Theoretische Informatik HS23

Nicolas Wehrli

Übungsstunde 10
2. Dezember 2023

ETH Zürich
nwehrl@ethz.ch

Heute

1. Feedback zur Serie
2. Catchup - Übriges von letzer Woche
3. Komplexitätstheorie

Feedback zur Serie

Feedback

1. Die Aussage
$$L \in \mathcal{L}_{\mathrm{RE}} \wedge L^{\complement} \in \mathcal{L}_{\mathrm{RE}} \implies L \in \mathcal{L}_{\mathrm{R}}$$
könnt ihr ab jetzt ohne Beweis nutzen, indem ihr auf dieses Aufgabenblatt verweist.
2. vorgeschaltete Turingmaschine gibt sich selbst weiter....
3. Bei der Kodierung von Turingmaschinen, dürft ihr nicht die zu beweisende Aussage (oder ähnliches) annehmen!
Beispiel:
Die TM M' akzeptiert, genau dann wenn die Berechnung von M auf w unendlich läuft.
Hier wurde angenommen, dass das Halteproblem L_{H} in $\mathcal{L}_{\mathrm{R}}$ ist!

Implikationsbeweis für Reduktion

Wenn eine Reduktion verlangt wird, dann dürft ihr die Implikation nicht trivial per Implikationsaussage zeigen.

Gemeint damit ist folgender Ansatz.

$L_1 \leq_R L_2$ soll gezeigt werden.

Da per Definition

$$L_1 \leq_R L_2 \iff (L_2 \in \mathcal{L}_R \implies L_1 \in \mathcal{L}_R)$$

folgt die gewünschte Aussage per $L_2 \notin \mathcal{L}_R$ (oder $L_1 \in \mathcal{L}_R$).

Dieser Ansatz gibt an der Prüfung 0 Punkte.

Neuer Trick für Reduktion von 24.b - Fortgeschritten

Aufgabe

Sei $L_{\text{all}} = \{\text{Kod}(M) \mid M \text{ akzeptiert jede Eingabe}\}$.

Zeigen Sie $L_{\text{H}}^{\complement} \leq_{\text{EE}} L_{\text{all}}$.

Kernidee

Für eine Eingabe $x = \text{Kod}(M)\#w$, generieren wir $\text{Kod}(A)$ einer TM A, die folgendes folgendes macht:

Neuer Trick für Reduktion von 24.b - Fortgeschritten

A :

Eingabe y

1. Berechnet $|y|$ Schritte von M auf w.
2. Falls danach die Berechnung nach $|y|$ noch nicht terminiert hat, akzeptiert A die Eingabe y.
3. Sonst verwirft A die Eingabe.

$$A \text{ akzeptiert jede Eingabe} \iff M \text{ läuft unendlich auf } w$$

Catchup - Übriges von letzer Woche

$L_1 \leq_{\mathbf{R}} L_2 \not\Longrightarrow (L_2 \in \mathcal{L}_{\mathbf{RE}} \implies L_1 \in \mathcal{L}_{\mathbf{RE}})$

Wir beweisen diese Aussage per Gegenbeispiel.

Sei $L_1 = L_{\text{diag}}$ und $L_2 = L_{\text{diag}}^{\complement}$.

Wir haben

- $L_1 = L_{\text{diag}} \notin \mathcal{L}_{\text{RE}}$ (Satz 5.5)
- $L_2 = L_{\text{diag}}^{\complement} \in \mathcal{L}_{\text{RE}} \setminus \mathcal{L}_{\text{R}}$ (Korollar 5.2, Lemma 5.5)

Per **Lemma 5.4** gilt $L_{\text{diag}} \leq_{\text{R}} L_{\text{diag}}^{\complement}$.

Die rechte Implikation gilt jedoch nicht.

■

$L_1 \leq_{\mathbf{R}} L_2 \not\Longleftarrow (L_2 \in \mathcal{L}_{\mathbf{RE}} \implies L_1 \in \mathcal{L}_{\mathbf{RE}})$

Sei $L_1 = L_{\mathrm{U}}$ und $L_2 = \{0^i \mid i \in \mathbb{N}\}$.

