Einführung in die Theoretische Informatik

Johannes Köbler

Institut für Informatik
Humboldt-Universität zu Berlin

WS 2016/17

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

$$
\begin{aligned}
\text{LINTIME} &= \bigcup_{c\geq 1} \text{DTIME}(cn+c) && \text{„Linearzeit“}\\
\text{P} &= \bigcup_{c\geq 1} \text{DTIME}(n^c+c) && \text{„Polynomialzeit“}\\
\text{E} &= \bigcup_{c\geq 1} \text{DTIME}(2^{cn+c}) && \text{„Lineare Exponentialzeit“}\\
\text{EXP} &= \bigcup_{c\geq 1} \text{DTIME}(2^{n^c+c}) && \text{„Exponentialzeit“}
\end{aligned}
$$

- 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 DTIME($\mathcal{F}$) $= \bigcup_{t\in\mathcal{F}} \text{DTIME}(t(n))$ (die Klassen NTIME($\mathcal{F}$) und 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

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

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:

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

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.
- Da jedoch nur Probleme in P als effizient lösbar angesehen werden, hat das so genannte P-NP-Problem, also die Frage, ob alle NP-Probleme effizient lösbar sind, 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 $B \in \mathsf{P}$ und gelte $A \leq^p B$ mittels einer Funktion $f \in \mathsf{FP}$.
- 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. $\square$

Satz

1. $A \leq^p B$ und A ist NP-hart $\Rightarrow$ B ist NP-hart.
2. $A \leq^p B$, A ist NP-hart und $B \in \mathrm{NP}$ $\Rightarrow$ $B \in \mathrm{NPC}$.
3. $\mathrm{NPC} \cap \mathrm{P} \neq \varnothing$ $\Rightarrow$ $\mathrm{P} = \mathrm{NP}$.

Beweis

1. Da A NP-hart ist, ist jede NP-Sprache L auf A reduzierbar.
   Da zudem $A \leq^p B$ gilt und $\leq^p$ transitiv ist, folgt $L \leq^p B$.
2. Klar, da B mit (1) NP-hart und nach Voraussetzung in NP ist.
3. Sei B eine NP-vollständige Sprache in P. Dann ist jede NP-Sprache A auf B reduzierbar und da P unter $\leq^p$ abgeschlossen ist, folgt $A \in \mathrm{P}$. $\square$

Platzkomplexität von Turingmaschinen

- Als nächstes definieren wir den Platzverbrauch von NTMs.
- Intuitiv ist dies die Anzahl aller besuchten Bandfelder.
- Wollen wir auch sublinearen Platz sinnvoll definieren, so dürfen wir hierbei das erste Band offensichtlich nicht berücksichtigen.
- Um sicherzustellen, dass eine NTM M das erste Band nur zum Lesen der Eingabe und nicht auch zum Speichern von weiteren Informationen benutzt, verlangen wir, dass M
  - die Felder auf dem Eingabeband nicht verändert und
  - sich höchstens ein Feld von der Eingabe entfernt.

Definition

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

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

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

Definition

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

$$space_M(x) = \sup \left\{ s \geq 1 \,\middle|\, \begin{array}{l} \exists K = (q, u_1, a_1, v_1, \ldots, u_k, a_k, v_k) \\ \text{mit } K_x \vdash^* K \text{ und } s = \sum_{i=2}^{k} |u_i a_i v_i| \end{array} \right\}.$$

- 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

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

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

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

Die wichtigsten Platzkomplexitätsklassen

- Die wichtigsten deterministischen Platzkomplexitätsklassen sind

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

- Die nichtdeterministischen Klassen NL, NLINSPACE und NPSPACE 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

