universität innsbruck

WS 2024 lecture 5

Automata and Logic

Aart Middeldorp and Johannes Niederhauser

Outline

Definitions

DFA $M = (Q, \Sigma, \delta, s, F)$

- state p is inaccessible if $\widehat{\delta}(s, x) \neq p$ for all $x \in \Sigma^*$
- states p and q are indistinguishable $(p \approx q)$ if $\widehat{\delta}(p, x) \in F \iff \widehat{\delta}(q, x) \in F$ for all $x \in \Sigma^*$

Minimization Algorithm

DFA $M = (Q, \Sigma, \delta, s, F)$

① remove inaccessible states

② determine which states are indistinguishable by marking algorithm

③ collapse indistinguishable states

Marking Algorithm

given DFA $M = (Q, \Sigma, \delta, s, F)$ without inaccessible states

① tabulate all unordered pairs $\{p, q\}$ with $p, q \in Q$, initially unmarked

② mark $\{p, q\}$ if $p \in F$ and $q \notin F$ or $p \notin F$ and $q \in F$

③ repeat until no change: mark $\{p, q\}$ if $\{\delta(p, a), \delta(q, a)\}$ is marked for some $a \in \Sigma$

Lemmata

- $p \approx q \iff \{p, q\}$ is unmarked
- $\approx$ is equivalence relation on Q

Notation

$[p]_\approx = \{q \in Q \mid p \approx q\}$ denotes equivalence class of p

Definition (Collapsing Indistinguishable States)

DFA $M/{\approx}$ is defined as $(Q', \Sigma, \delta', s', F')$ with

- $Q' = \{[p]_\approx \mid p \in Q\}$
- $\delta'([p]_\approx, a) = [\delta(p,a)]_\approx$ well-defined: $p \approx q \implies \delta(p,a) \approx \delta(q,a)$
- $s' = [s]_\approx$
- $F' = \{[p]_\approx \mid p \in F\}$

Theorem

$L(M/{\approx}) = L(M)$

Question

is $M/{\approx}$ minimum-state DFA for $L(M)$?

Definitions

- first-order variables $V_1 = \{x, y, \ldots\}$ ranging over natural numbers
- second-order variables $V_2 = \{X, Y, \ldots\}$ ranging over finite sets of natural numbers
- formulas of weak monadic second-order logic (WMSO)

$$\varphi \ ::= \ \bot \mid x < y \mid X(x) \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \exists\, x.\, \varphi \mid \exists\, X.\, \varphi$$

with $x, y \in V_1$ and $X \in V_2$

Abbreviations

$$\varphi \wedge \psi \ := \ \neg(\neg\varphi \vee \neg\psi) \qquad \varphi \rightarrow \psi \ := \ \neg\varphi \vee \psi$$

$$\forall\, x.\, \varphi \ := \ \neg\exists\, x.\, \neg\varphi \qquad \forall\, X.\, \varphi \ := \ \neg\exists\, X.\, \neg\varphi$$

$$x \leqslant y \ := \ \neg(y < x) \qquad x = y \ := \ x \leqslant y \wedge y \leqslant x$$

$$\top \ := \ \neg\bot \qquad x = 0 \ := \ \neg\exists\, y.\, y < x$$

$$X(0) \ := \ \exists\, x.\, X(x) \wedge x = 0 \qquad z = y + 1 \ := \ y < z \wedge \neg\exists\, x.\, y < x \wedge x < z$$

Remarks

- $X(x)$ represents $x \in X$
- MSO is WMSO without restriction to finite sets

Definitions

- assignment α is mapping from variables $x \in V_1$ to $\mathbb{N}$ and $X \in V_2$ to finite subsets of $\mathbb{N}$
- assignment α satisfies formula φ ($\alpha \vDash \varphi$):

$$
\begin{array}{llll}
\alpha \nvDash \bot & & & \\
\alpha \vDash x < y & \iff & \alpha(x) < \alpha(y) & \\
\alpha \vDash X(x) & \iff & \alpha(x) \in \alpha(X) & \\
\alpha \vDash \neg\varphi & \iff & \alpha \nvDash \varphi & \\
\alpha \vDash \varphi_1 \vee \varphi_2 & \iff & \alpha \vDash \varphi_1 \text{ or } \alpha \vDash \varphi_2 & \\
\alpha \vDash \exists x.\, \varphi & \iff & \alpha[x \mapsto n] \vDash \varphi & \text{for some } n \in \mathbb{N} \\
\alpha \vDash \exists X.\, \varphi & \iff & \alpha[X \mapsto N] \vDash \varphi & \text{for some finite subset } N \subset \mathbb{N}
\end{array}
$$

Definitions

- formula φ is satisfiable if $\alpha \vDash \varphi$ for some assignment α
- formula φ is valid if $\alpha \vDash \varphi$ for all assignments α
- model of formula φ is assignment α such that $\alpha \vDash \varphi$
- size of model α is smallest n such that
  1. $\alpha(x) < n$ for $x \in V_1$
  2. $\alpha(X) \subseteq \{0, \dots, n-1\}$ for $X \in V_2$

Definition

given alphabet Σ and string $x = a_0 \cdots a_{n-1} \in \Sigma^*$

- second-order variables $V_2 = \{P_a \mid a \in \Sigma\}$
- $\alpha_x(P_a) = \{i < n \mid x_i = a\}$

Notation

$\underline{x}$ for α_x

Definitions

- given alphabet Σ and WMSO formula φ with free variables (exclusively) in $\{P_a \mid a \in \Sigma\}$

$$L(\varphi) = \{x \in \Sigma^* \mid \underline{x} \vDash \varphi\}$$

- set $A \subseteq \Sigma^*$ is WMSO definable if $A = L(\varphi)$ for some WMSO formula φ

Theorem

set $A \subseteq \Sigma^*$ is regular if and only if A is WMSO definable

Automata

- (deterministic, non-deterministic, alternating) finite automata
- regular expressions
- (alternating) Büchi automata

Logic

- (weak) monadic second-order logic
- Presburger arithmetic
- linear-time temporal logic

Outline

Theorem

set $A \subseteq \Sigma^*$ is regular if and only if A is WMSO definable

Theorem

set $A \subseteq \Sigma^*$ is regular if and only if A is WMSO definable

Proof ($\Longleftarrow$)

next week

Theorem

set $A \subseteq \Sigma^*$ is regular if and only if A is WMSO definable

Proof ($\Longleftarrow$)

next week

Definitions

DFA $M = (Q, \Sigma, \delta, s, F)$

- run of M on input $x = a_1 \cdots a_n \in \Sigma^*$ is sequence $q_0, \dots, q_n$ of states such that

$$s = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n$$

Theorem

set $A \subseteq \Sigma^*$ is regular if and only if A is WMSO definable

Proof ($\Longleftarrow$)

next week

Definitions

DFA $M = (Q, \Sigma, \delta, s, F)$

- run of M on input $x = a_1 \cdots a_n \in \Sigma^*$ is sequence $q_0, \dots, q_n$ of states such that

$$s = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n$$

- run $q_0, \dots, q_n$ is accepting if $q_n \in F$

Proof ($\Longrightarrow$)

- DFA $M = (Q, \Sigma, \delta, s, F)$

Proof ($\Longrightarrow$)

- DFA $M = (Q, \Sigma, \delta, s, F)$ with $Q = \{q_1, \ldots, q_m\}$

Proof ($\Longrightarrow$)

- DFA $M = (Q, \Sigma, \delta, s, F)$ with $Q = \{q_1, \dots, q_m\}$
- second-order variables $X_{q_1}, \dots, X_{q_m}$

Remarks

- $X_q = \{ i \mid \widehat{\delta}(s, a_1 \cdots a_i) = q \}$ for input $a_1 \cdots a_n \in \Sigma^*$

Remarks

- $X_q = \{\, i \mid \widehat{\delta}(s, a_1 \cdots a_i) = q \,\}$ for input $a_1 \cdots a_n \in \Sigma^*$

Example

- run $s \xrightarrow{a} p \xrightarrow{a} q \xrightarrow{b} p$

Remarks

- $X_q = \{ i \mid \widehat{\delta}(s, a_1 \cdots a_i) = q \}$ for input $a_1 \cdots a_n \in \Sigma^*$

