

EA und Reguläre Sprachen

Reguläre Sprachen

$L \subseteq \Sigma^*$ heisst regulär, wenn:

$L = a$, $a \in \Sigma$, $L = \epsilon$ oder $L = \emptyset$.

Seien L_1, L_2 reg. Sprachen, dann ist L auch regulär, wenn:

$L = L_1 \cup L_2$, $L = L_1 \cdot L_2$ oder $L = L_1^*$

Regulärer Ausdruck genauso, nur ohne $L(\cdot)$.

DEA Automatenminimierung

Suchen nach Zeugen für Nichtäquivalenz:

1. Betrachte alle Zustände als eine Klasse
2. Betrachte Wörter der Länge 0, also ϵ . Dieser trennt Q in $Q \setminus F$ und F .
3. Betrachte alle Wörter der Länge i und alle Klassen K . Teile K in K_1 und K_2 gdw. $\exists q_1, q_2 \in K$ und $w \in \Sigma^i$ mit $\delta(q_1, w) \in F$ und $\delta(q_2, w) \notin F$.
4. In Schritt (3) nichts getrennt? \rightarrow Fertig! Sonst wiederhole Schritt (3) mit Wörtern der Länge $i+1$.

DEA \rightarrow RegExp

DEA $A := / \{q_1, \dots, q_n\}, \Sigma, \delta, q_1, F$.

$L_f := \{w \in \Sigma^* \mid A \text{ endet nach Abarbeitung von } w \text{ in } f\}$

$L_{q_i, q_j}^m := \{w \in \Sigma^* \mid \text{Abarbeitung von } w \text{ aus } q_i \text{ nach } q_j \text{ hat nur}$

Zwischenzustände $\{q_1, \dots, q_m\}$

$L(A) = \bigcup_{f \in F} L_f = \bigcup_{f \in F} L_{q_1, f}^n$

$L(A)$ ist ind. definiert:

Falls $m=0$ und $q_j = q_i$: $L_{q_i, q_j}^0 := \{a \in \Sigma \mid \delta(q_i, a) = q_i\} \cup \{\epsilon\}$

Falls $m=0$ und $q_j \neq q_i$: $L_{q_i, q_j}^0 := \{a \in \Sigma \mid \delta(q_i, a) = q_j\}$

Sonst ($m > 0$): $L_{q_i, q_j}^{m+1} := L_{q_i, q_j}^m \cup (L_{q_i, q_{m+1}}^m (L_{q_{m+1}, q_{m+1}}^m)^* L_{q_{m+1}, q_j}^m$

Nerode Relation zur Sprache L

$R_L := \{(x, y) \in \Sigma^* \times \Sigma^* \mid \forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\}$

R_M verfeinert R_L !

Satz v. Nerode: $Index(R_L) = \infty \rightarrow L$ ist nicht regulär! Es gilt sogar \Leftrightarrow

Pumping Lemma fuer reg. Sprachen

L regulär \rightarrow

$\exists n \in \mathbb{N} : \forall w \in L, |w| > n :$

$\exists u, v, x : w = uvx, |uv| < n, |v| > 0, \forall i \in \mathbb{N}_0 : uv^i x \in L$

Berechenbarkeit

TM

Schreibweise: $\delta(q, a) = \delta(q, a, N)$

Beim Graph: $Eingabe \mid Ausgabe, \{L, R, N\}$

Endzustände kennzeichnen, **Startzustand nicht!**

Def

T akzeptiert w

$\Leftrightarrow T$ akzeptiert alle $w \in L$ und kein $w \notin L$

L ist **rekursiv aufzählbar** (= **semientscheidbar**)

$\Leftrightarrow \exists T : T$ akzept. L

L ist **rekursiv** (= **entscheidbar**)

$\Leftrightarrow \exists T : T$ akzept. $L \wedge \forall w \in \Sigma^* : T$ hält

Funktion $g : \Sigma^* \rightarrow \Gamma^*$ ist (**turing**)-**berechenbar** wenn $g(w)=y$ genau dann, wenn es eine TM gibt, für die $(s)w \Rightarrow^* x(f)y$, wobei $x, y \in \Gamma^*$

Sprache $L \subseteq \Sigma^*$ ist **entscheidbar** gdw ihre charakteristische Funktion χ_L ist berechenbar. Dabei gilt:

$\chi_L : \Sigma^* \rightarrow \{0, 1\}$ mit $\chi_L(w) = \{1, \text{ falls } w \in L, \text{ sonst } 0$

Sprache $L \subseteq \Sigma^*$ ist **semi-entscheidbar** gdw die Funktion χ_L^* ist berechenbar. Dabei gilt:

$\chi_L^* : \Sigma^* \rightarrow \{0, 1\}$ mit $\chi_L^*(w) = \{1, \text{ falls } w \in L, \text{ sonst undefiniert}$

Diagonalsprache L_d

Sei M_i die TM mit Gödelnummer $\langle M_i \rangle = i$ und w_i die Binärrepräsentation von i .

