Abschlusseigenschaften → Abschnitt 2.2.4

Abschlusseigenschaften für NFA/DFA erkennbare Sprachen

Nachweis: Automatenkonstruktionen Lemmata 2.2.11/14

Vereinigung
zu DFA $\mathcal{A}_1, \mathcal{A}_2$ existiert DFA $\mathcal{A}$ mit $L(\mathcal{A}) = L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$.

Durchschnitt
zu DFA $\mathcal{A}_1, \mathcal{A}_2$ existiert DFA $\mathcal{A}$ mit $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$.

Komplement
zu DFA $\mathcal{A}_1$ existiert DFA $\mathcal{A}$ mit $L(\mathcal{A}) = \overline{L(\mathcal{A}_1)}$.

Konkatenation
zu NFA $\mathcal{A}_1, \mathcal{A}_2$ existiert NFA $\mathcal{A}$ mit $L(\mathcal{A}) = L(\mathcal{A}_1) \cdot L(\mathcal{A}_2)$.

Stern-Operation
zu NFA $\mathcal{A}_1$ existiert NFA $\mathcal{A}$ mit $L(\mathcal{A}) = \big(L(\mathcal{A}_1)\big)^*$.

Abschlusseigenschaften

Durchschnitt und Vereinigung (für DFA)

zu $\mathcal{A}_1 = \big(\Sigma, Q^{(1)}, q_0^{(1)}, \delta^{(1)}, A^{(1)}\big)$
$\mathcal{A}_2 = \big(\Sigma, Q^{(2)}, q_0^{(2)}, \delta^{(2)}, A^{(2)}\big)$

Produktautomat $\mathcal{A} = \big(\Sigma, Q, q_0, \delta, A\big)$ mit

$$Q := Q^{(1)} \times Q^{(2)}$$
$$q_0 := (q_0^{(1)}, q_0^{(2)})$$
$$\delta\big((q_1, q_2), a\big) := \big(\delta^{(1)}(q_1, a), \delta^{(2)}(q_2, a)\big)$$

simuliert $\mathcal{A}_1/\mathcal{A}_2$ parallel in erster/zweiter Komponente

$$A := \begin{cases} A^{(1)} \times A^{(2)} & \text{für Durchschnitt} \\ (A^{(1)} \times Q^{(2)}) \cup (Q^{(1)} \times A^{(2)}) & \text{für Vereinigung} \end{cases}$$

Abschlusseigenschaften

Konkatenation (für NFA)

aus NFA $\mathcal{A}_1 = \big(\Sigma, Q^{(1)}, q_0^{(1)}, \Delta^{(1)}, A^{(1)}\big)$
$\mathcal{A}_2 = \big(\Sigma, Q^{(2)}, q_0^{(2)}, \Delta^{(2)}, A^{(2)}\big)$

mit $Q^{(1)} \cap Q^{(2)} = \emptyset$ und $q_0^{(1)} \notin A^{(1)}$ $(*)$

gewinne **Hintereinanderschaltung** als NFA $\mathcal{A} = \big(\Sigma, Q, q_0, \delta, A\big)$

$$Q := Q^{(1)} \cup Q^{(2)}$$
$$q_0 := q_0^{(1)}$$
$$A := A^{(2)}$$
$$\Delta := \Delta^{(1)} \cup \Delta^{(2)} \cup \Delta^{(1)\to(2)}$$

$\Delta^{(1)\to(2)} := \big\{(q, a, q_0^{(2)})\colon q \in Q^{(1)}, (q, a, q') \in \Delta^{(1)} \text{ für ein } q' \in A^{(1)}\big\}$

$(*)$: was ist andernfalls zu tun?

