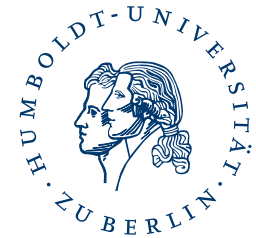
Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2022/23

Zeitkomplexität von Turingmaschinen

Die Laufzeit einer NTM M bei Eingabe x ist die maximale Anzahl an Rechenschritten, die $M(x)$ ausführt

Definition

- Die Laufzeit einer NTM M bei Eingabe x ist definiert als

$$time_M(x) = \sup\{t \geq 0 \mid \exists K : K_x \vdash^t K\},$$

  wobei $\sup \mathbb{N} = \infty$ ist
- Sei $t : \mathbb{N} \to \mathbb{N}$ eine monoton wachsende Funktion
- Dann ist M $t(n)$-zeitbeschränkt, falls für alle Eingaben x gilt:

$$time_M(x) \leq t(|x|)$$

Die Zeitschranke $t(n)$ beschränkt also die Laufzeit bei allen Eingaben der Länge n (worst-case Komplexität)

Zeitkomplexitätsklassen

Wir fassen alle Sprachen und Funktionen, die in einer vorgegebenen Zeitschranke $t(n)$ entscheidbar bzw. berechenbar sind, in folgenden Komplexitätsklassen zusammen

Definition

- Die in deterministischer Zeit $t(n)$ entscheidbaren Sprachen bilden die Sprachklasse

$$\mathrm{DTIME}(t(n)) = \{L(M) \mid M \text{ ist eine } t(n)\text{-zeitbeschränkte DTM}\}$$

- Die in nichtdeterministischer Zeit $t(n)$ entscheidbaren Sprachen bilden die Sprachklasse

$$\mathrm{NTIME}(t(n)) = \{L(M) \mid M \text{ ist eine } t(n)\text{-zeitbeschränkte NTM}\}$$

- Die in deterministischer Zeit $t(n)$ berechenbaren Funktionen bilden die Funktionenklasse

$$\mathrm{FTIME}(t(n)) = \left\{ f \;\middle|\; \begin{array}{l} \text{es gibt eine } t(n)\text{-zeitbeschränkte} \\ \text{DTM } M, \text{ die } f \text{ berechnet} \end{array} \right\}$$

Die wichtigsten Zeitkomplexitätsklassen

- Die wichtigsten deterministischen Zeitkomplexitätsklassen sind

$$\mathrm{LINTIME} = \bigcup_{c \geq 1} \mathrm{DTIME}(cn + c) \qquad \text{„Linearzeit“}$$

$$\mathrm{P} = \bigcup_{c \geq 1} \mathrm{DTIME}(n^c + c) \qquad \text{„Polynomialzeit“}$$

$$\mathrm{E} = \bigcup_{c \geq 1} \mathrm{DTIME}(2^{cn+c}) \qquad \text{„Lineare Exponentialzeit“}$$

$$\mathrm{EXP} = \bigcup_{c \geq 1} \mathrm{DTIME}(2^{n^c+c}) \qquad \text{„Exponentialzeit“}$$

- Die nichtdeterministischen Klassen NLINTIME, NP, NE, NEXP und die Funktionenklassen FLINTIME, FP, FE, FEXP sind analog definiert
- Für eine Klasse $\mathcal{F}$ von Funktionen sei $\mathrm{DTIME}(\mathcal{F}) = \bigcup_{t \in \mathcal{F}} \mathrm{DTIME}(t(n))$ (die Klassen $\mathrm{NTIME}(\mathcal{F})$ und $\mathrm{FTIME}(\mathcal{F})$ sind analog definiert)

Asymptotische Laufzeit und Landau-Notation

Definition

Seien f und g Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+ \cup \{0\} = [0, \infty)$

- Wir schreiben $f(n) = \mathcal{O}(g(n))$, falls es Zahlen n_0 und c gibt mit

  $$\forall n \geq n_0 : f(n) \leq c \cdot g(n)$$

  Bedeutung: „f wächst nicht wesentlich schneller als g“

- Formal bezeichnet der Term $\mathcal{O}(g(n))$ die Klasse aller Funktionen f, die obige Bedingung erfüllen, d.h.