Example

- run $s \xrightarrow{a} p \xrightarrow{a} q \xrightarrow{b} p$
- assignment

$$P_a = \{0, 1\} \qquad P_b = \{2\}$$

Remarks

- $X_q = \{ i \mid \widehat{\delta}(s, a_1 \cdots a_i) = q \}$ for input $a_1 \cdots a_n \in \Sigma^*$

Example

- run $s \xrightarrow{a} p \xrightarrow{a} q \xrightarrow{b} p$
- assignment

$$P_a = \{0, 1\} \qquad P_b = \{2\} \qquad X_s = \{0\} \qquad X_p = \{1, 3\} \qquad X_q = \{2\}$$

Proof ($\Longrightarrow$)

- DFA $M = (Q, \Sigma, \delta, s, F)$ with $Q = \{q_1, \dots, q_m\}$
- second-order variables $X_{q_1}, \dots, X_{q_m}$ to encode accepting runs of M as WMSO formula φ_M

Proof ($\Longrightarrow$)

- DFA $M = (Q, \Sigma, \delta, s, F)$ with $Q = \{q_1, \dots, q_m\}$
- second-order variables $X_{q_1}, \dots, X_{q_m}$ to encode accepting runs of M as WMSO formula φ_M:

$$\varphi_M := \exists X_{q_1}. \cdots \exists X_{q_m}. \exists \ell. \bigwedge_{a \in \Sigma} \neg P_a(\ell) \;\wedge\; \Big(\forall x. \bigwedge_{a \in \Sigma} \neg P_a(x) \to \ell \leqslant x\Big) \;\wedge\; \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

Remarks

- $X_q = \{ i \mid \widehat{\delta}(s, a_1 \cdots a_i) = q \}$ for input $a_1 \cdots a_n \in \Sigma^*$
- ℓ denotes length n of input

Example

- run $s \xrightarrow{a} p \xrightarrow{a} q \xrightarrow{b} p$
- assignment

$$P_a = \{0,1\} \qquad P_b = \{2\} \qquad X_s = \{0\} \qquad X_p = \{1,3\} \qquad X_q = \{2\}$$

Proof ($\Longrightarrow$)

- DFA $M = (Q, \Sigma, \delta, s, F)$ with $Q = \{q_1, \dots, q_m\}$
- second-order variables $X_{q_1}, \dots, X_{q_m}$ to encode accepting runs of M as WMSO formula φ_M:

$$\varphi_M := \exists X_{q_1}. \cdots \exists X_{q_m}. \exists \ell. \bigwedge_{a \in \Sigma} \neg P_a(\ell) \ \wedge \ \left(\forall x. \bigwedge_{a \in \Sigma} \neg P_a(x) \rightarrow \ell \leqslant x\right) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

$$\psi_1 := X_s(0)$$

Proof ($\Longrightarrow$)

- DFA $M = (Q, \Sigma, \delta, s, F)$ with $Q = \{q_1, \dots, q_m\}$
- second-order variables $X_{q_1}, \dots, X_{q_m}$ to encode accepting runs of M as WMSO formula φ_M:

$$\varphi_M := \exists X_{q_1}. \cdots \exists X_{q_m}. \exists \ell. \bigwedge_{a \in \Sigma} \neg P_a(\ell) \wedge \left(\forall x. \bigwedge_{a \in \Sigma} \neg P_a(x) \to \ell \leqslant x\right) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

$$\psi_1 := X_s(0)$$

$$\psi_2 := \forall x. x \leqslant \ell \to \left(\bigvee_{q \in Q} X_q(x)\right) \wedge \bigwedge_{p \neq q} \neg\left(X_p(x) \wedge X_q(x)\right)$$

Proof ($\Longrightarrow$)

- DFA $M = (Q, \Sigma, \delta, s, F)$ with $Q = \{q_1, \dots, q_m\}$
- second-order variables $X_{q_1}, \dots, X_{q_m}$ to encode accepting runs of M as WMSO formula φ_M:

$$\varphi_M := \exists X_{q_1}. \cdots \exists X_{q_m}. \exists \ell. \bigwedge_{a \in \Sigma} \neg P_a(\ell) \wedge \left(\forall x. \bigwedge_{a \in \Sigma} \neg P_a(x) \rightarrow \ell \leqslant x \right) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

$$\psi_1 := X_s(0)$$

$$\psi_2 := \forall x. x \leqslant \ell \rightarrow \left(\bigvee_{q \in Q} X_q(x) \right) \wedge \bigwedge_{p \neq q} \neg \left(X_p(x) \wedge X_q(x) \right)$$

$$\psi_3 := \forall x. x < \ell \rightarrow \bigvee_{a \in \Sigma,\, q \in Q} X_q(x) \wedge P_a(x) \wedge \exists y. y = x + 1 \wedge X_{\delta(q,a)}(y)$$

Proof ($\Longrightarrow$)

- DFA $M = (Q, \Sigma, \delta, s, F)$ with $Q = \{q_1, \ldots, q_m\}$
- second-order variables $X_{q_1}, \ldots, X_{q_m}$ to encode accepting runs of M as WMSO formula φ_M:

$$\varphi_M := \exists X_{q_1}. \cdots \exists X_{q_m}. \exists \ell. \bigwedge_{a \in \Sigma} \neg P_a(\ell) \;\wedge\; \Big(\forall x. \bigwedge_{a \in \Sigma} \neg P_a(x) \to \ell \leqslant x\Big) \;\wedge\; \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

$$\psi_1 := X_s(0)$$

$$\psi_2 := \forall x. x \leqslant \ell \to \Big(\bigvee_{q \in Q} X_q(x)\Big) \;\wedge\; \bigwedge_{p \neq q} \neg \big(X_p(x) \wedge X_q(x)\big)$$

$$\psi_3 := \forall x. x < \ell \to \bigvee_{a \in \Sigma, q \in Q} X_q(x) \;\wedge\; P_a(x) \;\wedge\; \exists y. y = x + 1 \;\wedge\; X_{\delta(q,a)}(y)$$

$$\psi_4 := \bigvee_{q \in F} X_q(\ell)$$

Remarks

- $X_q = \{ i \mid \widehat{\delta}(s, a_1 \cdots a_i) = q \}$ for input $a_1 \cdots a_n \in \Sigma^*$
- ℓ denotes length n of input

Example

- run $s \xrightarrow{a} p \xrightarrow{a} q \xrightarrow{b} p$
- assignment

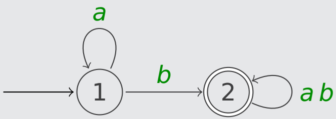
$$P_a = \{0,1\} \qquad P_b = \{2\} \qquad X_s = \{0\} \qquad X_p = \{1,3\} \qquad X_q = \{2\}$$

Proof ($\Longrightarrow$, cont'd)

- $L(\varphi_M) = L(M)$

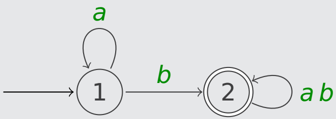
Example

- DFA M



Example

- DFA M

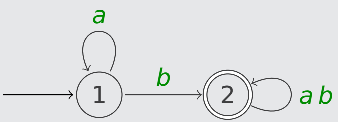


- WMSO formula φ_M

$$\exists X_1 . \exists X_2 . \exists \ell . \neg P_a(\ell) \wedge \neg P_b(\ell) \wedge \big(\forall x . \neg P_a(x) \wedge \neg P_b(x) \to \ell \leqslant x\big) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

Example

- DFA M



- WMSO formula φ_M

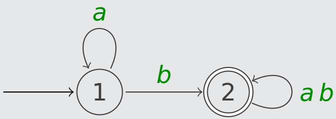
$$\exists X_1.\,\exists X_2.\,\exists \ell.\,\neg P_a(\ell) \wedge \neg P_b(\ell) \wedge \big(\forall x.\,\neg P_a(x) \wedge \neg P_b(x) \to \ell \leqslant x\big) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

with

$$\psi_1 = X_1(0)$$

Example

- DFA M



- WMSO formula φ_M

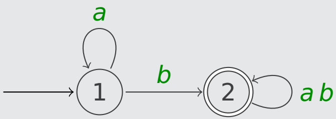
$$\exists X_1.\exists X_2.\exists \ell.\neg P_a(\ell) \wedge \neg P_b(\ell) \wedge \left(\forall x.\neg P_a(x) \wedge \neg P_b(x) \rightarrow \ell \leqslant x\right) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

with

$$\psi_1 = X_1(0) \qquad \psi_2 = \forall x.x \leqslant \ell \rightarrow \left(X_1(x) \vee X_2(x)\right) \wedge \neg\left(X_1(x) \wedge X_2(x)\right)$$