Abschlusseigenschaften Korollar 2.2.16

alle regulären Sprachen von NFA/DFA erkannt

per Induktion über reguläre Ausdrücke zeige:

$\big(\forall \alpha \in \mathrm{REG}(\Sigma)\big)$ $L(\alpha)$ Automaten-erkennbar

Induktionsanfang: $\alpha = \emptyset$ und $\alpha = \mathbf{a}$ für $a \in \Sigma$.
$L(\emptyset) = \emptyset$ und $L(\mathbf{a}) = \{a\}$ Automaten-erkennbar. (Übung!)

Induktionsschritte: von α_1, α_2 zu $\begin{cases} \alpha_1 + \alpha_2, \\ \alpha_1\alpha_2, \\ {\alpha_1}^* \end{cases}$

wenn $L(\alpha_1)$, $L(\alpha_2)$ Automaten-erkennbar sind,
so auch

$$\begin{cases} L(\alpha_1 + \alpha_2) = L(\alpha_1) \cup L(\alpha_2) \\ L(\alpha_1\alpha_2) = L(\alpha_1) \cdot L(\alpha_2) \\ L({\alpha_1}^*) = \big(L(\alpha_1)\big)^* \end{cases}$$

Satz von Kleene

→ Abschnitt 2.3

Satz 2.3.1 (Kleene's Theorem)

L regulär $\Leftrightarrow$ L DFA/NFA-erkennbar

reguläre Ausdrücke	—	Automaten-Berechnung
erzeugen (Sprache)	—	erkennen (Zugehörigkeit)
deskriptiv	—	prozedural
Syntax	—	Semantik

Übersicht

reguläre Σ-Sprachen	**NFA/DFA erkennbare Σ-Sprachen**
$L = L(\alpha)$: $\alpha \in \mathrm{REG}(\Sigma)$	$L = L(\mathcal{A})$: Σ-NFA/DFA $\mathcal{A}$
$L(\emptyset) = \emptyset$, $L(a) = \{a\}$, ...	$\emptyset$, $\{a\}$, ...
abgeschlossen unter	abgeschlossen unter
Vereinigung $\cup$ ja (triv)	Vereinigung $\cup$ ja
Konkatenation $\cdot$ ja (triv)	Konkatenation $\cdot$ ja
Stern-Operation * ja (triv)	Stern-Operation * ja
Durchschnitt $\cap$ **?**	Durchschnitt $\cap$ ja
Komplement $^-$ **?**	Komplement $^-$ ja

Satz von Kleene: dies sind alternative Beschreibungen derselben Sprachklasse

DFA/NFA erkennbare Sprachen sind regulär

zum Beweis vom Satz von Kleene (Satz 2.3.1)

Aufgabe: gewinne systematisch zu Σ-DFA/NFA $\mathcal{A}$ regulären Ausdruck $\alpha \in \mathrm{REG}(\Sigma)$ mit $L(\alpha) = L(\mathcal{A})$

Idee: sukzessive Berechnung von α' für Hilfssprachen L' so, dass kompliziertere α'/L' sich einfach aus einfacheren zusammensetzen (algorithmisch vgl. Idee des dynamischen Programmierens)

o.B.d.A. betrachte DFA $\mathcal{A} = (\Sigma, Q, q_0, \delta, A)$ mit $Q = \{1, \ldots, n\}$

zum Beweis vom Satz von Kleene

DFA $\mathcal{A} = (\Sigma, Q, q_0, \delta, A)$ $\qquad Q = \{1, \ldots, n\}$

zu $0 \leqslant k \leqslant n$ und $1 \leqslant \ell, m \leqslant n$ sei

$$L^k_{\ell,m} := \left\{ w \in \Sigma^* : \begin{array}{l} \mathcal{A} \text{ hat Lauf von Zustand } \ell \text{ nach Zustand } m \\ \text{auf } w \text{ über Zwischenzustände } q \in \{1, \ldots, k\} \end{array} \right\}$$

$$L^0_{\ell,m} = \begin{cases} \{a \in \Sigma : \delta(\ell, a) = m\} & \text{falls } \ell \neq m \\ \{\varepsilon\} \cup \{a \in \Sigma : \delta(\ell, a) = \ell\} & \text{falls } \ell = m \end{cases} \quad \text{(endlich)}$$

$$L^{k+1}_{\ell,m} = \underbrace{L^k_{\ell,m}}_{(1)} \cup \underbrace{L^k_{\ell,k+1}}_{(2)} \cdot \underbrace{(L^k_{k+1,k+1})^*}_{(3)} \cdot \underbrace{L^k_{k+1,m}}_{(4)}$$

