Kapitel 7: Randomisierte Berechnungen

Randomisierte Algorithmen =
Algorithmen, die im Lauf ihrer Berechnungen "Zufallsbits" verwenden dürfen

7.1 Ein probabilistischer Primzahltest

Theorem 7.1

Es gibt einen randomisierten Algorithmus A, der bei Eingabe einer natürlichen Zahl n nach poly(log n) Schritten anhält, und für dessen Ausgabe $A(n)$ gilt:

- falls n eine Primzahl ist, so ist $A(n) =$ "ja"
- falls n keine Primzahl ist, so gilt mit Wahrscheinlichkeit $\geq \frac{1}{2}$: $A(n) =$ "nein"

Zum Beweis von Theorem 7.1 benötigen wir einige Begriffe und Resultate aus der elementaren Zahlentheorie:

Definition 7.2 (Legendre-Symbol $(m|p)$ und Jacobi-Symbol $(m|n)$)

(a) Sei p eine Primzahl $\neq 2$ und sei $m \in \mathbb{N}$.
Das Legendre-Symbol $(m|p)$ ist die wie folgt definierte Zahl aus $\{0, 1, -1\}$:

$$(m|p) := \begin{cases} 0 & \text{falls } p \text{ ein Teiler von } m \text{ ist} \\ 1 & \text{falls es ein } w \in \mathbb{N} \text{ gibt, s.d. } m \equiv w^2 \pmod p \text{ ist} \\ -1 & \text{sonst} \end{cases}$$

Bemerkung: $(m|p) = 1$ bedeutet, dass m ein sog. quadratischer Rest modulo p ist

(b) Sei n eine ungerade natürliche Zahl > 1 und sei $p_1^{k_1} \cdots p_\ell^{k_\ell}$ die Primzahlzerlegung von n (d.h.: $p_1 < \cdots < p_\ell$ sind paarweise verschiedene Primzahlen, $k_1, \dots, k_\ell \geq 1$ und $n = p_1^{k_1} \cdots p_\ell^{k_\ell}$.
Sei $m \in \mathbb{N}$. Das Jacobi-Symbol $(m|n)$ ist die wie folgt definierte Zahl aus $\{0, 1, -1\}$:

$$(m|n) := (m \,|\, p_1^{k_1} \cdots p_\ell^{k_\ell}) := (m|p_1)^{k_1} \cdot \ldots \cdot (m|p_\ell)^{k_\ell}$$

Beispiel 7.3

$$(11\,|\,91) = (11\,|\,7 \cdot 13) = (11|7) \cdot (11|13) = (4|7) \cdot (11|13) = 1 \cdot (-1) = -1$$

Denn: • $4 \equiv 2^2 \pmod 7$, also $(4|7) = 1$
• quadratische Reste modulo 13 sind: $1^2 = \underline{\underline{1}}$, $2^2 = \underline{\underline{4}}$, $3^2 = \underline{\underline{9}}$, $4^2 = 16 \equiv \underline{\underline{3}} \pmod{13}$,

$5^2 = 25 \equiv \underline{\underline{12}} \pmod{13}$, $\quad 6^2 = 36 \equiv \underline{\underline{10}} \pmod{13}$, $\quad 7^2 \equiv (-6)^2 \equiv 6^2 \equiv \underline{\underline{10}} \pmod{13}$

$8^2 \equiv (-5)^2 \equiv 5^2 \equiv \underline{\underline{12}} \pmod{13}$, $\quad 9^2 \equiv (-4)^2 \equiv 4^2 \equiv \underline{\underline{3}} \pmod{13}$,

$10^2 \equiv (-3)^2 \equiv 3^2 \equiv \underline{\underline{9}} \pmod{13}$, $\quad 11^2 \equiv (-2)^2 \equiv 2^2 \equiv \underline{\underline{4}} \bmod (13)$, $\quad 12^2 \equiv (-1)^2 \equiv 1^2 \equiv \underline{\underline{1}} \pmod{1$

D.h.: Die Zahlen $1, 3, 4, 9, 10, 12$ sind sämtliche quadratischen Reste modulo 13. Insbes. ist $(11 \mid 13) = -1$.