Example

- DFA M



- WMSO formula φ_M

$$\exists X_1 . \exists X_2 . \exists \ell . \neg P_a(\ell) \wedge \neg P_b(\ell) \wedge \big(\forall x . \neg P_a(x) \wedge \neg P_b(x) \rightarrow \ell \leqslant x\big) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

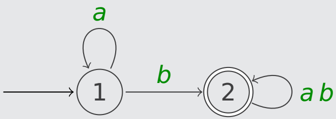
with

$$\psi_1 = X_1(0) \qquad \psi_2 = \forall x . x \leqslant \ell \rightarrow \big(X_1(x) \vee X_2(x)\big) \wedge \neg\big(X_1(x) \wedge X_2(x)\big)$$

$$\psi_3 = \forall x . x < \ell \rightarrow \big(X_1(x) \wedge P_a(x) \wedge \exists y . y = x + 1 \wedge X_1(y)\big)$$

Example

- DFA M



- WMSO formula φ_M

$$\exists X_1.\exists X_2.\exists \ell.\neg P_a(\ell) \wedge \neg P_b(\ell) \wedge (\forall x.\neg P_a(x) \wedge \neg P_b(x) \rightarrow \ell \leqslant x) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

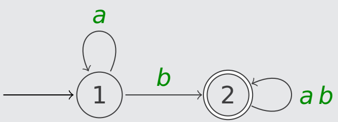
with

$$\psi_1 = X_1(0) \qquad \psi_2 = \forall x.x \leqslant \ell \rightarrow (X_1(x) \vee X_2(x)) \wedge \neg(X_1(x) \wedge X_2(x))$$

$$\psi_3 = \forall x.x < \ell \rightarrow (X_1(x) \wedge P_a(x) \wedge \exists y.y = x+1 \wedge X_1(y)) \vee (X_1(x) \wedge P_b(x) \wedge \exists y.y = x+1 \wedge X_2(y))$$

Example

- DFA M



- WMSO formula φ_M

$$\exists X_1.\exists X_2.\exists \ell.\neg P_a(\ell) \wedge \neg P_b(\ell) \wedge (\forall x.\neg P_a(x) \wedge \neg P_b(x) \to \ell \leqslant x) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

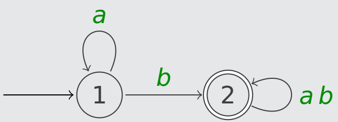
with

$$\psi_1 = X_1(0) \qquad \psi_2 = \forall x.x \leqslant \ell \to (X_1(x) \vee X_2(x)) \wedge \neg(X_1(x) \wedge X_2(x))$$

$$\begin{aligned}\psi_3 = \forall x.x < \ell \to\ & (X_1(x) \wedge P_a(x) \wedge \exists y.y = x+1 \wedge X_1(y)) \vee \\ & (X_1(x) \wedge P_b(x) \wedge \exists y.y = x+1 \wedge X_2(y)) \vee \\ & (X_2(x) \wedge P_a(x) \wedge \exists y.y = x+1 \wedge X_2(y)) \vee \\ & (X_2(x) \wedge P_b(x) \wedge \exists y.y = x+1 \wedge X_2(y))\end{aligned}$$

Example

- DFA M



- WMSO formula φ_M

$$\exists\, X_1.\, \exists\, X_2.\, \exists\, \ell.\, \neg P_a(\ell) \wedge \neg P_b(\ell) \wedge \big(\forall\, x.\, \neg P_a(x) \wedge \neg P_b(x) \to \ell \leqslant x\big) \wedge \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$$

with

$$\psi_1 = X_1(0) \qquad \psi_2 = \forall\, x.\, x \leqslant \ell \to \big(X_1(x) \vee X_2(x)\big) \wedge \neg\big(X_1(x) \wedge X_2(x)\big)$$

$$\begin{aligned}\psi_3 = \forall\, x.\, x < \ell \to\ & \big(X_1(x) \wedge P_a(x) \wedge \exists\, y.\, y = x+1 \wedge X_1(y)\big) \vee \\ & \big(X_1(x) \wedge P_b(x) \wedge \exists\, y.\, y = x+1 \wedge X_2(y)\big) \vee \\ & \big(X_2(x) \wedge P_a(x) \wedge \exists\, y.\, y = x+1 \wedge X_2(y)\big) \vee \\ & \big(X_2(x) \wedge P_b(x) \wedge \exists\, y.\, y = x+1 \wedge X_2(y)\big)\end{aligned}$$

$$\psi_4 = X_2(\ell)$$

Outline

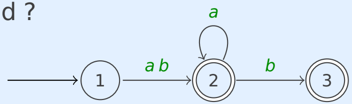
Question

Consider the language encoded by the following WMSO formula over $\Sigma = \{a, b\}$:

$$\varphi = \exists\, \ell.\, \neg P_a(\ell) \wedge \neg P_b(\ell) \wedge \big(\forall\, x.\, \neg P_a(x) \wedge \neg P_b(x) \to \ell \leqslant x\big) \wedge \big(\forall\, x.\, P_b(x) \to x = 0 \vee \ell = x + 1\big)$$

Which of the following statements hold ?

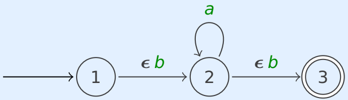
A $L(\varphi) = L(N)$ for the NFA N:



B $L(\varphi) = \Sigma^*$

C $L(\varphi) = L(a^* + ba^* + a^*b + ba^*b)$

D $L(\varphi) = L(N')$ for the NFA$_\epsilon$ N':



Outline

Definition

equivalence relation $\equiv_M$ on Σ^* for DFA $M = (Q, \Sigma, \delta, s, F)$ is defined as follows:

$$x \equiv_M y \quad \Longleftrightarrow \quad \widehat{\delta}(s, x) = \widehat{\delta}(s, y)$$

Definition

equivalence relation $\equiv_M$ on Σ^* for DFA $M = (Q, \Sigma, \delta, s, F)$ is defined as follows:

$$x \equiv_M y \iff \widehat{\delta}(s, x) = \widehat{\delta}(s, y)$$

Lemmata

- $\equiv_M$ is right congruent: for all $x, y \in \Sigma^*$ $\quad x \equiv_M y \implies$ for all $a \in \Sigma \quad xa \equiv_M ya$

Definition

equivalence relation $\equiv_M$ on Σ^* for DFA $M = (Q, \Sigma, \delta, s, F)$ is defined as follows:

$$x \equiv_M y \quad \Longleftrightarrow \quad \widehat{\delta}(s, x) = \widehat{\delta}(s, y)$$

Lemmata

- $\equiv_M$ is right congruent: for all $x, y \in \Sigma^*$ $\quad x \equiv_M y \quad \Longrightarrow \quad$ for all $a \in \Sigma$ $\quad xa \equiv_M ya$
- $\equiv_M$ refines $L(M)$: for all $x, y \in \Sigma^*$ $\quad x \equiv_M y \quad \Longrightarrow \quad$ either $x, y \in L(M)$ or $x, y \notin L(M)$

Definition

equivalence relation $\equiv_M$ on Σ^* for DFA $M = (Q, \Sigma, \delta, s, F)$ is defined as follows:

$$x \equiv_M y \iff \widehat{\delta}(s,x) = \widehat{\delta}(s,y)$$

Lemmata

- $\equiv_M$ is right congruent: for all $x, y \in \Sigma^*$ $\quad x \equiv_M y \implies$ for all $a \in \Sigma$ $\quad xa \equiv_M ya$
- $\equiv_M$ refines $L(M)$: for all $x, y \in \Sigma^*$ $\quad x \equiv_M y \implies$ either $x, y \in L(M)$ or $x, y \notin L(M)$
- $\equiv_M$ is of finite index: $\equiv_M$ has finitely many equivalence classes

