solving it?

Example

Merge Sort and Heap Sort achieve the assymptotically best time possible for sorting (in a certain model).

Blum's Speedup Theorem

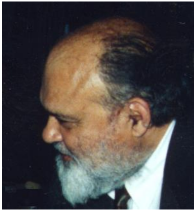
No! There are computationally solvable problems that have no optimal algorithms! In other words, there are problems for which any algorithm can always be made faster.



Anil Nerode

Myhill-Nerode Theorem

There is a "unique" "optimal" "algorithm" for every problem that can be solved using finite memory.

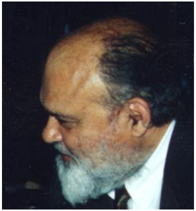


Anil Nerode

Myhill-Nerode Theorem

There is a "unique" "optimal" "algorithm" for every problem that can be solved using finite memory.

- "algorithm" here means a deterministic machine

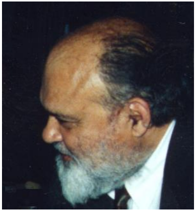


Anil Nerode

Myhill-Nerode Theorem

There is a "unique" "optimal" "algorithm" for every problem that can be solved using finite memory.

- "algorithm" here means a deterministic machine
- "optimal" means requires least memory, i.e., has fewest states



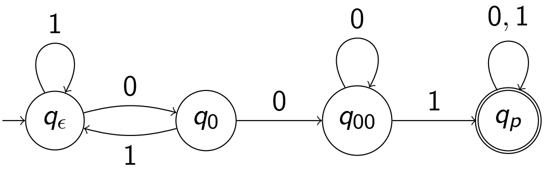
Anil Nerode

Myhill-Nerode Theorem

There is a "unique" "optimal" "algorithm" for every problem that can be solved using finite memory.

- "algorithm" here means a deterministic machine
- "optimal" means requires least memory, i.e., has fewest states
- "unique" means that any two DFAs with fewest states for a language are "isomorphic"

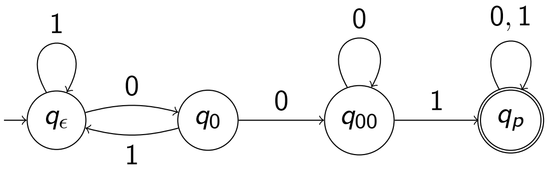
Suffix Language of a State



DFA M

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$,
$\text{suffix}(M, q) = \{w \in \Sigma^* \mid \hat{\delta}(q, w) \in F\}$.

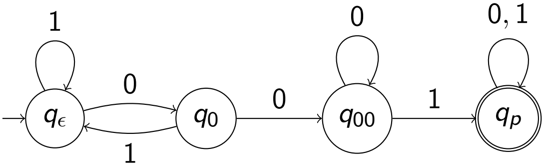
Suffix Language of a State



DFA M

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$,
$\mathrm{suffix}(M, q) = \{w \in \Sigma^* \mid \hat{\delta}(q, w) \in F\}$. In other words, it is the collection of all words accepted if q were the initial state.

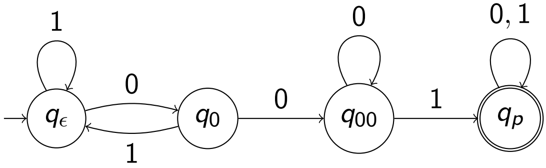
Suffix Language of a State



DFA M

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$,
$\mathrm{suffix}(M, q) = \{w \in \Sigma^* \mid \hat{\delta}(q, w) \in F\}$. In other words, it is the collection of all words accepted if q were the initial state.
For example, $\mathrm{suffix}(M, q_{00}) =$

Suffix Language of a State



DFA M

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$,
$\mathrm{suffix}(M, q) = \{w \in \Sigma^* \mid \hat{\delta}(q, w) \in F\}$. In other words, it is the collection of all words accepted if q were the initial state.
For example, $\mathrm{suffix}(M, q_{00})$ $=0^*1(0 \cup 1)^*$.

Suffix Languages

Definition

For a language $L \subseteq \Sigma^*$, and a string $x \in \Sigma^*$, the suffix language of L with respect to x, is defined as

$$\mathrm{suffix}(L, x) = \{y \in \Sigma^* \mid xy \in L\}$$

Suffix Languages

Definition

For a language $L \subseteq \Sigma^*$, and a string $x \in \Sigma^*$, the suffix language of L with respect to x, is defined as

$$\mathrm{suffix}(L, x) = \{y \in \Sigma^* \mid xy \in L\}$$

The class of suffix languages of L is

$$\mathcal{C}_{\mathrm{suf}}(L) = \{\mathrm{suffix}(L, x) \mid x \in \Sigma^*\}$$

Example: L_{odd}

Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- $\mathrm{suffix}(L_{\mathrm{odd}}, \epsilon) =$

Example: L_{odd}

Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- $\mathrm{suffix}(L_{\mathrm{odd}}, \epsilon) = L_{\mathrm{odd}}$

Example: L_{odd}

Example

Consider $L_{\text{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- $\mathrm{suffix}(L_{\text{odd}}, \epsilon) = L_{\text{odd}}$
- $\mathrm{suffix}(L_{\text{odd}}, 0) =$

Example: L_{odd}

Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- $\mathrm{suffix}(L_{\mathrm{odd}}, \epsilon) = L_{\mathrm{odd}}$
- $\mathrm{suffix}(L_{\mathrm{odd}}, 0) = L_{\mathrm{odd}}$

Example: L_{odd}

Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- $\mathrm{suffix}(L_{\mathrm{odd}}, \epsilon) = L_{\mathrm{odd}}$
- $\mathrm{suffix}(L_{\mathrm{odd}}, 0) = L_{\mathrm{odd}}$
- $\mathrm{suffix}(L_{\mathrm{odd}}, 1) =$

Example: L_{odd}

Example

Consider $L_{\text{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- $\text{suffix}(L_{\text{odd}}, \epsilon) = L_{\text{odd}}$
- $\text{suffix}(L_{\text{odd}}, 0) = L_{\text{odd}}$
- $\text{suffix}(L_{\text{odd}}, 1) = \{w \in \{0,1\}^* \mid w \text{ has an even number of 1s}\} = L_{\text{even}}$

Example: L_{odd}

Class of Suffix Languages

Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- For x that has an even number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) =$

Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- For x that has an even number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) = L_{\mathrm{odd}}$

Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- For x that has an even number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) = L_{\mathrm{odd}}$
- For x that has an odd number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) =$