Aus der Zahlentheorie ist folgender Satz bekannt:

<u>Satz 7.4</u>

Sei n eine ungerade natürliche Zahl > 1.

(<u>a</u>) Falls n eine Primzahl ist, so gilt für alle $m \in \{1, \ldots, n-1\}$: $\quad (m \mid n) \equiv m^{\frac{n-1}{2}} \pmod{n}$

(<u>b</u>) Falls n keine Primzahl ist, so gilt für mindestens die Hälfte aller Zahlen $m \in \{2, \ldots, n-1\}$ mit $ggT(m,n) = 1$, dass $(m \mid n) \not\equiv m^{\frac{n-1}{2}} \pmod{n}$.

<u>Bemerkung:</u> $ggT(m,n)$ bezeichnet hier den größten gemeinsamen Teiler von m und n, d.h. die größte natürliche Zahl g, so dass es natürliche Zahlen m' und n' gibt mit $m = gm'$ und $n = gn'$.

(Hier ohne Beweis).

Unter Verwendung von Satz 7.4 können wir nun den zum Beweis von Theorem 7.1 gesuchten randomisierten Primzahltest angeben:

Beweis von Theorem 7.1:

Sei A der folgende randomisierte Algorithmus:

Eingabe: eine natürliche Zahl n

Frage: Ist n eine Primzahl?

Algorithmus:

Falls $n=0$ oder $n=1$, so STOPP mit Ausgabe "nein".

Falls $n=2$, so STOPP mit Ausgabe "ja"

Falls n eine gerade Zahl ist, so STOPP mit Ausgabe "nein".

Sonst (d.h.: n ist eine ungerade nat. Zahl >1):

Wähle zufällig eine Zahl $m \in \{2, \dots, n-1\}$.

Berechne $g := ggT(m,n)$

Falls $g \neq 1$, so STOPP mit Ausgabe "nein"
(denn: g ist ein Teiler von n und n somit keine Primzahl)
$\neq n$

Sonst: Berechne $j := (m|n)$

Berechne $k := m^{\frac{n-1}{2}} \pmod n$

Falls $j \neq k$, so STOPP mit Ausgabe "nein"
(wegen Satz 7.4(a) ist n dann nämlich keine Primzahl)

Sonst: STOPP mit Ausgabe "ja".

Für diesen Algorithmus gilt:

- Falls n eine Primzahl ist, so gilt für jedes mögliche $m \in \{2, \dots, n-1\}$, dass $g = ggT(m,n) = 1$, und wegen Satz 7.4(a) gilt: $j = (m|n) \equiv m^{\frac{n-1}{2}} \equiv k \pmod n$.
  Daher ist – unabhängig von der konkreten Wahl von m – stets $A(n) =$ "ja".

- Falls n keine Primzahl ist, so gilt:
  - für alle $m \in \{2,\dots,n-1\}$ mit $g := ggT(m,n) \neq 1$ ist $A(n) =$ "nein"
  - für mindestens die Hälfte aller $m \in \{2,\dots,n-1\}$ mit $g = ggT(m,n) =$ ist $A(n) =$ "nein" (wegen Satz 7.4 (b))

Insgesamt ist also für mindestens die Hälfte aller zufälligen Wahlen von m das Ergebnis $A(n) =$ "nein" — d.h. es gilt mit Wahrscheinlichkeit $\geq \frac{1}{2}$, dass $A(n) =$ "nein" ist

Somit weist unser Algorithmus A das von Theorem 7.1 geforderte Verhalten auf.