Wir haben

- $L_1 = L_{\mathrm{diag}} \in \mathcal{L}_{\mathrm{RE}} \setminus \mathcal{L}_{\mathrm{R}}$ (Satz 5.6 und 5.7)
- $L_2 = \{0^i \mid i \in \mathbb{N}\} \in \mathcal{L}_{\mathrm{R}}$ (da $\mathcal{L}_{\mathrm{EA}} \subset \mathcal{L}_{\mathrm{R}}$)

Da $L_1 \in \mathcal{L}_{\mathrm{RE}}$, gilt die Implikation auf der rechten Seite für dieses L_1 und L_2.

Da per Definition

$$L_1 \leq_{\mathrm{R}} L_2 \iff (L_2 \in \mathcal{L}_{\mathrm{R}} \implies L_1 \in \mathcal{L}_{\mathrm{R}})$$

folgt aus $L_1 \notin \mathcal{L}_{\mathrm{R}}$ und $L_2 \in \mathcal{L}_{\mathrm{R}}$, dass diese Instanzierung von L_1 und L_2 ein Gegenbeispiel ist.

■

Relation zu EE-Reduktion

Wir haben aber gezeigt, dass

$$L_1 \leq_{\text{EE}} L_2 \implies L_1 \leq_{\text{R}} L_2$$

und

$$L_1 \leq_{\text{EE}} L_2 \implies (L_2 \in \mathcal{L}_{\text{RE}} \implies L_1 \in \mathcal{L}_{\text{RE}})$$

Die Rückrichtung gilt jeweils nicht.

Komplexitätstheorie

Konfiguration

Wir erinneren uns:

Konfiguration einer k-Band-TM

Die Konfiguration einer k-Band-TM sieht wie folgt aus

$$(q, w, i, u_1, i_1, u_2, i_2, ..., u_k, i_k) \in Q \times \Sigma^* \times \mathbb{N} \times (\Gamma^* \times \mathbb{N})^k$$

wobei

- q der Zustand der TM ist
- $¢w\$$ der Inhalt des Eingabebandes, Lesekopf Eingabeband auf dem i-ten Feld
- für $j \in \{1, ..., k\}$ ist der Inhalt des j-ten Bandes $¢u_j␣␣␣$ und $i_j \leq |u_j|$ die Position des Kopfs auf dem j-ten Band.

Time

Sei M eine MTM oder TM, die immer hält. Sei Σ das Eingabealphabet von M.
Sei $x \in \Sigma^*$ und $D = C_1, C_2, ..., C_k$ die Berechnung von M auf x.
Die **Zeitkomplexität $\mathbf{Time_M(x)}$ der Berechnung von M auf x** ist definiert durch

$$\mathbf{Time_M(x)} = k - 1.$$

Die **Zeitkomplexität von M** ist die Funktion $\text{Time}_M : \mathbb{N} \to \mathbb{N}$, definiert durch

$$\mathbf{Time_M(n)} = \max \{\text{Time}_M(x) \mid x \in \Sigma^n\}.$$

Space

Sei $k \in \mathbb{N} \setminus \{0\}$. Sei M eine k-Band-TM, die immer hält.
Sei

$$C = (q, x, i, \alpha_1, i_1, \alpha_2, i_2, ..., \alpha_k, i_k)$$
$$\text{mit } 0 \leq i \leq |x| + 1 \text{ und } 0 \leq i_j \leq |\alpha_j| \text{ für } j = 1, ..., k$$

eine Konfiguration von M.
Die **Speicherplatzkomplexität von C** ist

$$\mathbf{Space_M(C)} = \max\{|\alpha_i| \mid i = 1, ..., k\}.$$

Space

Sei $C_1, C_2, ..., C_l$ die Berechnung von M auf x. Die **Speicherplatzkomplexität von M auf x** ist

$$\mathbf{Space_M}(\mathbf{x}) = \max\left\{\mathrm{Space}_M(C_i) \mid i = 1, ..., l\right\}.$$