$L_d := \{w_i : M_i \text{ akzeptiert } w_i \text{ nicht}\}$

Satz: L_d und ihr Komplement sind beide nicht entscheidbar.

L_d ist auch nicht semientscheidbar.

Universelle TM U akzeptiert $\langle M \rangle w$ gdw M akzeptiert w .

Halteproblem $H := \{w_i v : M_i \text{ angesetzt auf } v \text{ hält}\}$ ist nicht entscheidbar.

Beschränktes Halteproblem (nach x Schritten) ist aber entscheidbar.

Postches Korrespondenzproblem (PKP)

PKP ist semientscheidbar und nicht entscheidbar.

Geg: Endliche Anzahl von Wortpaaren

Frage: Existiert Anordnungsmoeglichkeit, so dass oben und unten gleich (nichts darf ueberstehen).

WHILE-Programm / LOOP Programm Aufbau

\mathbb{N} main($\mathbb{N} x_1, \dots, \mathbb{N} x_k$)

$\mathbb{N} x_0; \mathbb{N} x_{k+1}; \mathbb{N} x_{k+2}; \dots$

body;

return x_0 ;

}

Im Body: $x_i := x_j + c$, $c \in \{-1, 0, 1\}$, wichtig: $0 - 1 := 0$.

Erlaubt: Sequenzen ";", Unterprogrammaufrufe.

while($x_j \neq 0$): *Schleife*

loop x_i : Schleife. Wiederhole x_i mal. Relevant ist der Inhalt von x_i vor der ersten Schleifenausführung.

Ackermannfunktion $a(x, y)$

$a(0, y) = y + 1$

$a(x, 0) = a(x - 1, 1)$

$a(x, y) = a(x - 1, a(x, y - 1))$

Die Ackermannfunktion ist nicht Loop-berechenbar. Sie wächst schneller als jede LOOP berechenbare Funktion.

Lemma: $y < a(x, y)$

$a(x, y) < a(x, y + 1)$

$a(x, y + 1) \leq a(x + 1, y)$

$a(x, y) < a(x + 1, y)$

$a(x, y) \leq a(x', y')$, falls $x \leq x'$ und $y \leq y'$

Inverse Ackermannfunktion

$\alpha(m, n) := \min\{i \geq 0 : a(i, \text{floor}(m/n))\} > \log_2 n$. Hier gilt fast immer $\alpha(m, n) \leq 5$, aber $\alpha(m, n) \notin O(1)$.

Komplexität

P

$\text{time}_M(w) = \#$ Rechenschritte einer TM bei Eingabe von w .

$\text{TIME}(f(n)) = \{L : \exists (\text{Mehrband-})TM M : L(M) =$

$L \wedge \forall w \in \Sigma_M^* : \text{time}_M(w) \leq f(|w|)\}$

$P := \bigcup_{\text{Polynom } p} \text{TIME}(p(n))$

co-NP

$\text{co-NP} = \{L^c, L \subseteq \Sigma^*, L \in P\}$

Pseudopolynomielle Algorithmen

A ist ein pseudopolynomieller Algorithmus, falls $\text{Time}_A(p) \in P$.

Dabei ist n die Anzahl der Eingabebits unär kodiert (l^n).

Polynomiale Reduzierbarkeit

$A \subseteq \Sigma^*, B \subseteq \Gamma^*$ Sprachen.

$A \leq_p B$ (A ist auf B polynomial reduzierbar) \Leftrightarrow

$\exists f : \Sigma^* \rightarrow \Gamma^* : \forall w \in \Sigma^* : w \in A \Leftrightarrow f(w) \in B$,

wobei f in polynomialer Zeit berechenbar ist.