Um die gewünschte Laufzeit von poly(log n) zu erhalten, benötigen wir Verfahren, die

(1) bei Eingabe der Binärdarstellung von n durch poly(log n) viele Münzwürfe die Binärdarstellung einer zufällig, gleichverteilt aus $\{2,\dots,n-1\}$ gewählten Zahl m erzeugen und dabei nur poly(log n) Berechnungsschritte machen

(2) bei Eingabe der Binärdarstellungen zweier Zahlen m und n in poly(log m + log n) Berechnungsschritten die Binärdarstellung von $ggT(m,n)$ erzeugen

(3) bei Eingabe der Binärdarstellungen zweier Zahlen m und n (mit $n > 1$, ungerade, und $m \in \{2,\dots,n-1\}$) in poly(log m + log n) Berechnungsschritten das Jacobi-Symbol $(m|n) \in \{0,1,-1\}$ erzeugen

(4) bei Eingabe der Binärdarstellungen zweier Zahlen m und n mit $m \in \{2,\dots,n-1\}$ die Zahl $k := m^{\frac{n-1}{2}} \pmod n$ erzeugen.

Zu (1): Übungsaufgabe.

Zu (2): Hierzu verwenden wir den bekannten Euklidischen Algorithmus zur ggT-Berechnung:

ggT(m,n):

Falls $m=0$, so STOPP mit Ausgabe n

Falls $n=0$, so STOPP mit Ausgabe m

Sonst:

Falls $n \geq m$, so:

(*) berechne $r \in \{0,\dots,m-1\}$ mit $n \equiv r \pmod m$

STOPP mit Ausgabe ggT(m,r)

Sonst:

(**) berechne $r \in \{0,\dots,n-1\}$ mit $m \equiv r \pmod n$

STOPP mit Ausgabe ggT(r,n)

Beispiel:

$$\text{ggT}(51,91) = \text{ggT}(51,40) = \text{ggT}(11,40) = \text{ggT}(11,7)$$
$$= \text{ggT}(4,7) = \text{ggT}(4,3) = \text{ggT}(1,3)$$
$$= \text{ggT}(1,0) = 1$$

Laufzeitanalyse: Man sieht leicht, dass bei (*) $r \leq \frac{n}{2}$ ist (denn: $n = k \cdot m + r$ mit $k \geq 1$, $0 \leq r < m$) und bei (**) $r \leq \frac{m}{2}$ ist.

Daher wird in jedem Rekursionsschritt die größere der beiden Zahlen halbiert und daher ist die rekursive Berechnung von ggT(m,n) nach $O(\log m + \log n)$ Rekursionsschritten beendet.

Die für (*) bzw (**) nötige Division mit Rest ($r := n \text{ MOD } m$ bzw $r := m \text{ MOD } n$) kann in Zeit poly(log m + log n) durchgeführt werden (Details: Übung!)

<u>zu (3)</u>: Zur Berechnung des Jacobi-Symbols $(m|n)$ können wir ähnlich wie bei (2) vorgehen und dabei folgende aus der Zahlentheorie bekannte Rechenregeln (hier ohne Beweis) benutzen:

F.a. m, n mit n ungerade, $n > 1$ und $\text{ggT}(m,n) = 1$ gilt:

(a) $(1|n) = 1$

(b) $(2|n) = (-1)^{\frac{n^2-1}{8}}$

(c) Falls m gerade, so $(m|n) = (2|n) \cdot (\frac{m}{2}|n)$

(d) Falls $m \geq n$ ist, so gilt für $r := m \bmod n$:

$$(m|n) = (r|n)$$

(e) Falls $m < n$ ist, so gilt:

Falls $8 \mid (m-1)\cdot(n-1)$, so $(m|n) = (n|m)$

Sonst: $(m|n) = -(n|m)$

(sog. "quadratisches Reziprozitätsgesetz")

Details zum Algorithmus <u>Jacobi(m,n)</u>, der in poly(log m + log n) Berechnungsschritten das Jacobi-Symbol $(m|n)$ berechnet: Übung!

<u>zu (4)</u>: Übung!

Insgesamt erhalten wir, dass Algorithmus A stets nach poly(log n) Berechnungsschritten anhält. □

7.2 Probabilistische Turingmaschinen und Komplexitätsklassen

Probabilistische Turingmaschine $\hat{=}$
eine TM, die in jedem Schritt ihrer Berechnung eine Münze werfen kann.

