Karlsruher Institut für Technologie

Maximilian Katzmann
Hans-Peter Lehmann
Stefan Walzer

Übungsblatt 08

Randomisierte Algorithmik – Wintersemester 2023/2024

Aufgaben zur Vorlesung vom 14.12.2023
Abgabe im ILIAS bis 21.12.2023, 11:30 Uhr
Besprechung am 16.01.2024, 08:00 Uhr
Achte insbesondere bei handschriftlichen Abgaben auf Lesbarkeit. Die Abgabe erfolgt über das Übungsmodul im ILIAS. Gib Deine Ausarbeitungen in *einer* PDF-Datei ab.

Aufgabe 1 – Beziehungen zwischen Komplexitätsklassen

Begründe folgende Inklusionen:

(i) $\mathbf{ZPP} \subseteq \mathbf{RP}$ und $\mathbf{ZPP} \subseteq \text{co}-\mathbf{RP}$

(ii) $\mathbf{P} \subseteq \mathbf{ZPP}$

(iii) $\mathbf{RP} \subseteq \mathbf{NP}$ und $\text{co}-\mathbf{RP} \subseteq \text{co}-\mathbf{NP}$

(iv) $\mathbf{RP} \subseteq \mathbf{BPP}$ und $\text{co}-\mathbf{RP} \subseteq \mathbf{BPP}$

(v) $\mathbf{BPP} \subseteq \mathbf{PP}$

Lösung 1

(i) Gilt nach Definition von $\mathbf{ZPP} := \mathbf{RP} \cap \text{co}-\mathbf{RP}$.

(ii) Sei $L \in \mathbf{P}$. Zu zeigen ist $L \in \mathbf{ZPP}$. Sei T eine $\mathbf{P}$-DTM für L. Wir nutzen das „type-casting“ der Vorlesung. Aus T wird somit formal eine RTM (die nicht mehr wirklich randomisiert ist), die immernoch L entscheidet und immernoch polynomielle Laufzeit hat. Insbesondere bezeugt diese RTM, dass $L \in \mathbf{RP}$ ist. Weil $\mathbf{P} = \text{co}-\mathbf{P}$ gilt, ist $\bar{L} \in \mathbf{P}$ und nach dem Argument von eben daher auch $\bar{L} \in \mathbf{RP}$. Also gilt $L \in \text{co}-\mathbf{RP}$. Zusammen ergibt sich $L \in \mathbf{RP} \cap \text{co}-\mathbf{RP} = \mathbf{ZPP}$.

(iii) Sei $L \in \mathbf{RP}$ und T eine $\mathbf{RP}$-PTM für L. Durch „Vergessen“ der Wahrscheinlichkeiten entsteht eine NTM T'. Weil für jedes $w \in L$ galt, dass $\Pr[T(w) = \text{YES}] \geq \frac{1}{2}$, existiert insbesondere eine akzeptierende Berechnung für w. Also gilt $T'(w) = \text{YES}$. Für jedes $w \notin L$ galt, dass $\Pr[T(w) = \text{YES}] = 0$. Also existiert keine akzeptierende Berechnung

für w. Also gilt $T'(w) = \text{NO}$. Also wird L von T' entschieden. Also gilt $L \in \mathbf{NP}$. Weil L beliebig war folgt $\mathbf{RP} \subseteq \mathbf{NP}$.

Für den symmetrischen Fall können wir nun auch schlussfolgern:

$$L \in \text{co}-\mathbf{RP} \Leftrightarrow \bar{L} \in \mathbf{RP} \Rightarrow \bar{L} \in \mathbf{NP} \Leftrightarrow L \in \text{co}-\mathbf{NP}.$$

(iv) Sei $L \in \mathbf{RP}$ und T eine **RP**-PTM für L. Wenn T' die PTM ist, die T dreimal hintereinander ausführt (mit unabhängigem Zufall) und akzeptiert, wenn mindestens eine der Berechnungen von T akzeptiert, dann gilt für alle Wörter $w \in L$:

$$\Pr[T'(w) = \text{NO}] = \Pr[T(w) = \text{NO}]^3.$$

Für $w \in L$ ergibt sich:

$$\begin{aligned}\Pr[T'(w) = \text{YES}] &= 1 - \Pr[T'(w) = \text{NO}] = 1 - \Pr[T(w) = \text{NO}]^3 \\ &= 1 - (1 - \Pr[T(w) = \text{YES}])^3 \geq 1 - \left(1 - \frac{1}{2}\right)^3 = 1 - \frac{1}{8} > \frac{3}{4}.\end{aligned}$$

Für $w \notin L$ ergibt sich:

$$\Pr[T'(w) = \text{YES}] = 1 - \Pr[T'(w) = \text{NO}] = 1 - \Pr[T(w) = \text{NO}]^3 = 1 - 1^3 = 0 < \frac{1}{4}.$$

Also ist T' eine **BPP**-PTM für L. Somit gilt $L \in \mathbf{BPP}$. Weil L beliebig war folgt $\mathbf{RP} \subseteq \mathbf{BPP}$.