  $$\mathcal{O}(g(n)) = \{f : \mathbb{N} \to [0, \infty) \mid \exists n_0, c \in \mathbb{N}\, \forall n \geq n_0 : f(n) \leq c \cdot g(n)\}$$

- Die Gleichung $f(n) = \mathcal{O}(g(n))$ drückt also in Wahrheit eine Element-Beziehung $f \in \mathcal{O}(g(n))$ aus
- O-Terme können auch auf der linken Seite vorkommen. In diesem Fall wird eine Inklusionsbeziehung ausgedrückt
- So steht $n^2 + \mathcal{O}(n) = \mathcal{O}(n^2)$ für die Aussage

  $$\{n^2 + f \mid f \in \mathcal{O}(n)\} \subseteq \mathcal{O}(n^2)$$

Asymptotische Laufzeit und Landau-Notation

Beispiel

- $7\log(n) + n^3 = \mathcal{O}(n^3)$ ist richtig
- $7\log(n)n^3 = \mathcal{O}(n^3)$ ist falsch
- $2^{n+\mathcal{O}(1)} = \mathcal{O}(2^n)$ ist richtig
- $2^{\mathcal{O}(n)} = \mathcal{O}(2^n)$ ist falsch (siehe Übungen)

Mit der O-Notation lassen sich die wichtigsten deterministischen Zeitkomplexitätsklassen wie folgt charakterisieren:

$$\begin{aligned} \text{LINTIME} &= \text{DTIME}(\mathcal{O}(n)) && \text{„Linearzeit“} \\ \text{P} &= \text{DTIME}(n^{\mathcal{O}(1)}) && \text{„Polynomialzeit“} \\ \text{E} &= \text{DTIME}(2^{\mathcal{O}(n)}) && \text{„Lineare Exponentialzeit“} \\ \text{EXP} &= \text{DTIME}(2^{n^{\mathcal{O}(1)}}) && \text{„Exponentialzeit“} \end{aligned}$$

Das P-NP-Problem

- Wie wir gesehen haben, sind NTMs nicht mächtiger als DTMs, d.h. jede NTM kann von einer DTM simuliert werden
- Die Frage, wieviel Zeit eine DTM zur Simulation einer NTM benötigt, ist eines der wichtigsten offenen Probleme der Informatik
- Wegen $\mathrm{NTIME}(t) \subseteq \mathrm{DTIME}(2^{\mathcal{O}(t)})$ erhöht sich die Laufzeit im schlimmsten Fall exponentiell
- Insbesondere die Klasse NP enthält viele für die Praxis überaus wichtige Probleme, für die kein Polynomialzeitalgorithmus bekannt ist
- Für viele dieser Probleme A konnte folgende Implikation gezeigt werden:

  $$A \in \mathrm{P} \Rightarrow \mathrm{P} = \mathrm{NP}$$

- Da jedoch nur Probleme in P als effizient lösbar angesehen werden, hat das P-NP-Problem, also die Frage, ob NP = P ist, eine immense praktische Bedeutung

Die Polynomialzeitreduktion

Definition

- Eine Sprache $A \subseteq \Sigma^*$ ist auf $B \subseteq \Gamma^*$ in Polynomialzeit reduzierbar ($A \leq^p B$), falls eine Funktion $f : \Sigma^* \to \Gamma^*$ in FP existiert mit

$$\forall x \in \Sigma^* : x \in A \Leftrightarrow f(x) \in B$$

- Eine Sprache A heißt $\leq^p$-hart für eine Sprachklasse $\mathcal{C}$ (kurz: $\mathcal{C}$-hart oder $\mathcal{C}$-schwer), falls gilt:

$$\forall L \in \mathcal{C} : L \leq^p A$$

- Eine $\mathcal{C}$-harte Sprache A, die zu $\mathcal{C}$ gehört, heißt $\mathcal{C}$-vollständig (bzgl. $\leq^p$)
- NPC bezeichnet die Klasse aller NP-vollständigen Sprachen