Definition 7.5 (PTM)

(a) Eine <u>probabilistische Turingmaschine (PTM)</u> M mit zwei Transitionsfunktionen δ_0 und δ_1.
Bei Eingabe eines Worts $x \in \{0,1\}^*$ wird jeder einzelne Berechnungsschritt mit Wahrscheinlichkeit $1/2$ gemäß δ_0 und mit Wahrscheinlichkeit $1/2$ gemäß δ_1 durchgeführt.
M gibt am Ende ihrer Berechnung stets 1 (für "akzeptieren") oder 0 (für "verwerfen") aus.
Wir schreiben <u>$M(x)$</u>, um die Zufallsvariable (mit Werten aus $\{0,1\}$) zu bezeichnen, die der Ausgabe von M bei Eingabe x entspricht.

(b) Sei $T: \mathbb{N} \to \mathbb{N}$.
Eine PTM M ist <u>$T(n)$-zeitbeschränkt</u>, falls für jedes $x \in \{0,1\}^*$ gilt: Unabhängig von den zufälligen Entscheidungen, die M trifft, hält M bei Eingabe x nach höchstens $T(|x|)$ Schritten an.

Komplexitätsklassen mit einseitigem, beschränktem Fehler

Definition 7.6 (RTIME$(T(n))$, RP, coRP)

(a) Sei $T: \mathbb{N} \to \mathbb{N}$.

Die Klasse RTIME$(T(n))$ besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, für die es eine $T(n)$-zeitbeschränkte PTM M gibt, so dass für alle Eingaben $x \in \{0,1\}^*$ gilt

- falls $x \in L$, so ist $Pr[M(x) = 1] \geq 2/3$
- falls $x \notin L$, so ist $Pr[M(x) = 0] = 1$

D.h.: Jedes $x \notin L$ wird von M stets verworfen; jedes $x \in L$ wird von M mit Wahrscheinlichkeit $\geq 2/3$ akzeptiert.

(b) $RP := \bigcup_{c \geq 1} \text{RTIME}(n^c)$ ist die Klasse aller randomisiert in Polynomialzeit lösbaren Probleme

(c) $coRP := \{ L \subseteq \{0,1\}^* : \overline{L} \in RP \}$

D.h.: $L \in coRP \iff$ es gibt eine in Polynomialzeit laufende PTM M, so dass für alle Eingaben $x \in \{0,1\}^*$ gilt:

- falls $x \in L$, so ist $Pr[M(x) = 1] = 1$
- falls $x \notin L$, so ist $Pr[M(x) = 0] \geq 2/3$

D.h.: Jedes $x \in L$ wird von M stets akzeptiert, jedes $x \notin L$ wird von M mit Wahrscheinlichkeit $\geq 2/3$ verworfen.

Lemma 7.7 (Wahrscheinlichkeitverstärkung für RP)

Für jedes $L \in RP$ und jedes Polynom $p: \mathbb{N} \to \mathbb{N}$ gibt es eine polynomiell zeitbeschränkte PTM M, s.d. f.a. $x \in \{0,1\}^*$ gilt:

- falls $x \in L$, so ist $Pr[M(x)=1] \geq 1 - \frac{1}{2^{p(|x|)}}$
- falls $x \notin L$, so ist $Pr[M(x)=0] = 1$

Beweis:

Wegen $L \in RP$ gibt es eine polynomiell zeitbeschränkte PTM M' s.d. f.a. $x \in \{0,1\}^*$ gilt

- falls $x \in L$, so $Pr[M'(x)=1] \geq 2/3$
- falls $x \notin L$, so $Pr[M'(x)=0] = 1$

Sei M die PTM, die bei Eingabe $x \in \{0,1\}^*$ folgendes tut:

1) Berechne $k := p(|x|)$
2) Starte unabhängig voneinander k Läufe von M' bei Eingabe x
3) Falls mindestens einer dieser k Läufe die Ausgabe 1 liefert, so gib 1 aus; ansonsten gib 0 aus

Offensichtlicherweise ist M polynomiell zeitbeschränkt.
Außerdem gilt:

- Falls $x \notin L$, so gibt jeder der k Läufe von M' bei Eingabe x den Wert 0 aus. M gibt dann auch 0 aus, und es gilt:

  $Pr[M(x)=0] = 1$.