Der symmetrisch Fall ist wieder analog weil $\mathbf{BPP} = \text{co}-\mathbf{BPP}$ gilt.

(v) Hier ist überhaupt nichts zu tun. Jede **BPP**-PTM ist auch eine **PP**-PTM, weil die Anforderungen lediglich heruntergeschraubt werden. Also gibt es für jede Sprache L, für die es eine **BPP**-PTM gibt, auch eine **PP**-PTM.

Aufgabe 2 – Las Vegas Algorithmus für L impliziert $L \in$ ZPP

Sei **LV** (für Las Vegas) die Klasse aller Sprachen L, für die eine probabilistische Turingmaschine T mit folgenden Eigenschaften existiert:

- T entscheidet L. (Das heißt L liefert für alle $x \in L$ stets die Ausgabe 1 und für alle $x \notin L$ stets die Ausgabe 0.)
- Es gibt ein Polynom $p(n)$, sodass die *erwartete* Laufzeit von T bei Eingabe x durch $p(|x|)$ beschränkt ist.

In der Vorlesung haben wir bewiesen, dass $\mathbf{ZPP} \subseteq \mathbf{LV}$ gilt. Zeige nun, dass auch $\mathbf{LV} \subseteq \mathbf{ZPP}$ gilt. die beiden Klassen sind also identisch.

Nützliche Zutaten: Definition von **ZPP**, Markov-Ungleichung.

Lösung 2

Sei $L \in \mathbf{LV}$. Wir zeigen zunächst $L \in \mathbf{RP}$. Sei nun also T eine **LV**-TM für L mit zugehörigem Polynom $p(n)$. Wir betrachten folgende Turingmaschine T':

Algorithm $T'(w)$**:**
  $t_{\max} \leftarrow 2p(|w|)$
  simuliere T auf Eingabe w für $t_{\max}$ Schritte
  if *T hat mit Ausgabe r terminiert* **then**
    return r
  else // T hat noch nicht terminiert
    return NO

Wir zeigen nun, dass T' eine **RP**-TM für L ist. Durch die Verwendung von $t_{\max}$ ist klar, dass sich die Laufzeit von $T'(w)$ durch ein Polynom $q(|w|)$ beschränken lässt. Zu den Ausgaben von T' is zu sagen:

- Für $w \notin L$ ist die Ausgabe von $T'(w)$ stets NO, entweder, weil T das so entschieden hat oder weil wir im else-Fall angekommen sind. Also: $\Pr[T'(w) = \text{YES}] = 0$.
- Für $w \in L$ betrachten wir die zufällige Laufzeit $t(w)$ von T auf w. Nach Vorraussetzung gilt $\mathbb{E}[t(w)] \leq p(|w|)$. Mit der Markov Ungleichung folgt:

$$\Pr[t(w) > t_{\max}] \leq \frac{\mathbb{E}[t(w)]}{t_{\max}} \leq \frac{p(|w|)}{2p(|w|)} \leq \frac{1}{2}.$$

  Also terminiert T auf w mit Wahrscheinlichkeit mindestens $1/2$ innerhalb des gesetzten Zeitlimits (mit dem richtigen Ergebnis). Also gilt $\Pr[T'(w) = \text{YES}] \geq \frac{1}{2}$.

Also ist T' eine **RP**-TM für L. Also gilt $L \in \mathbf{RP}$. Analog folgt auch $L \in \text{co}-\mathbf{RP}$ (indem man die Ausgabe bei nicht-Terminierung auf YES setzt) und somit $L \in \mathbf{ZPP}$. Weil L beliebig war gilt $\mathbf{LV} \subseteq \mathbf{ZPP}$.