Example: L_{odd}

Class of Suffix Languages

Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- For x that has an even number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) = L_{\mathrm{odd}}$
- For x that has an odd number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) = L_{\mathrm{even}}$

Example: L_{odd}

Class of Suffix Languages

Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w$ has an odd number of 1s$\}$

- For x that has an even number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) = L_{\mathrm{odd}}$
- For x that has an odd number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) = L_{\mathrm{even}}$
- Thus, $\mathcal{C}_{\mathrm{suf}}(L_{\mathrm{odd}}) =$

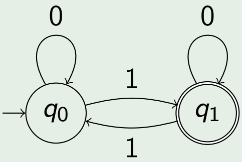
Example

Consider $L_{\mathrm{odd}} = \{w \in \{0,1\}^* \mid w \text{ has an odd number of 1s}\}$

- For x that has an even number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) = L_{\mathrm{odd}}$
- For x that has an odd number of 1s, $\mathrm{suffix}(L_{\mathrm{odd}}, x) = L_{\mathrm{even}}$
- Thus, $\mathcal{C}_{\mathrm{suf}}(L_{\mathrm{odd}}) = \{L_{\mathrm{odd}}, L_{\mathrm{even}}\}$

Example

Recall, $\mathcal{C}_{\text{suf}}(L_{\text{odd}}) = \{L_{\text{odd}}, L_{\text{even}}\}$. A DFA for L_{odd} is



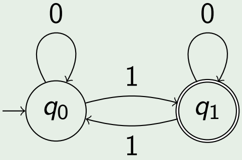
DFA for L_{odd}

Example: L_{odd}

DFA and Suffix Languages

Example

Recall, $\mathcal{C}_{\mathrm{suf}}(L_{\mathrm{odd}}) = \{L_{\mathrm{odd}}, L_{\mathrm{even}}\}$. A DFA for L_{odd} is



DFA for L_{odd}

Observe that $\mathrm{suffix}(M, q_0) = L_{\mathrm{odd}}$, and $\mathrm{suffix}(M, q_1) = L_{\mathrm{even}}$.

Example: L_{001}

Example

Consider $L_{001} = (0 \cup 1)^*001(0 \cup 1)^*$.

- $\mathrm{suffix}(L_{001}, \epsilon) =$

Example: L_{001}

Example

Consider $L_{001} = (0 \cup 1)^*001(0 \cup 1)^*$.

- $\mathrm{suffix}(L_{001}, \epsilon) = L_{001}$

Example: L_{001}

Example

Consider $L_{001} = (0 \cup 1)^*001(0 \cup 1)^*$.

- $\text{suffix}(L_{001}, \epsilon) = L_{001}$
- $\text{suffix}(L_{001}, 0) =$

Example: L_{001}

Example

Consider $L_{001} = (0 \cup 1)^* 001 (0 \cup 1)^*$.

- $\mathrm{suffix}(L_{001}, \epsilon) = L_{001}$
- $\mathrm{suffix}(L_{001}, 0) = 00^*1(0 \cup 1)^* \cup 1L_{001}$

Example: L_{001}

Example

Consider $L_{001} = (0 \cup 1)^*001(0 \cup 1)^*$.

- $\mathrm{suffix}(L_{001}, \epsilon) = L_{001}$
- $\mathrm{suffix}(L_{001}, 0) = 00^*1(0 \cup 1)^* \cup 1L_{001}$
- $\mathrm{suffix}(L_{001}, 00) =$

Example: L_{001}

Example

Consider $L_{001} = (0 \cup 1)^*001(0 \cup 1)^*$.

- $\mathrm{suffix}(L_{001}, \epsilon) = L_{001}$
- $\mathrm{suffix}(L_{001}, 0) = 00^*1(0 \cup 1)^* \cup 1L_{001}$
- $\mathrm{suffix}(L_{001}, 00) = 0^*1(0 \cup 1)^*$

Example: L_{001}

Example

Consider $L_{001} = (0 \cup 1)^* 001 (0 \cup 1)^*$.

- $\mathrm{suffix}(L_{001}, \epsilon) = L_{001}$
- $\mathrm{suffix}(L_{001}, 0) = 00^*1(0 \cup 1)^* \cup 1L_{001}$
- $\mathrm{suffix}(L_{001}, 00) = 0^*1(0 \cup 1)^*$
- $\mathrm{suffix}(L_{001}, 001) =$

Example: L_{001}

Example

Consider $L_{001} = (0 \cup 1)^*001(0 \cup 1)^*$.

- $\mathrm{suffix}(L_{001}, \epsilon) = L_{001}$
- $\mathrm{suffix}(L_{001}, 0) = 00^*1(0 \cup 1)^* \cup 1L_{001}$
- $\mathrm{suffix}(L_{001}, 00) = 0^*1(0 \cup 1)^*$
- $\mathrm{suffix}(L_{001}, 001) = (0 \cup 1)^*$

Example

Consider $L_{001} = (0 \cup 1)^* 001 (0 \cup 1)^*$.

Example

Consider $L_{001} = (0 \cup 1)^*001(0 \cup 1)^*$.

- For $x = \epsilon$ or $x \in (\Sigma^* \setminus L_{001}) \cap (0 \cup 1)^*1$, $\mathrm{suffix}(L_{001}, x) = L_{001}$

Example

Consider $L_{001} = (0 \cup 1)^*001(0 \cup 1)^*$.

- For $x = \epsilon$ or $x \in (\Sigma^* \setminus L_{001}) \cap (0 \cup 1)^*1$, $\mathrm{suffix}(L_{001}, x) = L_{001}$
- For $x \in (\Sigma^* \setminus L_{001}) \cap [0 \cup (0 \cup 1)^*10]$, $\mathrm{suffix}(L_{001}, x) = 00^*1(0 \cup 1)^* \cup 1L_{001}$

Example

Consider $L_{001} = (0 \cup 1)^*001(0 \cup 1)^*$.

- For $x = \epsilon$ or $x \in (\Sigma^* \setminus L_{001}) \cap (0 \cup 1)^*1$, $\mathrm{suffix}(L_{001}, x) = L_{001}$
- For $x \in (\Sigma^* \setminus L_{001}) \cap [0 \cup (0 \cup 1)^*10]$, $\mathrm{suffix}(L_{001}, x) = 00^*1(0 \cup 1)^* \cup 1L_{001}$
- For $x \in (\Sigma^* \setminus L_{001}) \cap (0 \cup 1)^*00$, $\mathrm{suffix}(L_{001}, x) = 0^*1(0 \cup 1)^*$

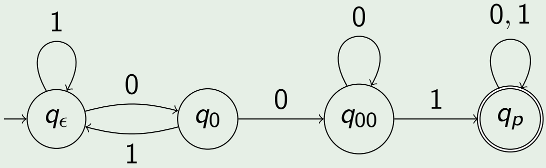
Example

Consider $L_{001} = (0 \cup 1)^* 001 (0 \cup 1)^*$.