Die **Speicherplatzkomplexität von M** ist die Funktion $\mathrm{Space}_M : \mathbb{N} \to \mathbb{N}$, definiert durch

$$\mathbf{Space_M}(\mathbf{n}) = \max\left\{\mathrm{Space}_M(x) \mid x \in \Sigma^n\right\}.$$

Space

Bemerkungen

1. Länge des Eingabewortes, hat keinen Einfluss auf die Speicherplatzkomplexität.
2. Mächtigkeit des Alphabets hat keinen Einfluss auf die Speicherplatzkomplexität.

Space

Lemma 6.1

Sei $k \in \mathbb{N} \setminus \{0\}$. Für jede k-Band-TM A, die immer hält, existiert eine äquivalente 1-Band-TM B, so dass

$$\text{Space}_B(n) \leq \text{Space}_A(n)$$

Beweisskizze:

Gleiche Konstruktion wie in Lemma 4.2.

Lemma 4.2 = "Für jede MTM A existiert eine äquivalente TM B".

Wir sehen, dass B genau so viele Felder braucht, wie A.

Space

Lemma 6.2

Zu jeder MTM A existiert eine äquivalente MTM B mit

$$\text{Space}_B(n) \leq \frac{\text{Space}_A(n)}{2} + 2$$

Beweisskizze:

Wir fassen jeweils 2 Felder von A zu einem Feld in B zusammen. $\Gamma_B = \Gamma_A \times \Gamma_A$. Wir addieren 1 für das ¢ am linken Rand und 1 für das Aufrunden im Fall von ungerader Länge.

Asymptotik

- $\mathcal{O}(\mathbf{f(n)})$:
  Menge aller Funktionen, die asymptotisch nicht schneller wachsen als $f(n)$.
- $\mathbf{\Omega(g(n))}$:
  Menge aller Funktionen, die asymptotisch mind. so schnell wachsen wie $g(n)$.
- $\mathbf{\Theta(h(n))}$:
  Menge aller Funktionen, die asymptotisch gleich schnell wachsen wie $h(n)$.

Small o-notation

Seien f und g zwei Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+$.
Falls $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, dann sagen wir, dass g asymptotisch **schneller wächst** als f:

$$f(n) \in o(g(n))$$

Bloomsches Speedup Theorem

Satz 6.1

Es **existiert** ein Entscheidungsproblem $(\Sigma_{\text{bool}}, L)$, so dass für jede MTM A, die $(\Sigma_{\text{bool}}, L)$ entscheidet, eine MTM B existiert, die auch $(\Sigma_{\text{bool}}, L)$ entscheidet, und für die gilt

$$\text{Time}_B(n) \leq \log_2(\text{Time}_A(n))$$

für unendlich viele $n \in \mathbb{N}$.

I.e. es existieren Entscheidungsprobleme, die keinen optimalen Algorithmus haben.

Deswegen fokussieren wir uns auf untere und obere Schranken der Komplexität eines Problemes und nicht auf die genaue Bestimmung davon.

Komplexität eines Entscheidungsproblems (Σ, L)

Sei L eine Sprache. Sei $f, g : \mathbb{N} \to \mathbb{R}^+$.

- $\mathcal{O}(g(n))$ ist eine **obere Schranke für die Zeitkomplexität von** L, falls eine MTM A **existiert**, die L entscheidet und $\text{Time}_A(n) \in \mathcal{O}(g(n))$.
- $\Omega(f(n))$ ist eine **untere Schranke für die Zeitkomplexität von** L, falls für **jede** MTM B die L entscheidet und $\text{Time}_B(n) \in \Omega(f(n))$.
- Eine MTM C heisst **optimal für** L, falls $\text{Time}_C(n) \in \mathcal{O}(f(n))$ und $\Omega(f(n))$ eine untere Schranke für die Zeitkomplexität ist.