Aufgabe 3 – Bonus: Probability Amplification für BPP

Sei T eine **BPP**-PTM für eine Sprache $L \subseteq \{0,1\}^*$. Betrachte die folgende PTM T':

Algorithm $T'(w, k)$**:**
  $a \leftarrow 0$
  for $i \leftarrow 1$ **to** k **do**
    $r \leftarrow T(w)$
    if $r =$ *YES* **then**
      $a \leftarrow a + 1$
  if $a > k/2$ **then**
    return YES
  else
    return NO

(i) Zeige (z.B. mit Chernoff Schranken), dass die Wahrscheinlichkeit, dass T' inkorrekt antwortet, höchstens $\exp(-\Omega(k))$ ist.

(ii) Nehmen wir an, wir betrachten nur Eingaben einer festen Länge $n \in \mathbb{N}$. Wie muss man k in Abhängigkeit von n wählen, sodass die Wahrscheinlichkeit, dass T' für eine gegebene Eingabe inkorrekt antwortet, weniger als 2^{-n} beträgt? Argumentiere, dass es eine Wahl für die Zufallsbits von T' gibt, sodass T' auf *allen* Eingaben der Länge n korrekt antwortet.

(iii) Recherchiere auf Wikipedia, was es mit der Komplexitätsklasse **P/poly** auf sich hat. Zeige, dass **BPP** $\subseteq$ **P/poly** gilt. Diese Einsicht heißt auch Adlemans Satz.

Lösung 3

(i) Sei zunächst $w \notin L$. Dann gilt für $p := \Pr[T(w) = \text{YES}] < \frac{1}{4}$. Für die Variable a gilt am Ende $a \sim \text{Bin}(k, p)$. Die Antwort von T' ist falsch wenn $a > k/2$ gilt. Die Fehlerwahrscheinlichkeit ist also höchstens

$$\Pr_{a \sim \text{Bin}(k,p)}[a > k/2].$$

Wir verwenden nun eine Chernoff-Schranke von Übungsblatt 4 Aufgabe 1 (a). Wir wissen, dass $\mathbb{E}[a] = pk \leq k/4 =: f$, also f eine obere Schranke an den Erwartungswert von a ist. Damit gilt:

$$\Pr_{a \sim \text{Bin}(k,p)}[a > k/2] \leq \Pr_{a \sim \text{Bin}(k,p)}[a \geq 2f] \leq \exp(-2^2 f/3) = \exp(-k/3).$$

Für $w \in L$ kann man analog argumentieren. Man betrachtet hier die Zufallsvariable $(k - a) \sim \text{Bin}(k, p')$ wobei $p' = \Pr[T(w) = \text{NO}] < \frac{1}{4}$. Diese Differenz ist wie eben die Anzahl der Male, an denen T sich irrt.

(ii) Wähle $k = 3\ln(2) \cdot n + 1$. Dann ist die Fehlerwahrscheinlichkeit von T' kleiner als

$$\exp(-3\ln(2)n/3) = 2^{-n}.$$

Im Folgenden argumentieren wir etwas formaler als verlangt, damit (iii) leichter wird. Wir betrachten die deterministische Turingmaschine T'' mit Eingaben $w \in \{0,1\}^n$ und $b \in \{0,1\}^{k \cdot p(n)}$, wobei $p(n)$ eine Schranke für die Laufzeit von T ist.

Die Aufgabe von T'' ist es, T' mit Eingabe w (und k wie oben festgelegt) zu similieren, dabei aber alle Zufallsentscheidungen gemäß b zu machen.

Sei $\text{fail}(b)$ die Anzahl von Wörtern w, für die T'' die falsche Ausgabe liefert. Wenn b uniform zufällig gewählt wird, dann gilt nach Linearität des Erwartungswertes und der Fehlerwahrscheinlichkeit von T'

$$\mathbb{E}_{b \sim \{0,1\}^{kp(n)}}[\text{fail}(b)] = \sum_{w \in \{0,1\}^n} \Pr_{b \sim \{0,1\}^{kp(n)}}[T''(w,b) \text{ falsch}] < \sum_{w \in \{0,1\}^n} 2^{-n} = 2^n \cdot 2^{-n} = 1.$$

Weil für nicht-negative Zufallsvariablen aus $\mathbb{E}[X] < 1$ folgt, dass $X = 0$ *möglich* ist (probabilistische Methode), gibt es eine Wahl b^* mit $\text{fail}(b^*) = 0$. Dies ist die gesuchte Folge von Bits.

(iii) Die Maschine T'' entscheidet L mit *advice* b^*. Freilich hängt b^* von n ab, weshalb auch nicht $L \in \mathbf{P}$ sondern nur $L \in \mathbf{P}/\mathbf{poly}$ folgt.