Lemma

- Aus $A \leq^p B$ folgt $A \leq B$
- Die Reduktionsrelation $\leq^p$ ist reflexiv und transitiv (s. Übungen)

Die Polynomialzeitreduktion

Satz

Die Klassen P und NP sind unter $\leq^p$ abgeschlossen

Beweis

- Sei $B \in \mathrm{P}$ und gelte $A \leq^p B$ mittels einer Funktion $f \in \mathrm{FP}$
- Seien M und T DTMs mit $L(M) = B$ und $T(x) = f(x)$
- Weiter seien p und q polynomielle Zeitschranken für M und T
- Betrachte die DTM M', die bei Eingabe x zuerst T simuliert, um $f(x)$ zu berechnen, und danach M bei Eingabe $f(x)$ simuliert. Dann gilt

$$x \in A \Leftrightarrow f(x) \in B \Leftrightarrow f(x) \in L(M) \Leftrightarrow x \in L(M')$$

- Also ist $L(M') = A$ und wegen

$$time_{M'}(x) \leq time_T(x) + time_M(f(x)) \leq q(|x|) + p(q(|x|))$$

ist M' polynomiell zeitbeschränkt und somit A in P

- Die Abgeschlossenheit von NP unter $\leq^p$ folgt analog □

NP-Vollständigkeit

Satz

$A \leq^p B$ und A ist NP-hart $\Rightarrow$ B ist NP-hart

Beweis

- Sei $L \in \text{NP}$
- Da A NP-hart ist, gilt $L \leq^p A$
- Da zudem $A \leq^p B$ gilt und $\leq^p$ transitiv ist, folgt $L \leq^p B$

Satz

Falls ein NP-hartes Problem A in P enthalten ist, folgt P = NP

Beweis

- Sei $L \in \text{NP}$
- Da A NP-hart ist, gilt $L \leq^p A$
- Da $A \in \text{P}$ ist und P unter $\leq^p$ abgeschlossen ist, folgt $L \in \text{P}$ □

Platzkomplexität von Turingmaschinen

- Als nächstes definieren wir den Platzverbrauch von NTMs
- Intuitiv ist dies die Anzahl aller von einer NTM M besuchten Bandfelder
- Wollen wir auch sublinearen Platz sinnvoll definieren, so dürfen wir die Bandfelder, auf denen die Eingabe steht, nicht mitzählen
- Um sicherzustellen, dass M das erste Band nur zum Lesen der Eingabe und nicht als Speicherplatz benutzt, darf M
  - die Felder auf dem Eingabeband nicht verändern und
  - sich höchstens ein Feld von der Eingabe entfernen

Definition

Eine k-NTM M heißt NTM mit Eingabeband (kurz offline-NTM), falls für jede von M bei Eingabe x erreichbare Konfiguration

$$K = (q, u_1, a_1, v_1, \ldots, u_k, a_k, v_k)$$

gilt, dass $u_1 a_1 v_1$ ein Teilwort von $\sqcup x \sqcup$ ist

Platzkomplexität von Turingmaschinen

Definition

- Der Platzverbrauch einer offline-NTM M bei Eingabe x ist definiert als

$$space_M(x) = \sup \left\{ s \geq 0 \;\middle|\; \begin{array}{l} \exists K = (q, u_1, a_1, v_1, \ldots, u_k, a_k, v_k) \\ \text{mit } K_x \vdash^* K \text{ und } s = \sum_{i=2}^{k} |u_i a_i v_i| \end{array} \right\}$$

- Sei $s : \mathbb{N} \to \mathbb{N}$ eine monoton wachsende Funktion
- M heißt $s(n)$-platzbeschränkt, falls für alle Eingaben x gilt:

$$space_M(x) \leq s(|x|) \text{ und } time_M(x) < \infty$$

Platzkomplexitätsklassen

Wir fassen alle Sprachen, die in einer vorgegebenen Platzschranke $s(n)$ entscheidbar sind, in folgenden Platzkomplexitätsklassen zusammen