Definition

equivalence relation $\equiv_M$ on Σ^* for DFA $M = (Q, \Sigma, \delta, s, F)$ is defined as follows:

$$x \equiv_M y \iff \widehat{\delta}(s, x) = \widehat{\delta}(s, y)$$

Lemmata

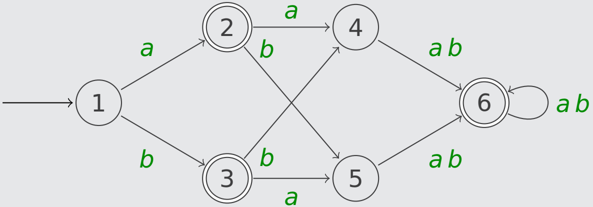
- $\equiv_M$ is right congruent: for all $x, y \in \Sigma^*$ $x \equiv_M y \implies$ for all $a \in \Sigma$ $xa \equiv_M ya$
- $\equiv_M$ refines $L(M)$: for all $x, y \in \Sigma^*$ $x \equiv_M y \implies$ either $x, y \in L(M)$ or $x, y \notin L(M)$
- $\equiv_M$ is of finite index: $\equiv_M$ has finitely many equivalence classes

Definition

Myhill–Nerode relation for $L \subseteq \Sigma^*$ is right congruent equivalence relation of finite index on Σ^* that refines L