- For $x = \epsilon$ or $x \in (\Sigma^* \setminus L_{001}) \cap (0 \cup 1)^* 1$, $\mathrm{suffix}(L_{001}, x) = L_{001}$
- For $x \in (\Sigma^* \setminus L_{001}) \cap [0 \cup (0 \cup 1)^* 10]$, $\mathrm{suffix}(L_{001}, x) = 00^* 1 (0 \cup 1)^* \cup 1 L_{001}$
- For $x \in (\Sigma^* \setminus L_{001}) \cap (0 \cup 1)^* 00$, $\mathrm{suffix}(L_{001}, x) = 0^* 1 (0 \cup 1)^*$
- For $x \in L_{001}$, $\mathrm{suffix}(L_{001}, x) = (0 \cup 1)^*$

Example

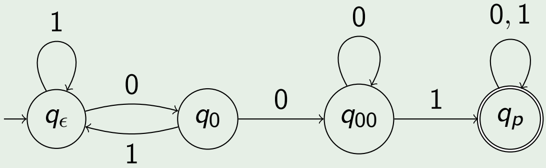
A DFA for L_{001} is



DFA for L_{odd}

Example

A DFA for L_{001} is



DFA for L_{odd}

Observe that the suffix languages of the states correspond to the class of suffix languages.

Example: L_{0n1n}

Example

Consider $L_{0n1n} = \{0^n 1^n \mid n \geq 0\}$.

Example: L_{0n1n}

Example

Consider $L_{0n1n} = \{0^n 1^n \mid n \geq 0\}$.

- $\mathrm{suffix}(L_{0n1n}, 0) =$

Example: L_{0n1n}

Example

Consider $L_{0n1n} = \{0^n 1^n \mid n \geq 0\}$.

- $\mathrm{suffix}(L_{0n1n}, 0) = \{0^{n-1} 1^n \mid n \geq 1\}$

Example: L_{0n1n}

Example

Consider $L_{0n1n} = \{0^n 1^n \mid n \geq 0\}$.

- $\mathrm{suffix}(L_{0n1n}, 0) = \{0^{n-1} 1^n \mid n \geq 1\}$
- $\mathrm{suffix}(L_{0n1n}, 0^i) =$

Example: L_{0n1n}

Example

Consider $L_{0n1n} = \{0^n 1^n \mid n \geq 0\}$.

- $\mathrm{suffix}(L_{0n1n}, 0) = \{0^{n-1} 1^n \mid n \geq 1\}$
- $\mathrm{suffix}(L_{0n1n}, 0^i) = \{0^{n-i} 1^n \mid n \geq i\}$

Example: L_{0n1n}

Example

Consider $L_{0n1n} = \{0^n1^n \mid n \geq 0\}$.

- $\mathrm{suffix}(L_{0n1n}, 0) = \{0^{n-1}1^n \mid n \geq 1\}$
- $\mathrm{suffix}(L_{0n1n}, 0^i) = \{0^{n-i}1^n \mid n \geq i\}$

Proposition

$\mathcal{C}_{\mathrm{suf}}(L_{0n1n})$ *has infinitely many languages.*

Proof.

Observe that for $i \neq j$, $\mathrm{suffix}(L_{0n1n}, 0^i) \neq \mathrm{suffix}(L_{0n1n}, 0^j)$. $\square$

Recap . . .

Observations

In the previous examples,

Recap . . .

Observations

In the previous examples,

- For regular L,

Recap . . .

Observations

In the previous examples,

- For regular L,
  - $\mathcal{C}_{\mathrm{suf}}(L)$ has only finitely many languages

Recap ...

Observations

In the previous examples,

- For regular L,
  - $\mathcal{C}_{\mathrm{suf}}(L)$ has only finitely many languages
  - There is a DFA D for L, the suffix languages of the states of D, correspond to the languages in $\mathcal{C}_{\mathrm{suf}}(L)$

Recap ...

Observations

In the previous examples,

- For regular L,
  - $\mathcal{C}_{\mathrm{suf}}(L)$ has only finitely many languages
  - There is a DFA D for L, the suffix languages of the states of D, correspond to the languages in $\mathcal{C}_{\mathrm{suf}}(L)$
- For non-regular L,

Recap . . .

Observations

In the previous examples,

- For regular L,
  - $\mathcal{C}_{\mathrm{suf}}(L)$ has only finitely many languages
  - There is a DFA D for L, the suffix languages of the states of D, correspond to the languages in $\mathcal{C}_{\mathrm{suf}}(L)$
- For non-regular L,
  - $\mathcal{C}_{\mathrm{suf}}(L)$ has infinitely many languages

Recap . . .

Observations

In the previous examples,

- For regular L,
  - $\mathcal{C}_{\mathrm{suf}}(L)$ has only finitely many languages
  - There is a DFA D for L, the suffix languages of the states of D, correspond to the languages in $\mathcal{C}_{\mathrm{suf}}(L)$
- For non-regular L,
  - $\mathcal{C}_{\mathrm{suf}}(L)$ has infinitely many languages

Are these observations true in general?

States and Suffix Languages

Proposition

Let $M = (Q, \Sigma, \delta, q_0, F)$ *and let* $L = L(M)$*. Then if* $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$ *(i.e., both* x *and* y *take* M *to the same state), then* $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$*.*

Proof.

$\square$

States and Suffix Languages

Proposition

Let $M = (Q, \Sigma, \delta, q_0, F)$ *and let* $L = L(M)$. *Then if* $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$ *(i.e., both* x *and* y *take* M *to the same state), then* $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$.

Proof.

Consider $z \in \mathrm{suffix}(L, x)$. Then $xz \in L$.

$\square$

States and Suffix Languages

Proposition

Let $M = (Q, \Sigma, \delta, q_0, F)$ and let $L = L(M)$. Then if $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$ (i.e., both x and y take M to the same state), then $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$.

Proof.

