Formale Sprachen und Komplexitätstheorie

WS 2019/20

Robert Elsässer

Laufzeit einer DTM

Definition

DTM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{n-1}, q_n)$ halte bei jeder Eingabe.

- Für w aus Σ^* ist $T_M(w)$ die Anzahl der Rechenschritte von M bei Eingabe w.
- Für eine natürliche Zahl n ist $T_M(n) := \max\{T_M(w) \mid w \text{ aus } \Sigma^{\leq n}\}$.
- Die Funktion T_M heißt Zeitkomplexität oder Laufzeit der DTM M.
- DTM M hat Laufzeit $O\big(f(n)\big)$, wenn $T_M(n) = O\big(f(n)\big)$.

Satz

Sei t eine monoton wachsende Funktion mit $t(n) \geq n$.
Jede Mehrband-DTM mit Laufzeit $t(n)$ kann durch eine 1-Band-DTM mit Laufzeit $O(t(n)^2)$ simuliert werden.

Verifizierer

Definition

Sei L eine Sprache. DTM V heißt Verifizierer für L, falls

$$L = \{w \mid \text{es gibt ein } c, \text{so dass } V \langle w, c \rangle \text{ akzeptiert}\}$$

c: Zertifikat oder Zeuge

V heißt polynomieller Verifizierer, falls eine natürliche Zahl k existiert mit

$$L = \{w \mid \text{es gibt ein } c \text{ mit } |c| \leq |w|^k, \text{sodass } V \langle w, c \rangle \text{ akzeptiert}\}$$

und die Laufzeit von V bei Eingabe $\langle w, c \rangle$ polynomiell in $|w|$ ist.

L heißt dann polynomiell verifizierbar.

Klasse NP

Definition

NP ist die Klasse der Sprachen, die polynomiell verifizierbar sind.

RS_{ent}, TSP_{ent} sind in NP.

Satz

P ist eine Teilmenge von NP.

Millenium-Problem

Ist $P = NP$? (Clay Mathematics Institute)

Nichtdeterministische Turingmaschinen

NTM $N = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ ist in Konfiguration $K = \alpha q \beta$, wenn gilt:

1. auf dem Band von N steht $\alpha\beta$, gefolgt von Blanks,
2. N befindet sich im Zustand q,
3. der Lesekopf von N steht auf dem ersten Symbol von β.

NTM – Rechenschritt

1. NTM N in Konfiguration $K = \alpha q a \beta$.

2. $\delta(q, a) = \{(q_1, b_1, D_1), \ldots, (q_l, b_l, D_l)\}$.

3. N kann jeden durch ein Tripel (q_i, b_i, D_i) aus $\delta(q, a)$ beschriebenen Rechenschritt ausführen.

Akzeptieren und Entscheiden

Definition

Sei N eine NTM. N akzeptiert w, wenn es mindestens eine akzeptierende Berechnung von N bei Eingabe w gibt.

NTM N hält bei Eingabe w, wenn alle Berechnungspfade von N bei Eingabe w endliche sind.

Definition

Die von einer NTM N akzeptierte Sprache $L(N)$ ist definiert als

$$L(N) \coloneqq \{w \mid N \text{ akzeptiert } w\}$$

NTM N akzeptiert die Sprache L, falls $L = L(N)$. N entscheidet die von ihr akzeptierte Sprache $L(N)$, wenn N immer hält.

Laufzeit einer NTM

Definition

Sei N eine NTM, die immer hält.

- Für w ist $T_N(w)$ die maximale Anzahl von Rechenschritten in einer Berechnung von N bei Eingabe w.
- Für eine natürliche Zahl n ist $T_N(n) \coloneqq \max\{T_N(w) \mid w \text{ aus } \Sigma^{\leq n}\}$.
- Die Funktion T_N heißt Zeitkomplexität oder Laufzeit der NTM N.
- N hat Laufzeit $O\big(f(n)\big)$, wenn $T_N(n) = O\big(f(n)\big)$.

Nichtdeterministische Zeitkomplexität

Definition

Sei t eine monoton wachsende Funktion. Die Klasse $NTIME\big(t(n)\big)$ ist dann definiert als

$$NTIME\big(t(n)\big) := \left\{ \begin{matrix} L \mid L \text{ ist eine Sprache, die von einer NTM} \\ \text{mit Laufzeit } O\big(t(n)\big) \text{ entschieden wird} \end{matrix} \right\}$$

Satz

NP ist die Klasse der Sprachen, die von einer nichtdeterministischen Turingmaschine mit polynomieller Laufzeit entschieden werden, d.h.,

$$NP = \bigcup_k NTIME\big(n^k\big)$$

Simulation einer NTM durch eine DTM

Satz

Sei t eine monoton wachsende Funktion mit $t(n) \geq n$ für alle natürlichen Zahlen n. Für jede NTM mit Laufzeit $t(n)$ gibt es eine DTM mit Laufzeit $2^{O(t(n))}$, die dieselbe Sprache entscheidet.