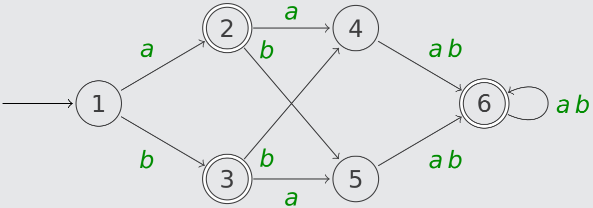
Example

M

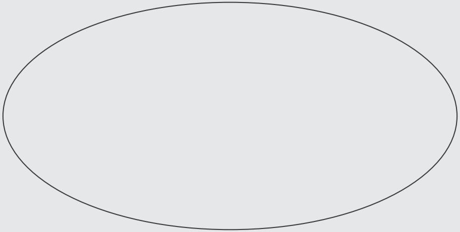


Example

M

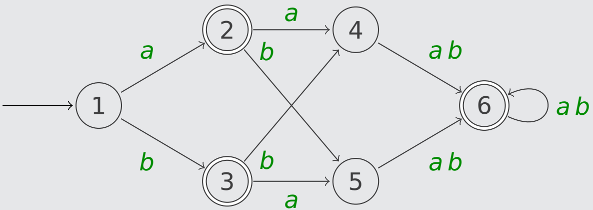


$\equiv_M$

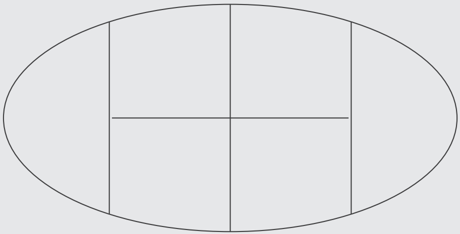


Example

M

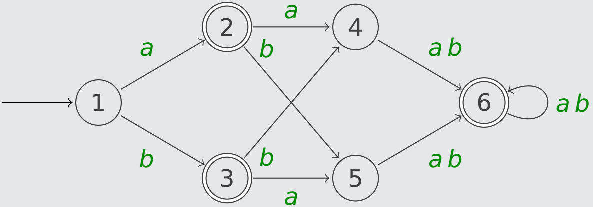


$\equiv_M$

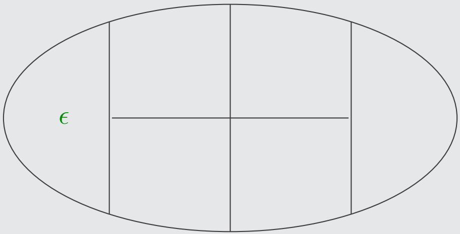


Example

M

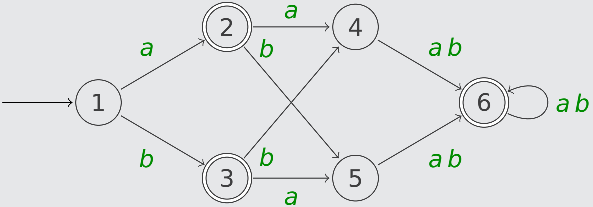


$\equiv_M$

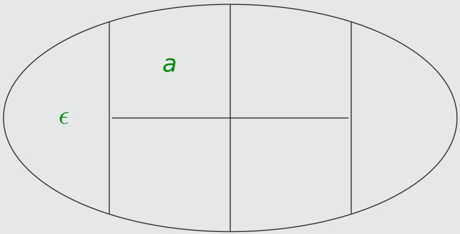


Example

M

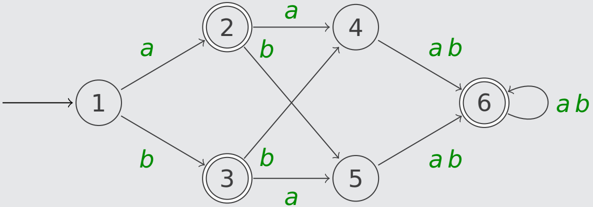


$\equiv_M$

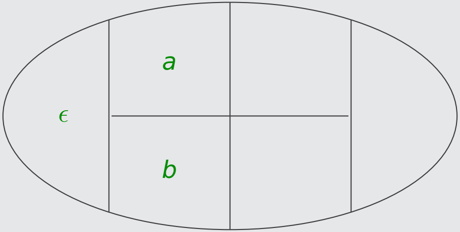


Example

M

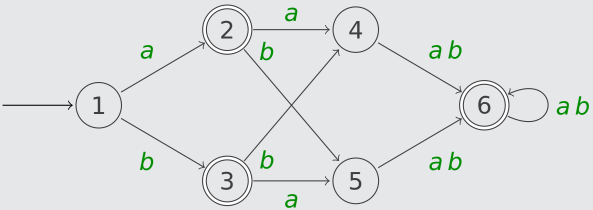


$\equiv_M$

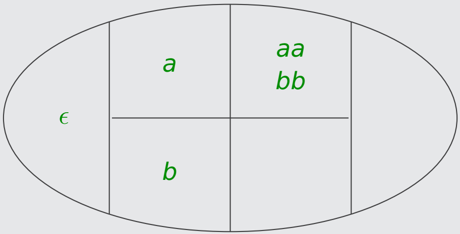


Example

M

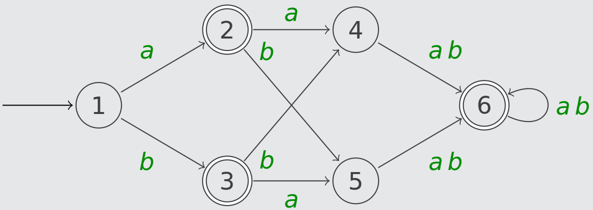


$\equiv_M$

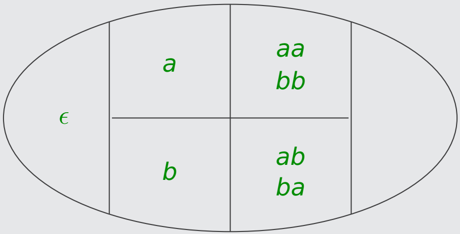


Example

M

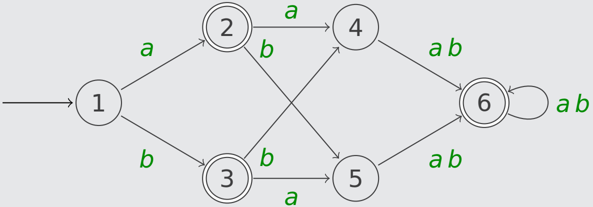


$\equiv_M$

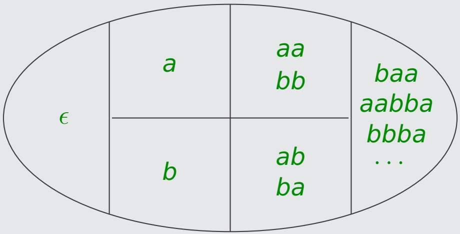


Example

M

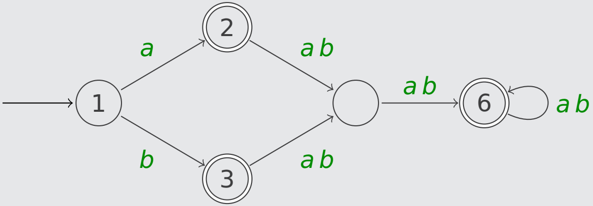


$\equiv_M$

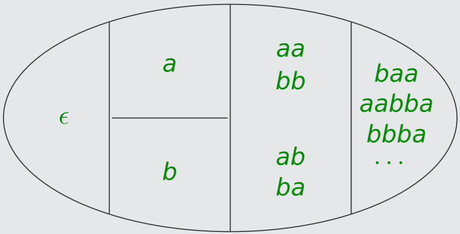


Example

M

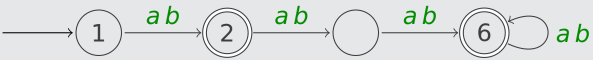


$\equiv_M$

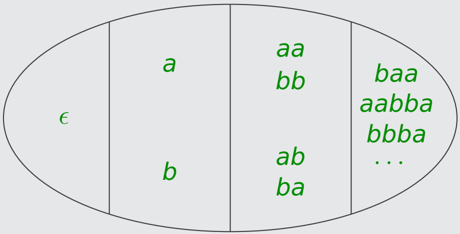


Example

M



$\equiv_M$



Definition

given Myhill–Nerode relation $\equiv$ for set $L \subseteq \Sigma^*$ DFA $M_{\equiv}$ is defined as $(Q, \Sigma, \delta, s, F)$ with

- $Q = \{ [x]_{\equiv} \mid x \in \Sigma^* \}$

Definition

given Myhill–Nerode relation $\equiv$ for set $L \subseteq \Sigma^*$ DFA $M_\equiv$ is defined as $(Q, \Sigma, \delta, s, F)$ with

- $Q = \{ [x]_\equiv \mid x \in \Sigma^* \}$
- $\delta([x]_\equiv, a) = [xa]_\equiv$

Definition

given Myhill–Nerode relation $\equiv$ for set $L \subseteq \Sigma^*$ DFA $M_\equiv$ is defined as $(Q, \Sigma, \delta, s, F)$ with

- $Q = \{ [x]_\equiv \mid x \in \Sigma^* \}$
- $\delta([x]_\equiv, a) = [xa]_\equiv$
- $s = [\epsilon]_\equiv$

Definition

given Myhill–Nerode relation $\equiv$ for set $L \subseteq \Sigma^*$ DFA $M_\equiv$ is defined as $(Q, \Sigma, \delta, s, F)$ with

- $Q = \{[x]_\equiv \mid x \in \Sigma^*\}$
- $\delta([x]_\equiv, a) = [xa]_\equiv$
- $s = [\epsilon]_\equiv$
- $F = \{[x]_\equiv \mid x \in L\}$

Definition

given Myhill–Nerode relation $\equiv$ for set $L \subseteq \Sigma^*$ DFA $M_\equiv$ is defined as $(Q, \Sigma, \delta, s, F)$ with

- $Q = \{[x]_\equiv \mid x \in \Sigma^*\}$
- $\delta([x]_\equiv, a) = [xa]_\equiv$ well-defined: $x \equiv y \implies xa \equiv ya$
- $s = [\epsilon]_\equiv$
- $F = \{[x]_\equiv \mid x \in L\}$

Definition

given Myhill–Nerode relation $\equiv$ for set $L \subseteq \Sigma^*$ DFA $M_\equiv$ is defined as $(Q, \Sigma, \delta, s, F)$ with

- $Q = \{[x]_\equiv \mid x \in \Sigma^*\}$
- $\delta([x]_\equiv, a) = [xa]_\equiv$ well-defined: $x \equiv y \implies xa \equiv ya$
- $s = [\epsilon]_\equiv$
- $F = \{[x]_\equiv \mid x \in L\}$

Lemma

1. $\widehat{\delta}([x]_\equiv, y) = [xy]_\equiv$ for all $y \in \Sigma^*$

for all $x \in \Sigma^*$

Definition

given Myhill–Nerode relation $\equiv$ for set $L \subseteq \Sigma^*$ DFA $M_\equiv$ is defined as $(Q, \Sigma, \delta, s, F)$ with

- $Q = \{ [x]_\equiv \mid x \in \Sigma^* \}$
- $\delta([x]_\equiv, a) = [xa]_\equiv$ well-defined: $x \equiv y \implies xa \equiv ya$
- $s = [\epsilon]_\equiv$
- $F = \{ [x]_\equiv \mid x \in L \}$

Lemma

1. $\widehat{\delta}([x]_\equiv, y) = [xy]_\equiv$ for all $y \in \Sigma^*$
2. $x \in L \iff [x]_\equiv \in F$

for all $x \in \Sigma^*$

Theorem

$L(M_{\equiv}) = L$

Theorem

$L(M_{\equiv}) = L$

Proof

$$x \in L(M_{\equiv}) \iff \widehat{\delta}([\epsilon]_{\equiv}, x) \in F$$

Theorem

$L(M_{\equiv}) = L$

Proof

$$x \in L(M_{\equiv}) \iff \widehat{\delta}([\epsilon]_{\equiv}, x) \in F \iff [x]_{\equiv} \in F$$

Theorem

$L(M_\equiv) = L$

Proof

$$x \in L(M_\equiv) \iff \widehat{\delta}([\epsilon]_\equiv, x) \in F \iff [x]_\equiv \in F \iff x \in L$$

Theorem

$L(M_{\equiv}) = L$

Proof

$$x \in L(M_{\equiv}) \iff \widehat{\delta}([\epsilon]_{\equiv}, x) \in F \iff [x]_{\equiv} \in F \iff x \in L$$

Corollary

if L admits Myhill–Nerode relation then L is regular

Theorem

two mappings (for $L \subseteq \Sigma^*$)

- $D \mapsto \equiv_D$ from DFAs for L to Myhill–Nerode relations for L
- $\approx \mapsto M_\approx$ from Myhill–Nerode relations for L to DFAs for L

are each others inverse (up to isomorphism of automata)

Theorem

two mappings (for $L \subseteq \Sigma^*$)

- $D \mapsto \equiv_D$ from DFAs for L to Myhill–Nerode relations for L
- $\approx \mapsto M_\approx$ from Myhill–Nerode relations for L to DFAs for L

are each others inverse (up to isomorphism of automata):

- $\equiv_D$ for every DFA D without inaccessible states

Theorem

two mappings (for $L \subseteq \Sigma^*$)

- $D \mapsto \equiv_D$ from DFAs for L to Myhill–Nerode relations for L
- $\approx \mapsto M_\approx$ from Myhill–Nerode relations for L to DFAs for L

are each others inverse (up to isomorphism of automata):

- $M_{(\equiv_D)}$ for every DFA D without inaccessible states

Theorem

two mappings (for $L \subseteq \Sigma^*$)

- $D \mapsto \equiv_D$ from DFAs for L to Myhill–Nerode relations for L
- $\approx \mapsto M_\approx$ from Myhill–Nerode relations for L to DFAs for L

are each others inverse (up to isomorphism of automata):

- $M_{(\equiv_D)} \simeq D$ for every DFA D without inaccessible states

Theorem

two mappings (for $L \subseteq \Sigma^*$)

- $D \mapsto \equiv_D$ from DFAs for L to Myhill–Nerode relations for L
- $\approx \mapsto M_\approx$ from Myhill–Nerode relations for L to DFAs for L

are each others inverse (up to isomorphism of automata):

- $M_{(\equiv_D)} \simeq D$ for every DFA D without inaccessible states
- $M_\approx$ for every Myhill–Nerode relation $\approx$

Theorem

two mappings (for $L \subseteq \Sigma^*$)

- $D \mapsto \equiv_D$ from DFAs for L to Myhill–Nerode relations for L
- $\approx \mapsto M_{\approx}$ from Myhill–Nerode relations for L to DFAs for L

are each others inverse (up to isomorphism of automata):