- Falls $x \in L$, so gilt für jeden einzelnen der k Läufe von M' bei Eingabe x: $Pr[M'(x)=1] \geq 2/3$,

also $Pr[M'(x)=0] \leq \frac{1}{3} < \frac{1}{2}$.

M gibt nur dann 0 aus, wenn jeder der k Läufe 0 von M' ausgibt. D.h.:

$Pr[M(x)=0] < \left(\frac{1}{2}\right)^k$, und $Pr[M(x)=1] > 1-\left(\frac{1}{2}\right)^k = 1-\frac{1}{2^{p(|x|)}}$

!

Beispiel 7.8

Aus Theorem 7.1 und der Wk-Verstärkungsmethode aus dem Beweis von Lemma 7.7 erhalten wir für das Primzahlproblem $PRIMES := \{ \llcorner n \lrcorner : n \text{ ist eine Primzahl} \}$:

$PRIMES \in coRP$.

Bemerkung 7.9

(a) $RP \subseteq NP$, denn aus Theorem 2.7 folgt f.a. $L \subseteq \{0,1\}^*$:

$L \in NP \iff$ es gibt eine polynomiell zeitbeschränkte PTM M, s.d. f.a. Eingaben $x \in \{0,1\}^*$ gilt:
- falls $x \in L$, so $Pr[M(x)=1] > 0$
- falls $x \notin L$, so $Pr[M(x)=0] = 1$.

(b) $P \subseteq RP$ (klar.)

Probabilistische Komplexitätsklassen ohne Fehler

Ist M eine PTM und x eine Eingabe für M, so schreiben wir $T_{M,x}$, um die Zufallsvariable zu bezeichnen, die die Laufzeit von M bei Eingabe x angibt.

D.h.: Für $t \in \mathbb{N}$ und $0 \leq p \leq 1$ gilt

$\Pr[T_{M,x} \leq t] = p$ genau dann, wenn p die Wahrscheinlichkeit dafür ist, dass M bei Eingabe x nach höchstens t Schritten anhält.

Sei $T: \mathbb{N} \to \mathbb{N}$. Wir sagen:

<u>M hat erwartete Laufzeit $\leq T(n)$</u>, falls für jedes $x \in \{0,1\}^*$ gilt: $E[T_{M,x}] \leq T(|x|)$, d.h.:

Der Erwartungswert der Zufallsvariable $T_{M,x}$ ist $\leq T(|x|)$

<u>Definition 7.10</u> (ZTIME$(T(n))$, ZPP)

<u>(a)</u> Sei $T: \mathbb{N} \to \mathbb{N}$.

Die Klasse ZTIME$(T(n))$ besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, für die es eine PTM M mit den folgenden Eigenschaften gibt:

<u>(1)</u> M hat erwartete Laufzeit $O(T(n))$ und

<u>(2)</u> für jedes $x \in \{0,1\}^*$ gilt:

- falls $x \in L$, so gibt M bei Eingabe x bei jedem terminierenden Lauf den Wert 1 aus

- falls $x \notin L$, so gibt M bei Eingabe x bei jedem terminierenden Lauf den Wert 0 aus.

Bemerkung: Der Buchstabe "Z" in ZTIME(T(n)) steht für "zero-sided error" dh "ohne Fehler".

(b) $\text{ZPP} := \bigcup_{c \geq 1} \text{ZTIME}(n^c)$

("zero-error probabilistic polynomial time")

klar: $P \subseteq \text{ZPP}$

Theorem 7.11

$\text{ZPP} = \text{RP} \cap \text{coRP}$

Beweis:

"$\subseteq$": Sei $L \in \text{ZPP}$ und sei M eine PTM mit erwarteter Laufzeit $\leq n^c$, s.d. für jedes $x \in \{0,1\}^*$ gilt:

- falls $x \in L$, so gibt M bei Eingabe x bei jedem terminierenden Lauf den Wert 1 aus
- falls $x \notin L$, so gibt M bei Eingabe x bei jedem terminierenden Lauf den Wert 0 aus.