Consider $z \in \mathrm{suffix}(L, x)$. Then $xz \in L$.

$$\hat{\delta}(q_0, xz)(\in F) \quad = \hat{\delta}(\hat{\delta}(q_0, x), z) \qquad \text{(prop. } \hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v)\text{)}$$

□

States and Suffix Languages

Proposition

Let $M = (Q, \Sigma, \delta, q_0, F)$ *and let* $L = L(M)$*. Then if* $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$ *(i.e., both* x *and* y *take* M *to the same state), then* $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$*.*

Proof.

Consider $z \in \mathrm{suffix}(L, x)$. Then $xz \in L$.

$$\begin{aligned}
\hat{\delta}(q_0, xz)(\in F) &= \hat{\delta}(\hat{\delta}(q_0, x), z) && (\text{prop. } \hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v)) \\
&= \hat{\delta}(\hat{\delta}(q_0, y), z) && (\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y))
\end{aligned}$$

$\square$

States and Suffix Languages

Proposition

Let $M = (Q, \Sigma, \delta, q_0, F)$ *and let* $L = L(M)$*. Then if* $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$ *(i.e., both* x *and* y *take* M *to the same state), then* $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$*.*

Proof.

Consider $z \in \mathrm{suffix}(L, x)$. Then $xz \in L$.

$$\begin{aligned}
\hat{\delta}(q_0, xz)(\in F) &= \hat{\delta}(\hat{\delta}(q_0, x), z) && (\text{prop. } \hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v)) \\
&= \hat{\delta}(\hat{\delta}(q_0, y), z) && (\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)) \\
&= \hat{\delta}(q_0, yz)(\in F) && (\text{prop. } \hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v))
\end{aligned}$$

$\square$

States and Suffix Languages

Proposition

Let $M = (Q, \Sigma, \delta, q_0, F)$ *and let* $L = L(M)$*. Then if* $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$ *(i.e., both* x *and* y *take* M *to the same state), then* $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$*.*

Proof.

Consider $z \in \mathrm{suffix}(L, x)$. Then $xz \in L$.

$$\begin{aligned}
\hat{\delta}(q_0, xz)(\in F) &= \hat{\delta}(\hat{\delta}(q_0, x), z) && (\text{prop. } \hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v)) \\
&= \hat{\delta}(\hat{\delta}(q_0, y), z) && (\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)) \\
&= \hat{\delta}(q_0, yz)(\in F) && (\text{prop. } \hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v))
\end{aligned}$$

Similarly, we can show that if $z \in \mathrm{suffix}(L, y)$ then $z \in \mathrm{suffix}(L, x)$. $\square$

Regularity and Suffix Languages

Corollary

If L is regular then $\mathcal{C}_{\mathrm{suf}}(L)$ is finite.

Regularity and Suffix Languages

Corollary

If L is regular then $\mathcal{C}_{\mathrm{suf}}(L)$ is finite.

Proof.

If L is regular then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L = L(M)$.

$\square$

Regularity and Suffix Languages

Corollary

If L is regular then $\mathcal{C}_{\rm suf}(L)$ is finite.

Proof.

If L is regular then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L = L(M)$.

- We have shown that, if x, y reach the same state in M then $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$

$\square$

Regularity and Suffix Languages

Corollary

If L is regular then $\mathcal{C}_{\mathrm{suf}}(L)$ is finite.

Proof.

If L is regular then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L = L(M)$.

- We have shown that, if x, y reach the same state in M then $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$
- Thus, $|\mathcal{C}_{\mathrm{suf}}(L)| \leq |Q|$, which is finite.

$\square$

Canonical DFAs for Regular Languages

Proposition

For a language $L \subseteq \Sigma^*$, if $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then there is a DFA M^L such that $L(M^L) = L$.

Canonical DFAs for Regular Languages

Proposition

For a language $L \subseteq \Sigma^*$, if $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then there is a DFA M^L such that $L(M^L) = L$.

Proof.

Define $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$ as follows.

Canonical DFAs for Regular Languages

Proposition

For a language $L \subseteq \Sigma^*$, if $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then there is a DFA M^L such that $L(M^L) = L$.

Proof.

Define $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$ as follows.

- $Q^L = \mathcal{C}_{\mathrm{suf}}(L)$

Canonical DFAs for Regular Languages

Proposition

For a language $L \subseteq \Sigma^*$, if $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then there is a DFA M^L such that $L(M^L) = L$.

Proof.

Define $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$ as follows.

- $Q^L = \mathcal{C}_{\mathrm{suf}}(L)$
- $q_0^L = \mathrm{suffix}(L, \epsilon)$

Canonical DFAs for Regular Languages

Proposition

For a language $L \subseteq \Sigma^*$, if $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then there is a DFA M^L such that $L(M^L) = L$.

Proof.

Define $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$ as follows.

- $Q^L = \mathcal{C}_{\mathrm{suf}}(L)$
- $q_0^L = \mathrm{suffix}(L, \epsilon)$
- $F^L = \{\mathrm{suffix}(L, x) \mid \epsilon \in \mathrm{suffix}(L, x)\}$

Canonical DFAs for Regular Languages

Proposition

For a language $L \subseteq \Sigma^*$, if $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then there is a DFA M^L such that $L(M^L) = L$.

Proof.

Define $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$ as follows.

- $Q^L = \mathcal{C}_{\mathrm{suf}}(L)$
- $q_0^L = \mathrm{suffix}(L, \epsilon)$
- $F^L = \{\mathrm{suffix}(L, x) \mid \epsilon \in \mathrm{suffix}(L, x)\}$
- $\delta^L(\mathrm{suffix}(L, x), a) = \mathrm{suffix}(L, xa)$

Canonical DFAs for Regular Languages

Proposition

For a language $L \subseteq \Sigma^*$, if $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then there is a DFA M^L such that $L(M^L) = L$.

Proof.

Define $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$ as follows.

- $Q^L = \mathcal{C}_{\mathrm{suf}}(L)$
- $q_0^L = \mathrm{suffix}(L, \epsilon)$
- $F^L = \{\mathrm{suffix}(L, x) \mid \epsilon \in \mathrm{suffix}(L, x)\}$
- $\delta^L(\mathrm{suffix}(L, x), a) = \mathrm{suffix}(L, xa)$

Is δ well-defined?

Canonical DFAs for Regular Languages

Proposition

For a language $L \subseteq \Sigma^*$, if $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then there is a DFA M^L such that $L(M^L) = L$.

Proof.

Define $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$ as follows.