Defs

A ist **NP-Hart**: $\Leftrightarrow \forall L \in NP : L \leq_p A$

A ist **NP-vollständig** $\Leftrightarrow A$ ist NP-Hart und $A \in NP$.

Reduktionsbeweise: Schema für NPV Problem A

- Zeige $A \in NP$ durch Algorithmus(-ansatz), großzügig abschätzen.
- Zeige A ist NP-Hart, also hier $Q \leq_p A$
 - Wähle bekanntes NPV Problem Q
 - Nehme Instanz von Q und erzeuge daraus Instanz von A
 - Zeige (begründe), dass wenn Graph G (oder ähnliches) keine Instanz von Q hat, so gibt es auch keine von A oder erzeuge aus Instanz von A eine Instanz von Q .
 - Begründe, dass die Transformation polynomial ist.

Folgerungen

$A \leq_p B \wedge B \in P \Rightarrow A \in P$

$A \leq_p B \wedge A \in NPV \Rightarrow B \in NP - \text{Hart}$

$A \leq_p B \wedge B \in NPV \Rightarrow A \in NP$

$A \leq_p B \wedge A \in P \Rightarrow B \neq \emptyset \wedge B \neq \Sigma^*$

NP-vollständige Problem

SAT

Gegeben: Formel F der Aussagen Logik mit

($\wedge, \vee, \neg, (,)$, Variablen)

Frage: Gibt es eine Variablenbelegung, so dass F erfüllbar ist?

3 SAT (KNF)

Gegeben: Formel F der Aussagen Logik mit maximal 3 Literalen pro Klausel. ($\wedge, \vee, \neg, (,)$, Variablen)

Frage: Gibt es eine Variablenbelegung, so dass F erfüllbar ist?

SET COVER

Gegeben: Grundmenge M , Teilmengen T_i von $\tau \in 2^M$, Parameter n

Frage: Gibt es Auswahl von $T_1 \in \tau, \dots, T_n \in \tau$ mit

$T_1 \cup \dots \cup T_n = M$? (Mengen müssen nicht disjunkt sein)

CLIQUE

Gegeben: ungerichteter Graph $G=(V(\text{Knoten}), E(\text{Kanten}))$ und ein Parameter k

Frage: $\exists V' \subseteq V : |V'| = k \wedge \forall u \neq v \in V' : \{u, v\} \in E$?

D.h. alle Knoten in der Clique sind mit allen Knoten in der Clique verbunden.

VERTEX COVER

Gegeben: ungerichteter Graph $G=(V(\text{Knoten}), E(\text{Kanten}))$ und ein Parameter k

Frage : $\exists V' \subseteq V : |V'| = k \wedge \forall \{u, v\} \in E : u \in V' \vee v \in V'$
D.h. alle Knoten, die mindestens ein Kantenende enthalten, von jeder Kante.

COLORING

Gegeben : ungerichteter Graph $G=(V(\text{Knoten}),E(\text{Kanten}))$ und ein Parameter k

Frage : Gibt es eine Knotenfärbung $c : V \rightarrow \{1, \dots, k\}$ mit $\forall u, v \in E : c(u) \neq c(v)$?

Geht schon für 3-COLORING, also mit $k=3$. 3-COLORING ist NPV.

SUBSET SUM

Gegeben : n Gegenstände mit Gewicht $w_i \in \mathbb{N}$ und ein Parameter W

Frage : Gibt es eine Teilmenge M von $\{1, \dots, n\}$, so dass

$$\sum_{i \in M} w_i = W$$

RUCKSACK

Gegeben : n Gegenstände mit Gewicht $w_i \in \mathbb{N}$, Profit p_i und ein Parameter W

Frage : Wähle eine Teilmenge x von Gegenstände, so dass

$$\sum_{i \in x} w_i \leq W \text{ und } \sum_{i \in x} p_i \geq W?$$

Partition

Gegeben : n Gegenstände mit Gewicht $w_i \in \mathbb{N}$

Frage : Gibt es eine Teilmenge M von $\{1, \dots, n\}$, so dass

$$\sum_{i \in M} w_i = \sum_{i \notin M} w_i$$

BIN PACKING

Gegeben : Plätzchdose $b \in \mathbb{N}$, Plätzchen der Größe $w_1, \dots, w_n \in \mathbb{N}$ und Anzahl der Plätzchensosen k

Frage : Passen alle Plätzchen irgendwie in alle Dosen? Also

$$\exists f : 1 \dots n \rightarrow 1 \dots k : \forall j \in 1 \dots k : \sum w_i : f(i) = j \leq b$$

Integer Linear Programming (ILP)

Gegeben : Vektoren $x = (x_1, \dots, x_n)$, Menge von Bedingungen der Form $a \cdot x \leq b$ mit $R \in \{\leq, \geq, =\}$, $b \in \mathbb{Z}$, $a \in \mathbb{Z}^n$

Frage : Gibt es eine Belegung für x aus \mathbb{Z}^n , so dass alle Bedingungen erfüllt sind?