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

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

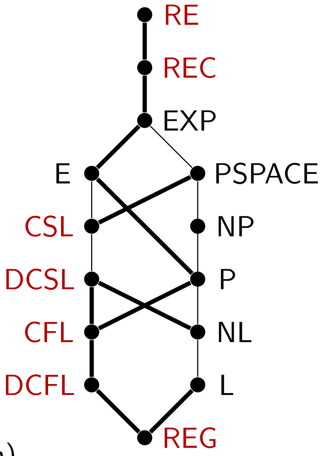
Korollar

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

Komplexität der Stufen der Chomsky-Hierarchie

$$
\begin{aligned}
\mathsf{REG} &= \mathsf{DSPACE}(\mathcal{O}(1)) = \mathsf{NSPACE}(\mathcal{O}(1)) \subsetneq \mathsf{L},\\
\mathsf{DCFL} &\subsetneq \mathsf{LINTIME},\\
\mathsf{CFL} &\subsetneq \mathsf{NLINTIME} \cap \mathsf{DTIME}(\mathcal{O}(n^3)) \subsetneq \mathsf{P},\\
\mathsf{DCSL} &= \mathsf{LINSPACE} \subseteq \mathsf{CSL},\\
\mathsf{CSL} &= \mathsf{NLINSPACE} \subseteq \mathsf{PSPACE} \cap \mathsf{E},\\
\mathsf{REC} &= \textstyle\bigcup_f \mathsf{DSPACE}(f(n))\\
&= \textstyle\bigcup_f \mathsf{NSPACE}(f(n))\\
&= \textstyle\bigcup_f \mathsf{DTIME}(f(n))\\
&= \textstyle\bigcup_f \mathsf{NTIME}(f(n)),
\end{aligned}
$$

wobei f alle (oder äquivalent: alle berechenbaren) Funktionen $f : \mathbb{N} \to \mathbb{N}$ durchläuft.



Aussagenlogische Formeln

- Die Menge der booleschen (oder aussagenlogischen) Formeln über den Variablen $x_1, \dots, x_n$, $n \geq 0$, ist induktiv wie folgt definiert:
  - Die Konstanten 0 und 1 sind boolesche Formeln.
  - Jede Variable x_i ist eine boolesche Formel.
  - Mit G und H sind auch die Konjunktion $(G \wedge H)$ und die Disjunktion $(G \vee H)$ von G und H sowie die Negation $\neg G$ von G Formeln.
- Eine Belegung von $x_1, \dots, x_n$ ist ein Wort $a = a_1 \dots a_n \in \{0,1\}^n$.
- Der Wert $F(a)$ von F unter a ist induktiv wie folgt definiert:

F	0	1	x_i	$\neg G$	$(G \wedge H)$	$(G \vee H)$
$F(a)$	0	1	a_i	$1 - G(a)$	$G(a)H(a)$	$G(a) + H(a) - G(a)H(a)$

- Durch die Formel F wird also eine n-stellige boolesche Funktion $F : \{0,1\}^n \to \{0,1\}$ definiert, die wir ebenfalls mit F bezeichnen.

Aussagenlogische Formeln

Notation

Wir benutzen die Implikation $G \to H$ als Abkürzung für die Formel $\neg G \vee H$ und die Äquivalenz $G \leftrightarrow H$ als Abkürzung für $(G \to H) \wedge (H \to G)$.

Beispiel (Wahrheitswerttabelle)

Die Formel $F = (G \to H)$ mit $G = (\neg x_1 \vee \neg x_2)$ und $H = (x_2 \wedge x_3)$ berechnet folgende boolesche Funktion $F : \{0,1\}^3 \to \{0,1\}$:

a	$G(a)$	$H(a)$	$F(a)$
000	1	0	0
001	1	0	0
010	1	0	0
011	1	1	1
100	1	0	0
101	1	0	0
110	0	0	1
111	0	1	1

◁

Aussagenlogische Formeln

Definition

- Zwei Formeln F und G heißen (logisch) äquivalent (kurz $F \equiv G$), wenn sie dieselbe boolesche Funktion berechnen.
- Eine Formel F heißt erfüllbar, falls es eine Belegung a mit $F(a) = 1$ gibt.
- Gilt sogar für alle Belegungen a, dass $F(a) = 1$ ist, so heißt F Tautologie.

Beispiel

- Die Formel $F = (G \to H)$ mit $G = (\neg x_1 \vee \neg x_2)$ und $H = (x_2 \wedge x_3)$ ist erfüllbar, da $F(111) = 1$ ist.
- F ist aber keine Tautologie, da $F(000) = 0$ ist. $\triangleleft$