(1) Läufe ohne Zustand $k+1$;
(2) Läufe von Zustand ℓ zum ersten $k+1$;
(3) Schleifen durch Zustand $k+1$;
(4) Läufe vom letzten $k+1$ nach m.

Folgerungen aus dem Satz von Kleene Korollar 2.3.2

die Klasse der regulären Sprachen ist abgeschlossen unter allen Booleschen Operationen sowie Konkatenation und Stern

alle Automaten-erkennbaren Sprachen lassen sich allein mit

- Vereinigung,
- Konkatenation und
- Stern

aus (einfachsten) endlichen Sprachen gewinnen

wieviele Zustände sind notwendig? → Abschnitt 2.4

Zustandszahlen von DFA $\mathcal{A}$ mit $L = L(\mathcal{A})$
als Maß für Komplexität von L

Grundidee zu *minimalem* DFA für L:

jeder Zustand beschreibt notwendige Information
verschiedene Zustände : *notwendige Unterscheidungen*

Methode: betrachte induzierte **Äquivalenzrelationen** auf Σ^*

$\sim_L$ zu gegebenem L $\quad w \not\sim_L w'$: "notwendige Unterscheidung"

$\sim_{\mathcal{A}}$ zu gegebenem $\mathcal{A}$ $\quad w \not\sim_{\mathcal{A}} w'$: "verschiedene Berechnungen"

die Äquivalenzrelation $\sim_L$

$L \subseteq \Sigma^*$ $\quad$ DFA $\mathcal{A} = (\Sigma, Q, q_0, \delta, A)$

$\sim_L$ zu $L \subseteq \Sigma^*$:

$$w \sim_L w' \quad \text{gdw} \quad (\forall x \in \Sigma^*)\,(wx \in L \Leftrightarrow w'x \in L)$$

- $\sim_L$ ist Äquivalenzrelation auf Σ^*:
  reflexiv, *symmetrisch*, *transitiv*
- $\sim_L$ ist *rechts-invariant*: $w \sim_L w' \;\Rightarrow\; wu \sim_L w'u$
- L besteht aus ganzen $\sim_L$-Äquivalenzklassen

die Äquivalenzrelation $\sim_{\mathcal{A}}$

$L \subseteq \Sigma^*$ $\quad$ DFA $\mathcal{A} = (\Sigma, Q, q_0, \delta, A)$

$\sim_{\mathcal{A}}$ zu DFA $\mathcal{A} = (\Sigma, Q, q_0, \delta, A)$

$$w \sim_{\mathcal{A}} w' \quad \text{gdw} \quad \hat{\delta}(q_0, w) = \hat{\delta}(q_0, w')$$

- $\sim_{\mathcal{A}}$ ist Äquivalenzrelation auf Σ^*:
  reflexiv, *symmetrisch*, *transitiv*
- $\sim_{\mathcal{A}}$ ist *rechts-invariant*: $w \sim_{\mathcal{A}} w' \;\Rightarrow\; wu \sim_{\mathcal{A}} w'u$
- $\sim_{\mathcal{A}}$ hat *endlichen Index*: $\operatorname{index}(\sim_{\mathcal{A}}) \leqslant |Q|$.