Polynomielle Reduktion

Definition

Sei Σ ein Alphabet. Eine Funktion $f: \Sigma^* \to \Sigma^*$ heißt polynomiell berechenbar, wenn es eine DTM M mit polynomieller Laufzeit gibt, die f berechnet.

Definition

Seien A, B zwei Sprachen. A heißt auf B polynomiell reduzierbar, wenn es eine polynomiell berechenbare Funktion f gibt mit:

$$w \text{ in } A \Leftrightarrow f(w) \text{ in B}$$

Die Funktion f wird polynomielle Reduktion genannt und man schreibt

$$A \leq_P B$$

Polynomielle Reduktion – Eigenschaften

Satz

Seien A, B zwei Sprachen. Gilt $A \leq_P B$ und B ist in P, so ist auch A in P.

Lemma

Die Relation $\leq_P$ ist transitiv.

Boolesche Variablen, Operatoren, Formeln

- **Boolesche Variablen x** können die beiden Werte wahr (1) oder falsch (0) annehmen
- **Boolesche Operatoren:** und $(\wedge)$; oder $(\vee)$; nicht $(\neg)$.
- **Boolesche Formeln:** Ausdruck bestehend aus Booleschen Variablen und Operatoren, korrekt formatiert.

Beispiel

$$\varphi = (\neg x \wedge y) \vee (x \wedge \neg z)$$

Boolesche Variablen, Operatoren, Formeln

- **Boolesche Formel φ** heißt erfüllbar, wenn es eine Belegung der Variablen in φ mit 1 und 0 gibt, sodass die Formel dann wahr ist.

Beispiel

$\varphi = (\neg x \wedge y) \vee (x \wedge \neg z)$ ist erfüllbar. Belegung $x = 0, y = 1, z = 0$.

Beispiel

$\varphi = (x \wedge \neg x) \vee (y \wedge \neg y)$ ist nicht erfüllbar.

Die Sprache SAT

Definition

$SAT \coloneqq \{\varphi \mid \varphi \text{ ist eine erfüllbare Boolesche Formel}\}$

Satz

SAT liegt in NP.

Boolesche Variablen, Operatoren, Formeln

- Literale sind Boolesche Variablen oder Negationen Boolescher Variablen.
- Eine Klausel ist die Disjunktion von Literalen.
- Eine Formel ist in konjunktiver Normalform (KNF), wenn sie die Konjunktion von Klauseln ist.
- In 3-KNF enthält jede Klausel 3 Literale.

Definition

$3SAT \coloneqq \{\varphi \mid \varphi \text{ ist eine erfüllbare } 3\text{−KNF Formel}\}$

Graphen und Cliquen

Definition

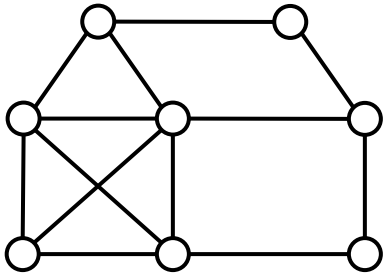
Sei $G = (V, E)$ ein ungerichteter Graph. Eine Teilmenge C von V heißt Clique, wenn alle Knoten aus C miteinander verbunden sind. C heißt k-Clique, wenn C genau k Knoten hat.

Definition

$Clique \coloneqq \{(G, k) \mid G \text{ ist ein ungerichteter Graph mit einer } k\text{−Clique}\}$

Polynomielle Reduktion – Eigenschaften

$G = (V, E)$



$(G, 4) \in Clique$

$(G, 5) \notin Clique$

Satz

$3SAT$ ist auf Clique polynomiell reduzierbar.

NP-Vollständigkeit

Definition

Eine Sprache L heißt NP-vollständig, wenn sie die folgenden Bedingungen erfüllt:

- L ist in NP
- Für jede Sprache L' aus NP gilt: $L' \leq_P L$

Satz

Ist L NP-vollständig und in P, so gilt $P = NP$.

Satz

Ist L in NP und gilt $L' \leq_P L$ für eine Sprache L', die NP-vollständig ist, so ist auch L NP-vollständig.