DIRECTED HAMILTON CYCLE (DHC)

Gegeben : gerichteter Graph $G=(V(\text{Knoten}),E(\text{Kanten}))$

Frage : Gibt es einen Hamiltonschen Kreis? D.h. jede Ecke des Graphen ist genau einmal enthalten.

$$M := \{G = (V, E \subseteq V \times V) : \exists C \subseteq E : |C| = |V|, C \text{ ist einfacher Kreis}\}$$

HAMILTON CYCLE (HC)

Gegeben : ungerichteter Graph $G=(V(\text{Knoten}),E(\text{Kanten}))$

Frage : Gibt es einen Hamiltonschen Kreis? D.h. jede Ecke des Graphen ist genau einmal enthalten.

$$M := \{G = (V, E \subseteq V \times V) : \exists C \subseteq E : |C| = |V|, C \text{ ist einfacher Kreis}\}$$

TRAVELLING SALESMAN PROBLEM (TSP)

Gegeben : ungerichteter Graph $G=(V(\text{Knoten}),E(\text{Kanten}))$ und ein Parameter M

Frage : Gibt es einen einfachen Kreis $C = (v_1, v_2, \dots, v_n, v_1)$ (Reihenfolge nicht nach Nummerierung), so dass $n = |V|$ und

$$\sum_{(u,v) \in C} d(u, v) \leq M?$$

STEINER TREE

Gegeben : ungerichteter Graph $G=(V(\text{Knoten}),E(\text{Kanten}))$ mit positiven Kantengewichten $c : E \rightarrow \mathbb{R}^+$, $V = R(\text{Pflichtknoten}) \cup F(\text{SteinerKnoten, nichtPflicht})$ und ein Parameter M

Frage : Gibt es einen Baum $T \subseteq E$, der mit Kosten $\leq M$ alle Pflichtknoten verbindet?

Chomsky Grammatiken

Übergangsrelation, L(G)

Grammatik $G = (V, \Sigma, P, S)$, $u \Rightarrow_G v$ falls

$$u = xyz \in (V \cup \Sigma)^*, v = xy'z \in (V \cup \Sigma)^* \text{ und } y \rightarrow y' \in P.$$

Chomsky-Hierarchie

$$G = (V, \Sigma, P, S) \forall l \rightarrow r \in P$$

Typ-0 beliebig

Typ-1 $|l| \leq |r|$, zusätzlich $S \rightarrow \epsilon$

Typ-2 Typ 1 und $l \in V$, zusätzlich $S \rightarrow \epsilon$

Typ-3 Typ 2 und $r \in \Sigma \cup \Sigma V$, zusätzlich $S \rightarrow \epsilon$

Chomsky- Normalform für ktf. Grammatiken (ohne ϵ)

- Elimination von zyklischen Einheitsproduktionen $A \rightarrow B \rightarrow A$
- Elimination von nichtzyklischen Einheitsproduktionen $A \rightarrow B$
- Elimination gemischer rechter Seiten $A \rightarrow aBc$
- Elimination zu langer rechter Seiten von links $A \rightarrow BCD \Rightarrow A \rightarrow HD, H \rightarrow BC$

Wortproblem $w \in L(G)$ und CYK Algorithmus

Spezialfall $s = \epsilon$ ist einfach. Sonst:

- Forme Grammatik $G = (V, \Sigma, P, S)$ in Chomsky-Normalform um.

- Schreibe w mit viel Abstand zwischen den Buchstaben in eine Zeile
- Schreibe alle Nichtterminale, aus dem man den Buchstaben erzeugen kann unter die jeweiligen Buchstaben.
- Danach prüfe gleiche Spalte ganz oben (i) und Diagonal rechts drüber (ii) zusammen, weiter: mit (i) 1 nach unten und mit (ii) 1 Diagonal nach schräg rechts oben.
- Erzeuge umgekehrte Treppe. Falls S in der Spitze, dann $w \in L$.