- $Q^L = \mathcal{C}_{\mathrm{suf}}(L)$
- $q_0^L = \mathrm{suffix}(L, \epsilon)$
- $F^L = \{\mathrm{suffix}(L, x) \mid \epsilon \in \mathrm{suffix}(L, x)\}$
- $\delta^L(\mathrm{suffix}(L, x), a) = \mathrm{suffix}(L, xa)$

Is δ well-defined? Same state can have multiple names (i.e., x, y s.t. $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$). $\cdot\!\cdot\!\longrightarrow$

Canonical DFAs for Regular Languages

Transition function is well-defined

Proof (contd).

- $\delta^L(\mathrm{suffix}(L, x), a) = \mathrm{suffix}(L, xa)$

Canonical DFAs for Regular Languages

Transition function is well-defined

Proof (contd).

- $\delta^L(\mathrm{suffix}(L, x), a) = \mathrm{suffix}(L, xa)$
- δ is well-defined

Proof (contd).

- $\delta^L(\mathrm{suffix}(L, x), a) = \mathrm{suffix}(L, xa)$
- δ is well-defined because if $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$ then, for all $a \in \Sigma$, $\mathrm{suffix}(L, xa) = \mathrm{suffix}(L, ya)$.

Canonical DFAs for Regular Languages

Transition function is well-defined

Proof (contd).

- $\delta^L(\mathrm{suffix}(L, x), a) = \mathrm{suffix}(L, xa)$
- δ is well-defined because if $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$ then, for all $a \in \Sigma$, $\mathrm{suffix}(L, xa) = \mathrm{suffix}(L, ya)$.
  - Suppose $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$. Then,

$$\begin{aligned} z \in \mathrm{suffix}(L, xa) &\iff xaz \in L \\ &\iff az \in \mathrm{suffix}(L, x) = \mathrm{suffix}(L, y) \\ &\iff yaz \in L \iff z \in \mathrm{suffix}(L, ya) \end{aligned}$$

Proof (contd).

- $\delta^L(\mathrm{suffix}(L, x), a) = \mathrm{suffix}(L, xa)$
- δ is well-defined because if $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$ then, for all $a \in \Sigma$, $\mathrm{suffix}(L, xa) = \mathrm{suffix}(L, ya)$.
  - Suppose $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$. Then,

$$
\begin{aligned}
z \in \mathrm{suffix}(L, xa) &\iff xaz \in L \\
&\iff az \in \mathrm{suffix}(L, x) = \mathrm{suffix}(L, y) \\
&\iff yaz \in L \iff z \in \mathrm{suffix}(L, ya)
\end{aligned}
$$

  - Hence, $\mathrm{suffix}(L, xa) = \mathrm{suffix}(L, ya)$.

Canonical DFAs for Regular Languages

Proof (contd).

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \mathrm{suffix}(L, x)$.

Canonical DFAs for Regular Languages

Proof (contd).

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \mathrm{suffix}(L, x)$.
  - $q_0^L = \mathrm{suffix}(L, \epsilon)$

Proof (contd).

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \mathrm{suffix}(L, x)$.
  - $q_0^L = \mathrm{suffix}(L, \epsilon) \xrightarrow{x_1}_{M^L} \mathrm{suffix}(L, x_1)$

Canonical DFAs for Regular Languages

Proof (contd).

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \mathrm{suffix}(L, x)$.
    - $q_0^L = \mathrm{suffix}(L, \epsilon) \xrightarrow{x_1}_{M^L} \mathrm{suffix}(L, x_1) \xrightarrow{x_2}_{M^L} \mathrm{suffix}(L, x_1 x_2)$

Canonical DFAs for Regular Languages

Proof (contd).

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \mathrm{suffix}(L, x)$.
  - $q_0^L = \mathrm{suffix}(L, \epsilon) \xrightarrow{x_1}_{M^L} \mathrm{suffix}(L, x_1) \xrightarrow{x_2}_{M^L} \mathrm{suffix}(L, x_1 x_2) \ldots \xrightarrow{x_n}_{M^L} \mathrm{suffix}(L, x)$

Canonical DFAs for Regular Languages

Proof (contd).

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \mathrm{suffix}(L, x)$.
  - $q_0^L = \mathrm{suffix}(L, \epsilon) \xrightarrow{x_1}_{M^L} \mathrm{suffix}(L, x_1) \xrightarrow{x_2}_{M^L} \mathrm{suffix}(L, x_1x_2) \ldots \xrightarrow{x_n}_{M^L} \mathrm{suffix}(L, x)$
  - Formally, by induction on $|x|$

Canonical DFAs for Regular Languages

Proof (contd).

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \mathrm{suffix}(L, x)$.
  - $q_0^L = \mathrm{suffix}(L, \epsilon) \xrightarrow{x_1}_{M^L} \mathrm{suffix}(L, x_1) \xrightarrow{x_2}_{M^L}$ $\mathrm{suffix}(L, x_1x_2) \ldots \xrightarrow{x_n}_{M^L} \mathrm{suffix}(L, x)$
  - Formally, by induction on $|x|$
- $x \in L$ iff $\epsilon \in \mathrm{suffix}(L, x)$

Canonical DFAs for Regular Languages

Proof (contd).

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \mathrm{suffix}(L, x)$.
  - $q_0^L = \mathrm{suffix}(L, \epsilon) \xrightarrow{x_1}_{M^L} \mathrm{suffix}(L, x_1) \xrightarrow{x_2}_{M^L} \mathrm{suffix}(L, x_1x_2) \ldots \xrightarrow{x_n}_{M^L} \mathrm{suffix}(L, x)$
  - Formally, by induction on $|x|$
- $x \in L$ iff $\epsilon \in \mathrm{suffix}(L, x)$ iff $\mathrm{suffix}(L, x) \in F^L$

Proof (contd).