Für $i \in \{0,1\}$ sei M_i die $4 \cdot n^c$-zeitbeschränkte PTM, die folgendes tut: M_i simuliert die ersten $4 \cdot n^c$ Schritte von M. Falls M in dieser Zeit anhält, so gibt M_i die Ausgabe von M aus. Ansonsten gibt M die Ausgabe i aus.

man leicht nachweisen, dass folgendes f.a. $x \in \{0,1\}^*$ gilt: (Details: Übung):

- Falls $x \in L$, so $\Pr[M_0(x) = 1] \geq 2/3$ und $\Pr[M_1(x) = 1] = 1$
- Falls $x \notin L$, so $\Pr[M_0(x) = 0] = 1$ und $\Pr[M_1(x) = 0] \geq 2/3$

<u>Zur Erinnerung</u>: Die Markov-Ungleichung besagt, dass für alle Zufallsvariablen X mit Werten ≥ 0 und für alle Zahlen $a > 0$ gilt: $\Pr[X > a] \leq \frac{E[X]}{a}$.

Somit bezeugt M_0, dass $L \in RP$ ist.
M_1 bezeugt, dass $L \in coRP$ ist.
Also ist $ZPP \subseteq RP \cap coRP$.

<u>"⊇"</u>: Sei $L \in RP \cap coRP$, und seien M_1 und M_2 zwei PTMen, die dies bezeugen. Sei $c \in \mathbb{N}$ s.d. M_1 und M_2 n^c-zeitbeschränkt sind.

Sei M eine PTM, die bei Eingabe $x \in \{0,1\}^*$ folgendes tut:

Für $r = 1, 2, 3, \ldots$ tue folgendes:

1) simuliere n^c Schritte von M_1 bei Eingabe x; sei $a_1 \in \{0,1\}$ die entsprechende Ausgabe
2) simuliere n^c Schritte von M_2 bei Eingabe x; sei $a_2 \in \{0,1\}$ die entsprechende Ausgabe
3) Falls $a_1 = 1$, so STOPP mit Ausgabe 1
   Falls $a_2 = 0$, so STOPP mit Ausgabe 0

Wir wissen, dass f.a. $x \in \{0,1\}^*$ gilt:

- Falls $x \in L$, so $\Pr[M_1(x)=1] \geq 2/3$ und $\Pr[M_2(x)=1] = 1$
- Falls $x \notin L$, so $\Pr[M_1(x)=0] = 1$ und $\Pr[M_2(x)=0] \geq 2/3$

Somit gilt für jeden <u>terminierenden</u> Lauf von M bei Eingabe x:

- M gibt 1 aus $\Longrightarrow$ M_1 gibt 1 aus $\Longrightarrow$ $x \in L$
- M gibt 0 aus $\Longrightarrow$ M_2 gibt 0 aus $\Longrightarrow$ $x \notin L$

Dh: M erfüllt die Bedingung (<u>2</u>) von Definition 7.10.

<u>Laufzeitanalyse:</u>

Für jede einzelne Runde r von M gilt:

$$p := \Pr[a_1 = 1 \text{ oder } a_2 = 0] \geq 2/3$$

In jeder einzelnen Runde r werden $\leq 2n^c + 2$ Schritte durchgeführt.

Somit ist die erwartete Laufzeit von M bei Eingabe x $\leq$

$$\sum_{r=1}^{\infty} \underbrace{r \cdot (2n^c+2)}_{\substack{\text{Laufzeit bei STOPP} \\ \text{nach genau } r \\ \text{Runden}}} \cdot \underbrace{(1-p)^{r-1} \cdot p}_{\substack{\text{Wk dafür, dass} \\ \text{kein STOPP nach Runden } 1,\dots,r-1, \\ \text{aber STOPP nach } r \text{ Runden}}}$$

$$\leq (2n^c+2) \cdot p \cdot \sum_{r=1}^{\infty} r \cdot (1-p)^{r-1} = (2n^c+2) \cdot \frac{p}{(1-p)} \cdot \underbrace{\sum_{r=1}^{\infty} r \cdot (1-p)^r}_{= \frac{(1-p)}{p^2}} = (2n^c+2) \cdot$$

Denn: Für $0 < c < 1$ gilt:

$$\sum_{r=1}^{\infty} r \cdot c^r = \frac{c}{(1-c)^2}$$

(Nachrechnen: Übung!)