Pumpinglemma für kontextfreie Sprachen

L kontextfrei \rightarrow

$$\exists n \in \mathbb{N} : z \in L, |z| > n :$$

$$\exists u, v, w, x, y : z = uvwxy, |vwx| \leq n, |vx| > 0,$$

$$\forall i \in \mathbb{N}_0 : uv^iwx^iy \in L$$

Nichtdet. Kellerautomaten

$$K = (Q, \Sigma, \Gamma, \delta, s, \#), (q, w, x) \in Q \times \Sigma^* \times \Gamma^*$$

Bei einem Übergang werden 0 oder 1 Zeichen von der Eingabe gelesen, 1 Zeichen vom Keller genommen und dieses Zeichen wird durch ein (ggf. gleiches) Zeichen ersetzt.

Kellerautomat akzeptiert Wort w gdw $\exists (s, w, \#) \rightarrow \dots \rightarrow (q, \epsilon, \epsilon)$ mit $q \in Q$

L kontextfrei gdw $\exists NKellerA M$ mit $L(M) = L$. Geht auch mit 1Zustand NKellerA.

Probleme für L(G)

Leerheitsproblem $L(G) = \emptyset?$

Endlichkeitsproblem $L(G) = \infty?$

Wortproblem $w \in L(G)?$

Linear beschränkte Nichtdet. Turingmaschine (NLBTM)

NTM $T = \{Q, \Sigma, \Gamma, \delta, s, F\}$ ist linear beschränkt, wenn

$$\forall a = a_1 \dots a_n \in \Sigma^+ : a \Rightarrow^* \alpha(q)\beta \rightarrow |\alpha\beta| \leq n$$

Überblick

Chomsky Typ und Beschreibungsmittel

CH-Typ	Beschreibungsmittel
3	rechtslin. XOR linkslin. Gramm., DEA, NEA, reg. Ausdr.
Det.KF	LR(k) Grammatik, DKellerA mit Endzustand
2	kontextfreie Grammatiken, (1Zustands)NKellerA
1	kontextsens. (expand.) Grammatiken $ l \leq r $, NLBTM
0	Typ-0 Grammatik, NTM, DTM, (semientscheidbar)

Abschlusseigenschaften

Typ	\cap	\cup	\cdot^- Kompl.	\cdot Konkat	$*$ Kleen.
3	ja	ja	ja	ja	ja
Det.KF	nein	nein	ja	nein	nein
2	nein	ja	nein	ja	ja
1	ja	ja	ja	ja	ja
0	ja	ja	nein	ja	ja

Entscheidbarkeitsprobleme

Typ	Wort-	Leerheits-	Äquivalenz	Schnitt
3	ja	ja	ja	ja
Det.KF	ja	ja	ja	nein
2	ja	ja	nein	nein
1	ja	nein	nein	nein
0	nein	nein	nein	nein

Komplexität des Wortproblems

CH-Typ	Komplexität des Wortproblems
3	$O(n)$
Det.KF	$O(n)$
2	$O(n^3)$
1	$ \Sigma ^{O(n)}$, NP-Hart
0	semientscheidbar

Approximationsalgorithmen

Sei L ein Optimierungsproblem und $I \in L$ eine Instanz von L . $OPT(I)$ - die Optimallösung von I .

$A(I)$ - die Lösung die ein Algorithmus A zu I liefert.

Zu einem Minimierungsproblem L ist ein Algorithmus A ein Approximationsalgorithmus mit relativer Gütegarantie p , falls gilt $A(I)/OPT(I) \leq p \forall I \in L$.

(Fully) Polynomial Time Approximation Scheme ((F)PTAS)

Algorithmus A ist PTAS für Problem I , falls

1. Eingabe: Instanz I , Fehlerparameter ϵ
 2. $f(x) \leq (1 + \epsilon) \cdot OPT$
 3. Time ist polynomiell in $|I|$
- Falls Time polynomiell in $|I|$ und $1/\epsilon$, dann ist A FPTAS.

Fixpunktsatz

1. Das Produkt zweier Wortmengen ist stetig.
2. Die Vereinigung zweier Wortmengen ist stetig.
3. Jede stetige Funktion ist monoton.
4. Jede stetige Funktion hat Fixpunkte.