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \text{suffix}(L, x)$.
  - $q_0^L = \text{suffix}(L, \epsilon) \xrightarrow{x_1}_{M^L} \text{suffix}(L, x_1) \xrightarrow{x_2}_{M^L} \text{suffix}(L, x_1x_2) \ldots \xrightarrow{x_n}_{M^L} \text{suffix}(L, x)$
  - Formally, by induction on $|x|$
- $x \in L$ iff $\epsilon \in \text{suffix}(L, x)$ iff $\text{suffix}(L, x) \in F^L$
- Hence, $x \in L$ iff M^L accepts x. $\square$

Example of Canonical DFA

Example

Consider $L_{001} = (0 \cup 1)^* 001 (0 \cup 1)^*$. Recall that the suffix languges are

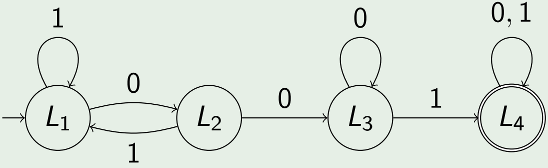
$$
\begin{aligned}
L_1 &= L_{001} = \mathrm{suffix}(L_{001}, \epsilon) = \mathrm{suffix}(L_{001}, (0 \cup \epsilon)1) \\
L_2 &= 00^*1(0 \cup 1)^* \cup 1L_{001} = \mathrm{suffix}(L_{001}, 0) \\
L_3 &= 0^*1(0 \cup 1)^* = \mathrm{suffix}(L_{001}, 00) = \mathrm{suffix}(L_{001}, 000) \\
L_4 &= (0 \cup 1)^* = \mathrm{suffix}(L_{001}, 001) = \mathrm{suffix}(L_{001}, 001(0 \cup 1))
\end{aligned}
$$

Example of Canonical DFA

Example

Consider $L_{001} = (0 \cup 1)^* 001 (0 \cup 1)^*$. Recall that the suffix languges are

$$\begin{aligned}
L_1 &= L_{001} = \mathrm{suffix}(L_{001}, \epsilon) = \mathrm{suffix}(L_{001}, (0 \cup \epsilon)1) \\
L_2 &= 00^*1(0 \cup 1)^* \cup 1L_{001} = \mathrm{suffix}(L_{001}, 0) \\
L_3 &= 0^*1(0 \cup 1)^* = \mathrm{suffix}(L_{001}, 00) = \mathrm{suffix}(L_{001}, 000) \\
L_4 &= (0 \cup 1)^* = \mathrm{suffix}(L_{001}, 001) = \mathrm{suffix}(L_{001}, 001(0 \cup 1))
\end{aligned}$$



Canonical DFA is the smallest DFA

Proposition

For any regular language L, M^L is the unique smallest DFA that recognizes L

Canonical DFA is the smallest DFA

Proposition

For any regular language L, M^L is the unique smallest DFA that recognizes L upto isomorphism.

Isomorphism

Definition

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be two DFAs. A function $f : Q_1 \to Q_2$ is said to be isomorphism iff

- f is bijective, i.e., one-to-one and onto
- $f(q_1) = q_2$
- For every $p \in Q_1$ and $a \in \Sigma$, $f(\delta_1(p, a)) = \delta_2(f(p), a)$
- $q \in F_1$ iff $f(q) \in F_2$

Isomorphism

Definition

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be two DFAs. A function $f : Q_1 \to Q_2$ is said to be isomorphism iff

- f is bijective, i.e., one-to-one and onto
- $f(q_1) = q_2$
- For every $p \in Q_1$ and $a \in \Sigma$, $f(\delta_1(p, a)) = \delta_2(f(p), a)$
- $q \in F_1$ iff $f(q) \in F_2$

M_1 and M_2 are said to be isomorphic if there is an isomorphism f from M_1 to M_2.

Isomorphism

Definition

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be two DFAs. A function $f : Q_1 \to Q_2$ is said to be isomorphism iff

- f is bijective, i.e., one-to-one and onto
- $f(q_1) = q_2$
- For every $p \in Q_1$ and $a \in \Sigma$, $f(\delta_1(p, a)) = \delta_2(f(p), a)$
- $q \in F_1$ iff $f(q) \in F_2$

M_1 and M_2 are said to be isomorphic if there is an isomorphism f from M_1 to M_2.
Thus, if M_1 and M_2 are isomorphic then they are the "same" machine except for possibly renaming states.

Canonical DFA is the smallest DFA

Proposition

For any regular language L, M^L is the unique (upto isomorphism) smallest DFA that recognizes L.

Canonical DFA is the smallest DFA

Proposition

For any regular language L, M^L is the unique (upto isomorphism) smallest DFA that recognizes L.

Proof.

Recall $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be some DFA such that $L(M) = L$.

Canonical DFA is the smallest DFA

Proposition

For any regular language L, M^L is the unique (upto isomorphism) smallest DFA that recognizes L.

Proof.

Recall $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be some DFA such that $L(M) = L$.

- Define $f : Q \to Q^L$ as: $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

Canonical DFA is the smallest DFA

Proposition

For any regular language L, M^L is the unique (upto isomorphism) smallest DFA that recognizes L.

Proof.

Recall $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be some DFA such that $L(M) = L$.

- Define $f : Q \to Q^L$ as: $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.
  - f is well-defined, because we showed that if x, y both take M to q, then $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$.

Canonical DFA is the smallest DFA

Proposition

For any regular language L, M^L is the unique (upto isomorphism) smallest DFA that recognizes L.

Proof.

Recall $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be some DFA such that $L(M) = L$.

- Define $f : Q \to Q^L$ as: $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.
  - f is well-defined, because we showed that if x, y both take M to q, then $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$.
- f is onto because

Canonical DFA is the smallest DFA

Proposition

For any regular language L, M^L is the unique (upto isomorphism) smallest DFA that recognizes L.

Proof.

Recall $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be some DFA such that $L(M) = L$.

- Define $f : Q \to Q^L$ as: $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.
  - f is well-defined, because we showed that if x, y both take M to q, then $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$.
- f is onto because a state $\mathrm{suffix}(L, x) \in Q^L$ is the image of q' under f, where $\hat{\delta}_M(q_0, x) = q'$.

Canonical DFA is the smallest DFA

Proposition

For any regular language L, M^L is the unique (upto isomorphism) smallest DFA that recognizes L.

Proof.

Recall $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be some DFA such that $L(M) = L$.

- Define $f : Q \to Q^L$ as: $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.
  - f is well-defined, because we showed that if x, y both take M to q, then $\mathrm{suffix}(L, x) = \mathrm{suffix}(L, y)$.
- f is onto because a state $\mathrm{suffix}(L, x) \in Q^L$ is the image of q' under f, where $\hat{\delta}_M(q_0, x) = q'$.
- Thus, $|Q| \geq |Q^L|$ $\cdots\!\longrightarrow$

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic.

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic. Recall, $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic. Recall, $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

- Since f is onto, f must be one-to-one. Thus, f is bijective.