Somit hat M erwartete polynomielle Laufzeit und bezeugt daher, dass $L \in ZPP$ ist.

Somit ist $RP \cap coRP \subseteq ZPP$. $\square$

Komplexitätsklassen mit zweiseitigem, beschränktem Fehler

Notation 7.12

Für eine Sprache $L \subseteq \{0,1\}^*$ ist der Wert $L(x) \in \{0,1\}$ f.a. $x \in \{0,1\}^*$ wie folgt definiert:

$$L(x) := \begin{cases} 0 & \text{falls } x \notin L \\ 1 & \text{falls } x \in L. \end{cases}$$

Definition 7.13 (BPTIME $(T(n))$, BPP)

(a) Sei $T: \mathbb{N} \to \mathbb{N}$ und sei $L \subseteq \{0,1\}^*$.

Eine PTM M entscheidet L in Zeit $T(n)$, falls M $T(n)$-zeitbeschränkt ist und f.a. $x \in \{0,1\}^*$ gilt:

$$Pr[M(x) = L(x)] \geq 2/3.$$

D.h.: Falls $x \in L$, so ist $Pr[M(x) = 1] \geq 2/3$;
Falls $x \notin L$, so ist $Pr[M(x) = 0] \geq 2/3$.

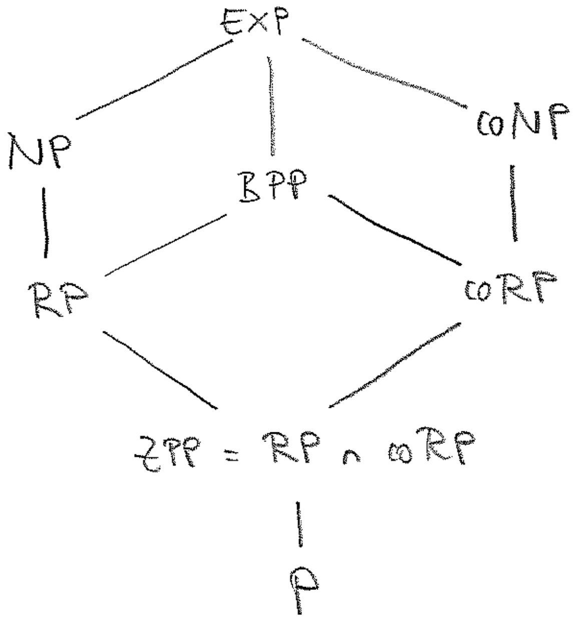
(b) Sei $T: \mathbb{N} \to \mathbb{N}$.
Die Klasse $\mathrm{BPTIME}(T(n))$ besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, die von einer $O(T(n))$-zeitbeschränkten PTM entschieden werden.

(c) $\mathrm{BPP} := \bigcup_{c \geq 1} \mathrm{BPTIME}(n^c)$

("bounded-error probabilistic polynomial time")

Bemerkung 7.14

Aus den Definitionen der Komplexitätsklassen sowie Theorem 7.11 und Bem. 7.16 ergibt sich folgende Inklusionsstruktur der einzelnen Klassen:



Offene Forschungsfrage: Ist BPP = P ?

Vermutung: Ja (!) ... und es gibt Resultate, die diese Vermutung sehr plausibel erscheinen lassen; siehe Kapitel 19 und 20 in [AB].