Untere Schranke finden und beweisen: **schwierig**.

Obere Schranke kann durch einen konkreten Algorithmus gezeigt werden.

Klassen

Für alle Funktionen $f, g : \mathbb{N} \to \mathbb{R}^+$ definieren wir

$$\begin{aligned}
\mathbf{TIME}(\mathbf{f}) &= \{L(B) \mid B \text{ ist eine MTM mit } \mathrm{Time}_B(n) \in \mathcal{O}(f(n))\} \\
\mathbf{SPACE}(\mathbf{g}) &= \{L(A) \mid A \text{ ist eine MTM mit } \mathrm{Space}_A(n) \in \mathcal{O}(g(n))\} \\
\mathbf{DLOG} &= \mathrm{SPACE}(\log_2 n) \\
\mathbf{P} &= \bigcup_{c \in \mathbb{N}} \mathrm{TIME}(n^c) \\
\mathbf{PSPACE} &= \bigcup_{c \in \mathbb{N}} \mathrm{SPACE}(n^c) \\
\mathbf{EXPTIME} &= \bigcup_{d \in \mathbb{N}} \mathrm{TIME}(2^{n^d})
\end{aligned}$$

Zeitkomplexität zu Platzkomplexität

Lemma 6.3

Für jede Funktion $t : \mathbb{N} \to \mathbb{R}^+$ gilt

$$\text{TIME}(t(n)) \subseteq \text{SPACE}(t(n))$$

Beweisskizze:

In $\mathcal{O}(t(n))$ Schritten sind höchstens $\mathcal{O}(t(n))$ Felder beschreibbar.

Korollar 6.1

$$\text{P} \subseteq \text{PSPACE}$$

Platzkonstruierbarkeit

EIne Funktion: $s : \mathbb{N} \to \mathbb{N}$ heisst **platzkonstruierbar**, falls eine 1-Band-TM M existiert, so dass

(i) $\text{Space}_M(n) \leq s(n)$ für alle $n \in \mathbb{N}$ und

(ii) für jede Eingabe 0^n, generiert M das Wort $0^{s(n)}$ auf ihrem Arbeitsband und hält in q_{accept}.

Zeitkonstruierbarkeit

EIne Funktion: $t : \mathbb{N} \to \mathbb{N}$ heisst **zeitkonstruierbar**, falls eine MTM A existiert, so dass

(i) $\text{Time}_A(n) \leq t(n)$ für alle $n \in \mathbb{N}$ und

(ii) für jede Eingabe 0^n, generiert A das Wort $0^{t(n)}$ auf dem ersten Arbeitsband und hält in q_{accept}.

Platzgarantien

Lemma 6.4 (verständlicher formuliert)

Sei $s : \mathbb{N} \to \mathbb{N}$ platzkonstruierbar.
Für jede MTM M, für welche $\text{Space}_M(w) \leq s(|w|)$ nur für alle $w \in L(M)$ erfüllt, existiert eine äquivalente MTM A, welche dies für alle $w \in \Sigma^*$ erfüllt.

Beweisskizze:

Erzeuge für jede Eingabe $x \in \Sigma^*$ zuerst $0^{s(|x|)}$ auf einem zusätzlichen Band und nutze das als Platzüberwachung.

Wenn A diesen Platz überschreiten will, wird die Simulation unterbrochen und die Eingabe verworfen.

Lemma 6.5 (verständlicher formuliert)

Sei $t : \mathbb{N} \to \mathbb{N}$ zeitkonstruierbar.
Zu jeder MTM M, welche $\text{Time}_M(w) \leq t(|w|)$ nur für alle $w \in L(M)$ erfüllt, existiert eine äquivalente MTM A, welche zumindest $\text{Time}_A(w) \leq 2t(|w|) \in \mathcal{O}(t(|w|))$ für alle $w \in \Sigma^*$ erfüllt.

$$\implies \text{Time}_A(n) \in \mathcal{O}(t(n))$$

Beweisskizze:

Schreibe für jede Eingabe $x \in \Sigma^*$ $0^{t(|x|)}$ auf ein zusätzliches Arbeitsband und nutze dies zur Zeitzählung.