Canonical DFA is the smallest DFA

f preserves transitions

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic. Recall, $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

- Since f is onto, f must be one-to-one. Thus, f is bijective.
- Since $\hat{\delta}_M(q_0, \epsilon) = q_0$, $f(q_0) = \mathrm{suffix}(L, \epsilon) = q_0^L$

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic. Recall, $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

- Since f is onto, f must be one-to-one. Thus, f is bijective.
- Since $\hat{\delta}_M(q_0, \epsilon) = q_0$, $f(q_0) = \mathrm{suffix}(L, \epsilon) = q_0^L$
- Suppose $\delta_M(q, a) = q'$ and $f(q) = \mathrm{suffix}(L, x)$.

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic. Recall, $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

- Since f is onto, f must be one-to-one. Thus, f is bijective.
- Since $\hat{\delta}_M(q_0, \epsilon) = q_0$, $f(q_0) = \mathrm{suffix}(L, \epsilon) = q_0^L$
- Suppose $\delta_M(q, a) = q'$ and $f(q) = \mathrm{suffix}(L, x)$. Then, $\hat{\delta}_M(q_0, xa) = \delta(\hat{\delta}(q_0, x), a) = \delta(q, a) = q'$.

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic. Recall, $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

- Since f is onto, f must be one-to-one. Thus, f is bijective.
- Since $\hat{\delta}_M(q_0, \epsilon) = q_0$, $f(q_0) = \mathrm{suffix}(L, \epsilon) = q_0^L$
- Suppose $\delta_M(q, a) = q'$ and $f(q) = \mathrm{suffix}(L, x)$. Then, $\hat{\delta}_M(q_0, xa) = \delta(\hat{\delta}(q_0, x), a) = \delta(q, a) = q'$. Thus, $f(q') = \mathrm{suffix}(L, xa) = \delta^L(\mathrm{suffix}(L, x), a)$.

Canonical DFA is the smallest DFA

f preserves transitions

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic. Recall, $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

- Since f is onto, f must be one-to-one. Thus, f is bijective.
- Since $\hat{\delta}_M(q_0, \epsilon) = q_0$, $f(q_0) = \mathrm{suffix}(L, \epsilon) = q_0^L$
- Suppose $\delta_M(q, a) = q'$ and $f(q) = \mathrm{suffix}(L, x)$. Then, $\hat{\delta}_M(q_0, xa) = \delta(\hat{\delta}(q_0, x), a) = \delta(q, a) = q'$. Thus, $f(q') = \mathrm{suffix}(L, xa) = \delta^L(\mathrm{suffix}(L, x), a)$.
- Suppose $q \in F$, and $f(q) = \mathrm{suffix}(L, x)$.

Canonical DFA is the smallest DFA

f preserves transitions

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic. Recall, $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

- Since f is onto, f must be one-to-one. Thus, f is bijective.
- Since $\hat{\delta}_M(q_0, \epsilon) = q_0$, $f(q_0) = \mathrm{suffix}(L, \epsilon) = q_0^L$
- Suppose $\delta_M(q, a) = q'$ and $f(q) = \mathrm{suffix}(L, x)$. Then, $\hat{\delta}_M(q_0, xa) = \delta(\hat{\delta}(q_0, x), a) = \delta(q, a) = q'$. Thus, $f(q') = \mathrm{suffix}(L, xa) = \delta^L(\mathrm{suffix}(L, x), a)$.
- Suppose $q \in F$, and $f(q) = \mathrm{suffix}(L, x)$. Hence, $\hat{\delta}_M(q_0, x) = q$, and so $x \in L$, and $\epsilon \in \mathrm{suffix}(L, x)$.

Canonical DFA is the smallest DFA

f preserves transitions

Proof (contd).

Suppose $|Q| = |Q^L|$. Then we need to show that M and M^L are isomorphic. Recall, $f(q) = \mathrm{suffix}(L, x)$ iff $\hat{\delta}_M(q_0, x) = q$.

- Since f is onto, f must be one-to-one. Thus, f is bijective.
- Since $\hat{\delta}_M(q_0, \epsilon) = q_0$, $f(q_0) = \mathrm{suffix}(L, \epsilon) = q_0^L$
- Suppose $\delta_M(q, a) = q'$ and $f(q) = \mathrm{suffix}(L, x)$. Then, $\hat{\delta}_M(q_0, xa) = \delta(\hat{\delta}(q_0, x), a) = \delta(q, a) = q'$. Thus, $f(q') = \mathrm{suffix}(L, xa) = \delta^L(\mathrm{suffix}(L, x), a)$.
- Suppose $q \in F$, and $f(q) = \mathrm{suffix}(L, x)$. Hence, $\hat{\delta}_M(q_0, x) = q$, and so $x \in L$, and $\epsilon \in \mathrm{suffix}(L, x)$. Thus, $f(q) \in F^L$. The converse holds by reversing this argument, and so f is an isomorphism. $\square$

Myhill-Nerode Theorem

In summary . . .

Theorem

The following facts are true.

1. *If L is regular then $\mathcal{C}_{\mathrm{suf}}(L)$ is finite.*

Myhill-Nerode Theorem

In summary . . .

Theorem

The following facts are true.

1. *If L is regular then $\mathcal{C}_{\mathrm{suf}}(L)$ is finite.*
2. *If $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then L is regular;*

Theorem

The following facts are true.

1. *If L is regular then $\mathcal{C}_{\mathrm{suf}}(L)$ is finite.*
2. *If $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then L is regular; more precisely, there is a DFA M^L, whose states are the languages in $\mathcal{C}_{\mathrm{suf}}(L)$, such that $L(M^L) = L$.*

Theorem

The following facts are true.

1. *If L is regular then $\mathcal{C}_{\mathrm{suf}}(L)$ is finite.*
2. *If $\mathcal{C}_{\mathrm{suf}}(L)$ is finite then L is regular; more precisely, there is a DFA M^L, whose states are the languages in $\mathcal{C}_{\mathrm{suf}}(L)$, such that $L(M^L) = L$.*
3. *For any regular language L, M^L is the unique (upto isomorphism) DFA with fewest states that recognizes L.*

Minimization

Problem

Given a DFA M, construct the DFA with fewest states M' such that $L(M') = L(M)$.

Minimization

Problem

Given a DFA M, construct the DFA with fewest states M' such that $L(M') = L(M)$.

Applications

Algorithms using DFAs run in time directly related to the number of states of DFA. Implementation of the DFA itself takes memory proportional to log number of states. So constructing small DFAs is very critical.