- $M_{(\equiv_D)} \simeq D$ for every DFA D without inaccessible states
- $\equiv_{(M_{\approx})}$ for every Myhill–Nerode relation $\approx$

Theorem

two mappings (for $L \subseteq \Sigma^*$)

- $D \mapsto \equiv_D$ from DFAs for L to Myhill–Nerode relations for L
- $\approx \mapsto M_\approx$ from Myhill–Nerode relations for L to DFAs for L

are each others inverse (up to isomorphism of automata):

- $M_{(\equiv_D)} \simeq D$ for every DFA D without inaccessible states
- $\equiv_{(M_\approx)} = \approx$ for every Myhill–Nerode relation $\approx$

for regular $L \subseteq \Sigma^*$

DFAs for L

Myhill–Nerode relations for L

for regular $L \subseteq \Sigma^*$

DFAs with inaccessible states

DFAs for L

Myhill–Nerode relations for L

for regular $L \subseteq \Sigma^*$

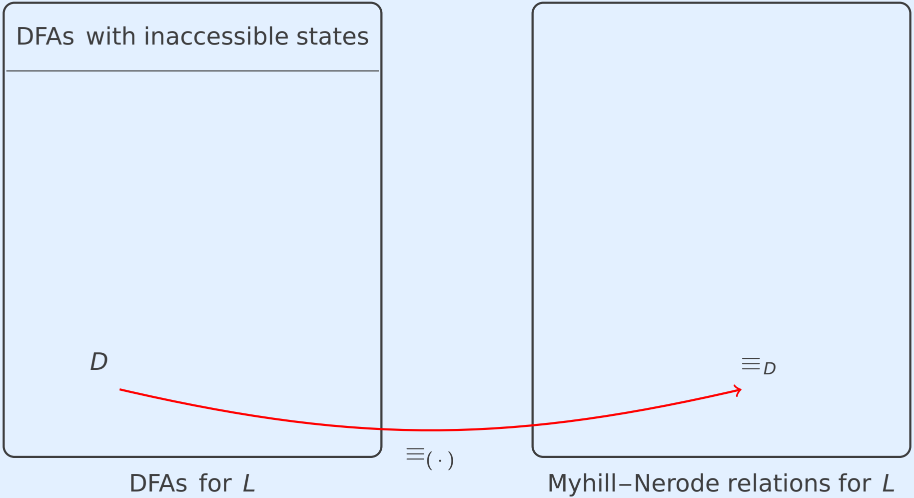
DFAs with inaccessible states

D

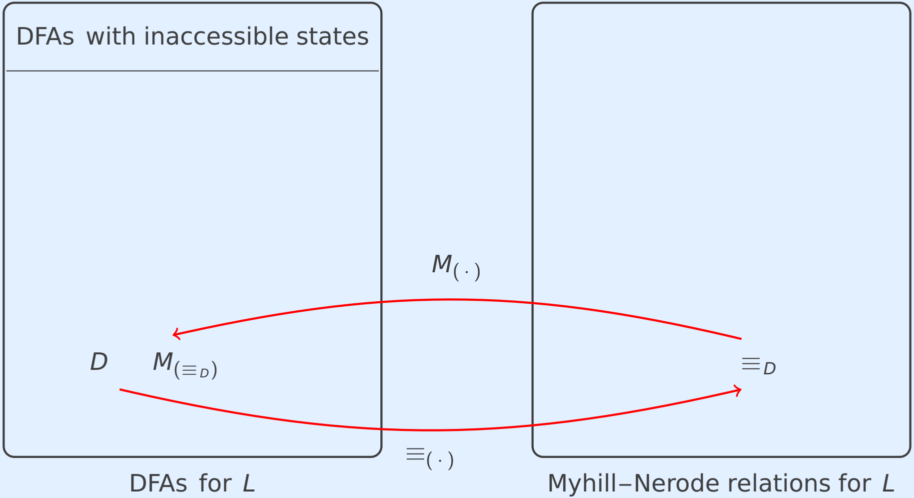
DFAs for L

Myhill–Nerode relations for L

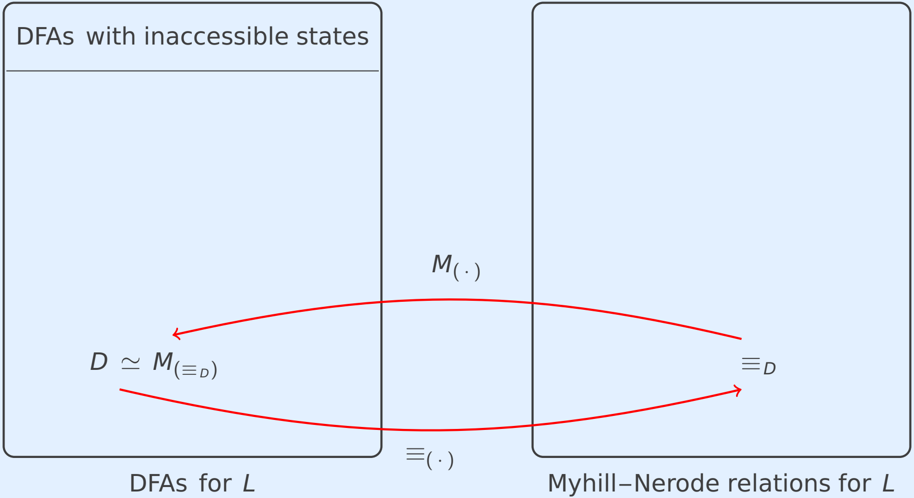
for regular $L \subseteq \Sigma^*$



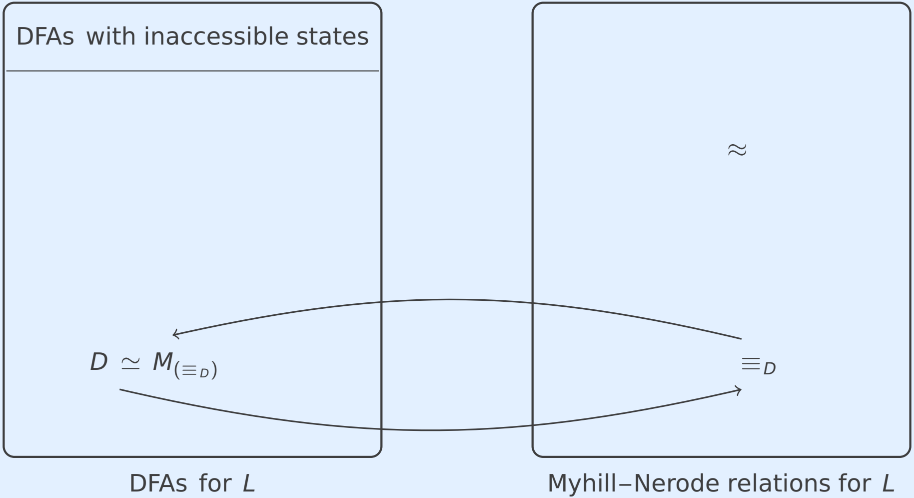
for regular $L \subseteq \Sigma^*$



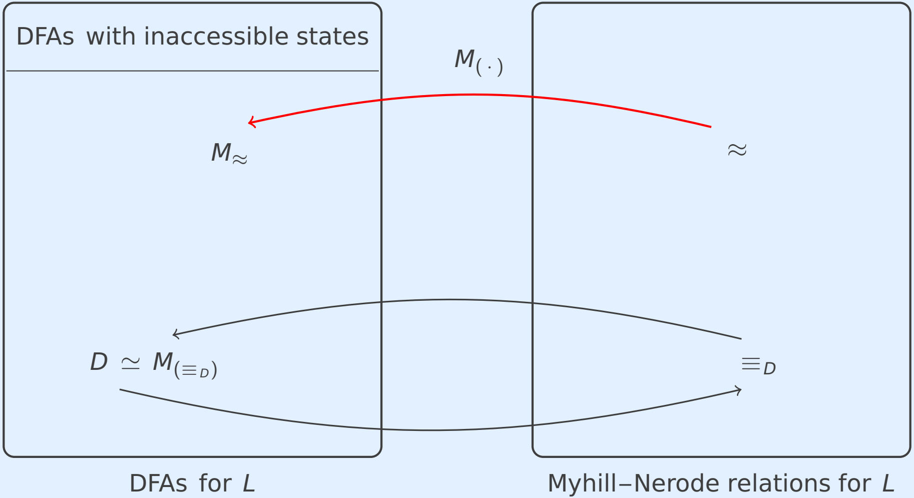
for regular $L \subseteq \Sigma^*$



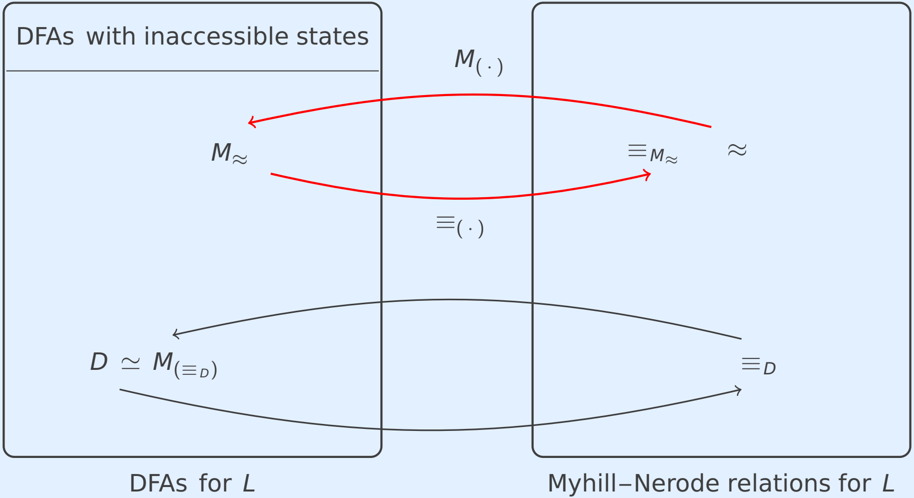
for regular $L \subseteq \Sigma^*$



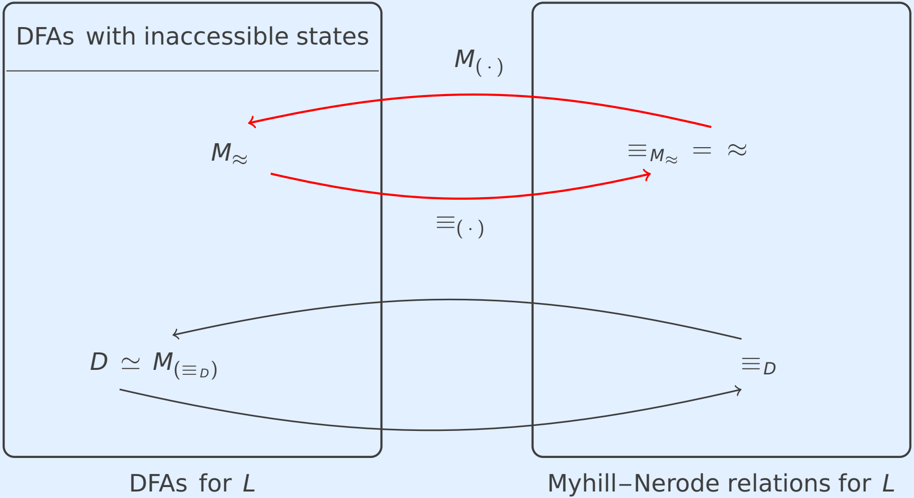
for regular $L \subseteq \Sigma^*$



for regular $L \subseteq \Sigma^*$



for regular $L \subseteq \Sigma^*$



Definition

for any set $L \subseteq \Sigma^*$ equivalence relation $\equiv_L$ on Σ^* is defined as follows:

$$x \equiv_L y \iff \text{for all } z \in \Sigma^* \ \left(xz \in L \iff yz \in L \right)$$

Definition

for any set $L \subseteq \Sigma^*$ equivalence relation $\equiv_L$ on Σ^* is defined as follows:

$$x \equiv_L y \iff \text{for all } z \in \Sigma^* \ \left(xz \in L \iff yz \in L\right)$$

Lemma

for any set $L \subseteq \Sigma^*$ $\equiv_L$ is coarsest right congruent refinement of L

Definition

for any set $L \subseteq \Sigma^*$ equivalence relation $\equiv_L$ on Σ^* is defined as follows:

$$x \equiv_L y \iff \text{for all } z \in \Sigma^* \ \left(xz \in L \iff yz \in L\right)$$

Lemma

for any set $L \subseteq \Sigma^*$ $\equiv_L$ is coarsest right congruent refinement of L:

if $\sim$ is right congruent equivalence relation refining L then

$$\text{for all } x, y \in \Sigma^* \quad x \sim y \implies x \equiv_L y$$

Definition

for any set $L \subseteq \Sigma^*$ equivalence relation $\equiv_L$ on Σ^* is defined as follows:

$$x \equiv_L y \iff \text{for all } z \in \Sigma^* \; \big(xz \in L \iff yz \in L\big)$$

Lemma

for any set $L \subseteq \Sigma^*$ $\equiv_L$ is coarsest right congruent refinement of L:

if $\sim$ is right congruent equivalence relation refining L then

$$\text{for all } x, y \in \Sigma^* \quad x \sim y \implies x \equiv_L y$$

$\equiv_L$ has fewest equivalence classes

Theorem

following statements are equivalent for any set $L \subseteq \Sigma^*$:

- L is regular
- L admits Myhill–Nerode relation

Theorem

following statements are equivalent for any set $L \subseteq \Sigma^*$:

- L is regular
- L admits Myhill–Nerode relation
- $\equiv_L$ is of finite index

Theorem

following statements are equivalent for any set $L \subseteq \Sigma^*$:

- L is regular
- L admits Myhill–Nerode relation
- $\equiv_L$ is of finite index

Corollary

for every regular set L $\quad M_{(\equiv_L)}$ is minimum-state DFA for L

Theorem

following statements are equivalent for any set $L \subseteq \Sigma^*$:

- L is regular
- L admits Myhill–Nerode relation
- $\equiv_L$ is of finite index

Corollary

for every regular set L $\quad M_{(\equiv_L)}$ is minimum-state DFA for L

Theorem

for every DFA M $\quad M/{\approx} \simeq M_{(\equiv_L)}$

Theorem

following statements are equivalent for any set $L \subseteq \Sigma^*$:

- L is regular
- L admits Myhill–Nerode relation
- $\equiv_L$ is of finite index

Corollary

for every regular set L $\quad M_{(\equiv_L)}$ is minimum-state DFA for L

Theorem

for every DFA M $\quad M/{\approx} \simeq M_{(\equiv_L)}$

Theorem

following statements are equivalent for any set $L \subseteq \Sigma^*$:

- L is regular
- L admits Myhill–Nerode relation
- $\equiv_L$ is of finite index

Corollary

for every regular set L $\quad M_{(\equiv_L)}$ is minimum-state DFA for L

Theorem

for every DFA M $\quad M/{\approx} \simeq M_{(\equiv_L)}$

Examples

1. $A = \{ a^n b^n \mid n \geqslant 0 \}$ is not regular

Examples

1. $A = \{a^n b^n \mid n \geqslant 0\}$ is not regular

   because $\equiv_A$ has infinitely many equivalence classes

Examples

1. $A = \{a^n b^n \mid n \geqslant 0\}$ is not regular

   because $\equiv_A$ has infinitely many equivalence classes:

$$i \neq j \implies a^i \not\equiv_A a^j$$

Examples

1. $A = \{a^n b^n \mid n \geqslant 0\}$ is not regular

   because $\equiv_A$ has infinitely many equivalence classes:

$$i \neq j \quad \Longrightarrow \quad a^i \not\equiv_A a^j \quad \left(a^i b^i \in A \text{ and } a^j b^i \notin A\right)$$

Examples

1. $A = \{a^n b^n \mid n \geqslant 0\}$ is not regular

   because $\equiv_A$ has infinitely many equivalence classes:

$$i \neq j \quad \Longrightarrow \quad a^i \not\equiv_A a^j \quad \left(a^i b^i \in A \text{ and } a^j b^i \notin A\right)$$