Wenn A mehr Schritte machen will, wird die Simulation abgebrochen und die Eingabe verworfen.

Satz 6.2

Für jede Funktion s mit $s(n) \geq \log_2(n)$ gilt:

$$\mathbf{SPACE}(s(n)) \subseteq \bigcup_{c \in \mathbb{N}} \mathbf{TIME}(c^{s(n)})$$

Beweis

Sei $L \in \mathbf{SPACE}(s(n))$. Nach Lemma 6.1 existiert eine 1-Band-TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, die **immer hält**, so dass $L = L(M)$ und $\text{Space}_M(n) \leq d \cdot s(n)$ für $d \in \mathbb{N}$ gelten.

Speicherplatzkomplexität zu Zeitkomplexität

Für jede Konfiguration $C = (q, w, i, x, j)$ von M definieren wir die **innere Konfiguration von** C als

$$\mathrm{In}(C) = (q, i, x, j).$$

Die innere Konfiguration enthält das Eingabewort w nicht, da dies sich während einer Berechnung nicht ändert.

Speicherplatzkomplexität zu Zeitkomplexität

Sei $\text{InKonf}_M(n)$ die Menge aller möglichen inneren Konfigurationen auf Eingabewörtern der Länge n.

Sei $X = |\text{InKonf}_M(n)|$ dessen Kardinalität.

Sei $D = C_1C_2...C_k$ eine endliche Berechnung von M auf einem Wort $w, |w| = n$.

Wir zeigen per Widerspruch, dass D maximal X verschiedene Konfigurationen haben kann, i.e. $k \leq X$.

Speicherplatzkomplexität zu Zeitkomplexität

Nehmen wir zum Widerspruch an $k > X$.

Dann muss es in $D = C_1C_2...C_i...C_j...C_k$, zwei identische innere Konfigurationen $\text{In}(C_i)$ und $\text{In}(C_j)$ geben (für $i < j$).

Da M deterministisch ist, sollte aber von $C_i = C_j$ aus immer die gleichen Berechnungsschritte ausgeführt werden.

Dann wäre aber D eine unendliche Berechnung mit der Endlosschleife $C_iC_{i+1}...C_j$. **Widerspruch**, da M immer hält.

Speicherplatzkomplexität zu Zeitkomplexität

Eine beliebige endliche Berechnung D von M auf w, $|w| = n$, kann höchstens X viele Zeitschritte (i.e. Konfigurationen) haben.

Jetzt müssen wir noch $X = |\text{InKonf}_M(n)$ abschätzen.

Wir wissen folgendes

- Es gibt $|Q|$ verschieden mögliche Zustände.
- Index des Eingabekopfes ist $0 \leq i \leq n + 1$ (Eingabeband ¢w\$ mit $|w| = n$)
- Inhalt des Arbeitsbandes x hat Länge: $|x| \leq \text{Space}_M(n) \leq d^{s}(n)$
- Index vom Kopf auf dem Arbeitsband: $0 \leq j \leq \text{Space}_M(n) \leq d \cdot s(n)$
- $x \in \Gamma^{|x|}$
- $n + 2 \leq 4^{\log_2 n} \leq 4^{s(n)}$ für $n \geq 2$

Speicherplatzkomplexität zu Zeitkomplexität

Setzen wir alles zusammen:

$$\begin{aligned}|\text{InKonf}_M(n)| &\leq |Q| \cdot (n+2) \cdot |\Gamma|^{\text{Space}_M(n)} \cdot \text{Space}_M(n) \\ &\leq (\max\{4, |Q|, |\Gamma|\})^{4d \cdot s(n)} \\ &\leq c^{s(n)}\end{aligned}$$

■