Ideas

Let $M = (Q, \Sigma, \delta, q_0, F)$, and $L = L(M)$. Recall that from the proof of Myhill-Nerode Theorem

Ideas

Let $M = (Q, \Sigma, \delta, q_0, F)$, and $L = L(M)$. Recall that from the proof of Myhill-Nerode Theorem

- M^L is the unique smallest DFA

Ideas

Let $M = (Q, \Sigma, \delta, q_0, F)$, and $L = L(M)$. Recall that from the proof of Myhill-Nerode Theorem

- M^L is the unique smallest DFA
- The function $f : Q \to Q^L$ maps a state q to the state of Q^L that corresponds to the language $\mathrm{suffix}(M, q)$.

Ideas

Let $M = (Q, \Sigma, \delta, q_0, F)$, and $L = L(M)$. Recall that from the proof of Myhill-Nerode Theorem

- M^L is the unique smallest DFA
- The function $f : Q \to Q^L$ maps a state q to the state of Q^L that corresponds to the language $\mathrm{suffix}(M, q)$.

Thus M can be "minimized" by collapsing states q_1 and q_2 if $\mathrm{suffix}(M, q_1) = \mathrm{suffix}(M, q_2)$.

Distinguishability

When must two states p and q of M not be collapsed?

$$\mathrm{suffix}(M, p) \neq \mathrm{suffix}(M, q)$$

Distinguishability

When must two states p and q of M not be collapsed?

$\mathrm{suffix}(M, p) \neq \mathrm{suffix}(M, q)$ iff
$\exists w.\ w \in \mathrm{suffix}(M, p)$ and $w \notin \mathrm{suffix}(M, q)$(or vice versa)

Distinguishability

When must two states p and q of M not be collapsed?

$\mathrm{suffix}(M, p) \neq \mathrm{suffix}(M, q)$ iff
$\exists w.\ w \in \mathrm{suffix}(M, p)$ and $w \notin \mathrm{suffix}(M, q)$(or vice versa) iff
$\exists w.\ \hat{\delta}(p, w) \in F$ and $\hat{\delta}(q, w) \notin F$

Distinguishability

When must two states p and q of M not be collapsed?

$\mathrm{suffix}(M, p) \neq \mathrm{suffix}(M, q)$ iff
$\exists w.\ w \in \mathrm{suffix}(M, p)$ and $w \notin \mathrm{suffix}(M, q)$(or vice versa) iff
$\exists w.\ \hat{\delta}(p, w) \in F$ and $\hat{\delta}(q, w) \notin F$

We will say that p and q are distinguishable when this happens.

Distinguishability can be inductively defined as follows

Distinguishability can be inductively defined as follows

- If $p \in F$ and $q \notin F$ then p and q are distinuishable

Distinguishability

Inductive Definition

Distinguishability can be inductively defined as follows

- If $p \in F$ and $q \notin F$ then p and q are distinuishable
- If for some a, $\delta(p, a) = p'$ and $\delta(q, a) = q'$, and p' and q' are distinguishable, then p and q are distinguishable

Let `distinct` be a table with an entry for each pair of states. Initially all entries are 0.

```
if p ∈ F and q ∉ F (or vice versa)
then distinct(p, q) := 1
repeat
for each pair (p,q) and symbol a
    if distinct(δ(p,a), δ(q,a)) = 1,
    then distinct(p, q) := 1
until no changes in table
```

Minimization Algorithm

1. Remove states that are not reachable from the initial state

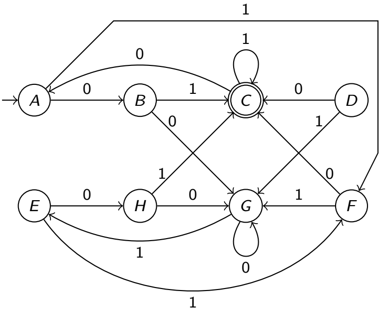
Minimization Algorithm

1. Remove states that are not reachable from the initial state
2. Find all pairs of states that are distinguishable

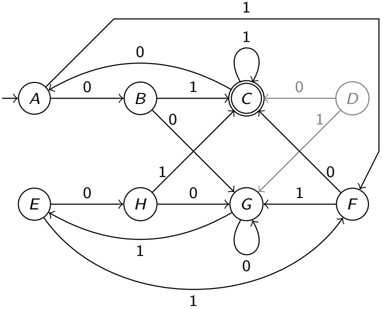
Minimization Algorithm

1. Remove states that are not reachable from the initial state
2. Find all pairs of states that are distinguishable
3. Collapse pairs that are not distinguishable

Example

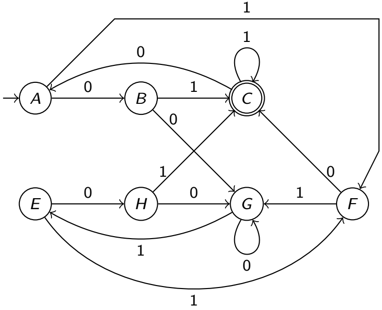


Example



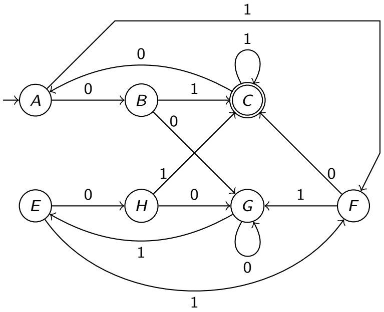
Example

	A	B	C	E	F	G
B						
C	⋆	⋆				
E			⋆			
F			⋆			
G			⋆			
H			⋆			



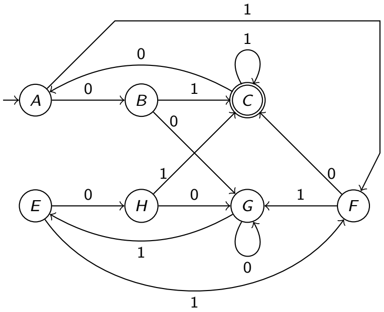
Example

	A	B	C	E	F	G
B	⋆					
C	⋆	⋆				
E		⋆	⋆			
F	⋆	⋆	⋆	⋆		
G		⋆	⋆		⋆	
H	⋆		⋆	⋆	⋆	⋆



Example

	A	B	C	E	F	G
B	⋆					
C	⋆	⋆				
E		⋆	⋆			
F	⋆	⋆	⋆	⋆		
G	⋆	⋆	⋆	⋆	⋆	
H	⋆		⋆	⋆	⋆	⋆



Example

	A	B	C	E	F	G
B	⋆					
C	⋆	⋆				
E		⋆	⋆			
F	⋆	⋆	⋆	⋆		
G	⋆	⋆	⋆	⋆	⋆	
H	⋆		⋆	⋆	⋆	⋆