Definition

- Die auf deterministischem Platz $s(n)$ entscheidbaren Sprachen bilden die Klasse

$$\text{DSPACE}(s(n)) = \{L(M) \mid M \text{ ist eine } s(n)\text{-platzb. offline-DTM}\}$$

- Die auf nichtdeterministischem Platz $s(n)$ entscheidbaren Sprachen bilden die Klasse

$$\text{NSPACE}(s(n)) = \{L(M) \mid M \text{ ist eine } s(n)\text{-platzb. offline-NTM}\}$$

Die wichtigsten Platzkomplexitätsklassen

- Die wichtigsten deterministischen Platzkomplexitätsklassen sind

$$\begin{aligned} \mathsf{L} &= \mathrm{DSPACE}(\mathcal{O}(\log n)) && \text{„Logarithmischer Platz“} \\ \mathsf{LINSPACE} &= \mathrm{DSPACE}(\mathcal{O}(n)) && \text{„Linearer Platz“} \\ \mathsf{PSPACE} &= \mathrm{DSPACE}(n^{\mathcal{O}(1)}) && \text{„Polynomieller Platz“} \\ \mathsf{EXPSPACE} &= \mathrm{DSPACE}(2^{n^{\mathcal{O}(1)}}) && \text{„Exponentieller Platz“} \end{aligned}$$

- Die nichtdeterministischen Klassen NL, NLINSPACE, NPSPACE und NEXPSPACE sind analog definiert

Elementare Beziehungen zwischen Komplexitätsklassen

Frage

Welche elementaren Beziehungen gelten zwischen den verschiedenen Zeit- und Platzklassen?

Satz

- Für jede Funktion $t(n) \geq n+2$ gilt

  $$\mathrm{DTIME}(t) \subseteq \mathrm{NTIME}(t) \subseteq \mathrm{DSPACE}(\mathcal{O}(t))$$

- Für jede Funktion $s(n) \geq \log n$ gilt

  $$\mathrm{DSPACE}(s) \subseteq \mathrm{NSPACE}(s) \subseteq \mathrm{DTIME}(2^{\mathcal{O}(s)}) \text{ und}$$

  $$\mathrm{NSPACE}(s) \subseteq \mathrm{DSPACE}(s^2) \qquad \text{(Satz von Savitch)}$$

Korollar

- $\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{NPSPACE} \subseteq \mathrm{EXP} \subseteq \mathrm{NEXP} \subseteq \mathrm{EXPSPACE}$
- $\mathrm{PSPACE} = \mathrm{NPSPACE}$ und $\mathrm{EXPSPACE} = \mathrm{NEXPSPACE}$

Komplexität der Stufen der Chomsky-Hierarchie

- $\mathrm{REG} = \mathrm{DSPACE}(\mathcal{O}(1)) = \mathrm{NSPACE}(\mathcal{O}(1)) \subsetneq \mathrm{L}$
- $\mathrm{DCFL} \subsetneq \mathrm{LINTIME}$
- $\mathrm{CFL} \subsetneq \mathrm{NLINTIME} \cap \mathrm{DTIME}(\mathcal{O}(n^3)) \subsetneq \mathrm{P}$
- $\mathrm{DCSL} = \mathrm{LINSPACE} \subseteq \mathrm{CSL}$
- $\mathrm{CSL} = \mathrm{NLINSPACE} \subseteq \mathrm{PSPACE} \cap \mathrm{E}$
- $\mathrm{REC} = \bigcup_f \mathrm{DTIME}(f(n)) = \bigcup_f \mathrm{DSPACE}(f(n))$
  $= \bigcup_f \mathrm{NTIME}(f(n)) = \bigcup_f \mathrm{NSPACE}(f(n))$,
  wobei f alle (oder äquivalent: alle berechenbaren) Funktionen $f : \mathbb{N} \to \mathbb{N}$ durchläuft