2. $B = \{a^{2^n} \mid n \geqslant 0\}$ is not regular

   because $\equiv_B$ has infinitely many equivalence classes

Examples

1. $A = \{a^n b^n \mid n \geqslant 0\}$ is not regular

   because $\equiv_A$ has infinitely many equivalence classes:

$$i \neq j \quad \Longrightarrow \quad a^i \not\equiv_A a^j \quad \big(a^i b^i \in A \text{ and } a^j b^i \notin A\big)$$

2. $B = \{a^{2^n} \mid n \geqslant 0\}$ is not regular

   because $\equiv_B$ has infinitely many equivalence classes:

$$i < j \quad \Longrightarrow \quad a^{2^i} \not\equiv_B a^{2^j}$$

Examples

1. $A = \{a^n b^n \mid n \geqslant 0\}$ is not regular

   because $\equiv_A$ has infinitely many equivalence classes:

$$i \neq j \quad \Longrightarrow \quad a^i \not\equiv_A a^j \quad \left(a^i b^i \in A \text{ and } a^j b^i \notin A\right)$$

2. $B = \{a^{2^n} \mid n \geqslant 0\}$ is not regular

   because $\equiv_B$ has infinitely many equivalence classes:

$$i < j \quad \Longrightarrow \quad a^{2^i} \not\equiv_B a^{2^j} \quad \left(a^{2^i} a^{2^i} = a^{2^{i+1}} \in B \text{ and } a^{2^j} a^{2^i} \notin B\right)$$

Examples

1. $A = \{a^n b^n \mid n \geqslant 0\}$ is not regular

   because $\equiv_A$ has infinitely many equivalence classes:

$$i \neq j \implies a^i \not\equiv_A a^j \quad \left(a^i b^i \in A \text{ and } a^j b^i \notin A\right)$$

2. $B = \{a^{2^n} \mid n \geqslant 0\}$ is not regular

   because $\equiv_B$ has infinitely many equivalence classes:

$$i < j \implies a^{2^i} \not\equiv_B a^{2^j} \quad \left(a^{2^i} a^{2^i} = a^{2^{i+1}} \in B \text{ and } a^{2^j} a^{2^i} \notin B\right)$$

3. $C = \{a^{n!} \mid n \geqslant 0\}$ is not regular

   because $\equiv_C$ has infinitely many equivalence classes

Examples

1. $A = \{a^n b^n \mid n \geqslant 0\}$ is not regular

   because $\equiv_A$ has infinitely many equivalence classes:

$$i \neq j \quad \Longrightarrow \quad a^i \not\equiv_A a^j \quad \left(a^i b^i \in A \text{ and } a^j b^i \notin A\right)$$

2. $B = \{a^{2^n} \mid n \geqslant 0\}$ is not regular

   because $\equiv_B$ has infinitely many equivalence classes:

$$i < j \quad \Longrightarrow \quad a^{2^i} \not\equiv_B a^{2^j} \quad \left(a^{2^i} a^{2^i} = a^{2^{i+1}} \in B \text{ and } a^{2^j} a^{2^i} \notin B\right)$$

3. $C = \{a^{n!} \mid n \geqslant 0\}$ is not regular

   because $\equiv_C$ has infinitely many equivalence classes:

$$i < j \quad \Longrightarrow \quad a^{i!} \not\equiv_C a^{j!}$$

Examples

1. $A = \{a^n b^n \mid n \geqslant 0\}$ is not regular

   because $\equiv_A$ has infinitely many equivalence classes:

$$i \neq j \quad \Longrightarrow \quad a^i \not\equiv_A a^j \quad (a^i b^i \in A \text{ and } a^j b^i \notin A)$$

2. $B = \{a^{2^n} \mid n \geqslant 0\}$ is not regular

   because $\equiv_B$ has infinitely many equivalence classes:

$$i < j \quad \Longrightarrow \quad a^{2^i} \not\equiv_B a^{2^j} \quad (a^{2^i} a^{2^i} = a^{2^{i+1}} \in B \text{ and } a^{2^j} a^{2^i} \notin B)$$

3. $C = \{a^{n!} \mid n \geqslant 0\}$ is not regular

   because $\equiv_C$ has infinitely many equivalence classes:

$$i < j \quad \Longrightarrow \quad a^{i!} \not\equiv_C a^{j!} \quad (a^{i!} a^{i!i} = a^{(i+1)!} \in C \text{ and } a^{j!} a^{i!i} \notin C)$$

Example

4. $D = \{a^p \mid p \text{ is prime}\}$ is not regular

   because $\equiv_D$ has infinitely many equivalence classes:

$$i < j \text{ and } i, j \text{ are primes} \implies a^i \not\equiv_D a^j$$

Example

4. $D = \{a^p \mid p \text{ is prime}\}$ is not regular

   because $\equiv_D$ has infinitely many equivalence classes:

$$i < j \text{ and } i, j \text{ are primes} \quad\Longrightarrow\quad a^i \not\equiv_D a^j$$

- suppose $a^i \equiv_D a^j$ and let $k = j - i$

Example

④ $D = \{a^p \mid p \text{ is prime}\}$ is not regular

because $\equiv_D$ has infinitely many equivalence classes:

$$i < j \text{ and } i, j \text{ are primes} \quad \Longrightarrow \quad a^i \not\equiv_D a^j$$

- suppose $a^i \equiv_D a^j$ and let $k = j - i$
- $a^i \equiv_D a^j = a^i a^k \equiv_D a^j a^k$

Example

④ $D = \{a^p \mid p \text{ is prime}\}$ is not regular

because $\equiv_D$ has infinitely many equivalence classes:

$$i < j \text{ and } i, j \text{ are primes} \implies a^i \not\equiv_D a^j$$

- suppose $a^i \equiv_D a^j$ and let $k = j - i$
- $a^i \equiv_D a^j = a^i a^k \equiv_D a^j a^k \equiv_D a^j a^k a^k = a^j a^{2k}$

Example

4. $D = \{a^p \mid p \text{ is prime}\}$ is not regular

because $\equiv_D$ has infinitely many equivalence classes:

$$i < j \text{ and } i, j \text{ are primes} \quad \Longrightarrow \quad a^i \not\equiv_D a^j$$

- suppose $a^i \equiv_D a^j$ and let $k = j - i$
- $a^i \equiv_D a^j = a^i a^k \equiv_D a^j a^k \equiv_D a^j a^k a^k = a^j a^{2k} \equiv_D \cdots \equiv_D a^j a^{jk} = a^{j(k+1)}$

Example

4. $D = \{a^p \mid p \text{ is prime}\}$ is not regular

   because $\equiv_D$ has infinitely many equivalence classes:

$$i < j \text{ and } i, j \text{ are primes} \implies a^i \not\equiv_D a^j$$

- suppose $a^i \equiv_D a^j$ and let $k = j - i$
- $a^i \equiv_D a^j = a^i a^k \equiv_D a^j a^k \equiv_D a^j a^k a^k = a^j a^{2k} \equiv_D \cdots \equiv_D a^j a^{jk} = a^{j(k+1)}$
- $a^i \in D$ and $a^{j(k+1)} \notin D$

Example

4. $D = \{a^p \mid p \text{ is prime}\}$ is not regular

because $\equiv_D$ has infinitely many equivalence classes:

$$i < j \text{ and } i,j \text{ are primes} \quad \Longrightarrow \quad a^i \not\equiv_D a^j$$

- suppose $a^i \equiv_D a^j$ and let $k = j - i$
- $a^i \equiv_D a^j = a^i a^k \equiv_D a^j a^k \equiv_D a^j a^k a^k = a^j a^{2k} \equiv_D \cdots \equiv_D a^j a^{jk} = a^{j(k+1)}$
- $a^i \in D$ and $a^{j(k+1)} \notin D$
- $\equiv_D$ does not refine D ↯

Outline

1. Summary of Previous Lecture
2. WMSO Definability
3. Intermezzo
4. Myhill–Nerode Relations
5. **Further Reading**

Kozen

- Lectures 13 – 16

Kozen

- Lectures 13 – 16

Important Concepts

- coarse
- finite index
- Myhill–Nerode relation
- refinement
- right congruence
- (accepting) run

Kozen

- Lectures 13 – 16

Important Concepts

- coarse
- finite index
- Myhill–Nerode relation
- refinement
- right congruence
- (accepting) run

homework for November 8