$\sim_L$ und $\sim_{\mathcal{A}}$ Korollare 2.4.5/6

für $L = L(\mathcal{A})$: $\sim_{\mathcal{A}}$ *Verfeinerung* von $\sim_L$:

$(\forall w, w' \in \Sigma^*)\ w \sim_{\mathcal{A}} w' \ \Rightarrow\ w \sim_L w'$

$$\boxed{\operatorname{index}(\sim_L) \leqslant \operatorname{index}(\sim_{\mathcal{A}}) \leqslant |Q|}$$

Korollare aus dem Vergleich von $\sim_L$ und $\sim_{\mathcal{A}}$

- L regulär $\Rightarrow$ $\sim_L$ hat endlichen Index
- für reguläres L: jeder DFA, der L erkennt, hat mindestens $\operatorname{index}(\sim_L)$ viele Zustände

Ziel: Satz von Myhill-Nerode

- $\sim_L$ hat endlichen Index $\Rightarrow$ L regulär
- für reguläres L: ex. DFA mit $\operatorname{index}(\sim_L)$ Zuständen für L

Myhill–Nerode → Abschnitt 2.4.1

der Äquivalenzklassen-Automat

Idee: assoziiere je einen Zustand mit jeder $\sim_L$-Äquivalenzklasse und erhalte minimalen DFA, der L erkennt

$[w] := \{v \in \Sigma^* : v \sim_L w\}$ die $\sim_L$-Äquivalenzklasse von w

$$\begin{aligned} \mathcal{A} = (\Sigma, Q, q_0, \delta, A) \qquad Q &:= \Sigma^*/\sim_L \\ q_0 &:= [\varepsilon] \\ \delta([w], a) &:= [wa] \qquad \text{(wohldefiniert!)} \\ A &:= \{[w] : w \in L\} \end{aligned}$$

$L = L(\mathcal{A})$ folgt aus: $(\forall w \in \Sigma^*)\hat{\delta}(q_0, w) = [w]$ (Induktion!)

Satz von Myhill-Nerode Satz 2.4.7

für $L \subseteq \Sigma^*$ sind äquivalent:

(i) $\sim_L$ hat endlichen Index.

(ii) L ist regulär.

Korollar aus dem Beweis:

kleinste DFA für reguläre L mit genau $\operatorname{index}(\sim_L)$ vielen Zuständen

Folgerung aus dem Satz:

$L \subseteq \Sigma^*$ nicht-regulär $\Leftrightarrow$

es gibt eine Folge $(w_n)_{n\in\mathbb{N}}$ in Σ^* mit $w_n \not\sim_L w_m$ für $n \neq m$

Beispiel: $L = \{a^n b^n : n \in \mathbb{N}\} \subseteq \{a, b\}^*$ nicht regulär

das syntaktische Monoid → Abschnitt 2.4.2

anstelle von $\sim_L$ betrachte die Verfeinerung $\approx_L$:

$$\boxed{w \approx_L w' \quad \text{gdw} \quad (\forall x, y \in \Sigma^*)\ (xwy \in L \Leftrightarrow xw'y \in L)}$$

- $\approx_L$ ist Äquivalenzrelation auf Σ^*
- $\approx_L$ ist Verfeinerung von $\sim_L$: $w \approx_L w' \ \Rightarrow\ w \sim_L w'$

  $\operatorname{index}(\sim_L) \leqslant \operatorname{index}(\approx_L)$
  $\operatorname{index}(\approx_L) \leqslant n^n$ wenn L von DFA mit n Zuständen erkennbar
- $\approx_L$ ist verträglich mit Konkatenation (*Kongruenzrelation*):

  $u \approx_L u'$ und $v \approx_L v' \ \Rightarrow\ uv \approx_L u'v'$

$\left(\Sigma^*/\approx_L, \cdot, [\varepsilon]_{\approx_L}\right)$ heisst **syntaktisches Monoid zu L**

$$\left.\begin{array}{ccc} \Sigma^* & \longrightarrow & \Sigma^*/\approx_L \\ w & \longmapsto & [w]_{\approx_L} \end{array}\right\} \text{ ist Monoid-Homomorphismus}$$