Ähnlich wie NP lässt sich BPP auch durch deterministische TMen mit "Zusatzeingabe" charakterisieren

Satz 7.15

Sei $L \subseteq \{0,1\}^*$. Es gilt:

$L \in BPP \iff$ Es gibt eine polynomiell zeitbeschränkte deterministische TM M und ein Polynom $p: \mathbb{N} \to \mathbb{N}$, s.d. f.a. $x \in \{0,1\}^*$ gilt:

$$\Pr_{r \in \{0,1\}^{p(|x|)}} \left[M(x,r) = L(x) \right] \geq 2/3 .$$

Dabei ist für $i \in \{0,1\}$

$$\Pr_{r \in \{0,1\}^{p(|x|)}} \left[M(x,r) = i \right] := \frac{|\{r \in \{0,1\}^{p(|x|)} : M(x,r) = i\}|}{2^{p(|x|)}} .$$

Beweis: Übung!

Bemerkung 7.16

Aus Satz 7.15 folgt direkt, dass $BPP \subseteq EXP$ ist.

Lemma 7.17 (Wahrscheinlichkeitverstärkung für BPP)

Sei $L \subseteq \{0,1\}^*$, sei p ein Polynom > 1 und sei M eine polynomiell zeitbeschränkte PTM, s.d. f.a. $x \in \{0,1\}^*$ gilt

$$\Pr\left[M(x) = L(x) \right] \geq \frac{1}{2} + \frac{1}{p(|x|)}.$$

Dann gibt es für jedes Polynom q eine polynomiell zeitbeschränkte PTM M', s.d. f.a. $x \in \{0,1\}^*$ gilt:

$$\Pr\left[M'(x) = L(x) \right] \geq 1 - \frac{1}{2^{q(|x|)}} \quad (*)$$

Beweis:

Bei Eingabe $x \in \{0,1\}^*$ geht M' wie folgt vor.

1) Berechne $k := 2 \cdot \ln 2 \cdot q(|x|) \cdot p(|x|)^2$

2) Starte unabhängig voneinander k Läufe von M bei Eingabe x;
seien $y_1, \dots, y_k \in \{0,1\}$ die Ausgaben dieser k Läufe

3) Falls die Mehrheit der $y_1, \dots, y_k$ gleich 1 ist, gib 1 aus; ansonsten gib 0 aus.

Offensichtlicherweise ist M' eine polynomiell zeitbeschränkte PTM.

Sei $n := |x|$. Um $(*)$ nachzuweisen, reicht es, zu zeigen, dass gilt:

$$\Pr[M'(x) \neq L(x)] \leq 2^{-q(n)} = e^{-\ln 2 \cdot q(n)} \quad (**)$$

Dazu sei f.a. $i \in \{1, \dots, k\}$ X_i eine Zufallsvariable mit

$$s := \Pr[X_i = 1] = \Pr[M(x) = L(x)] \geq \frac{1}{2} + \frac{1}{p(n)} \quad \text{und}$$

$$1 - s = \Pr[X_i = 0].$$

Sei $X := \sum_{i=1}^{k} X_i$. D.h.: X gibt an, wie viele der k Aufrufe von M bei Eingabe x das korrekte Ergebnis liefern.

Außerdem gilt:

$M'(x) \neq L(x)$ $\iff$ M' gibt bei Eingabe x das falsch Ergebnis aus
$\iff$ weniger als $k/2$ der Aufrufe von M bei Eingabe x liefern das korrekte Ergebnis
$\iff$ $X < k/2$

Um $\Pr[M'(x) \neq L(x)] = \Pr[X < k/2]$ abzuschätzen, nützen wir das folgende aus der Stochastik bekannte Lemma:

Lemma 7.18 (Chernoff-Schranke)

Sei $k > 0$, seien $X_1, \dots, X_k$ unabhängige Zufallsvariablen, sei s mit $0 < s < 1$ s.d. f.a. $i \in \{1, \dots, k\}$ gilt:
$\Pr[X_i = 1] = s$ und $\Pr[X_i = 0] = 1 - s$.
Sei $X := \sum_{i=1}^{k} X_i$. Dann gilt:

(a) $E[X] = k \cdot s$ und

(b) F.a. δ mit $0 < \delta \leq 1$ ist
$$\Pr[X < (1-\delta) \cdot E[X]] < e^{-\frac{\delta^2}{2} \cdot E[X]}$$

(Hier ohne Beweis)

Um die Chernoff-Schranke anzuwenden, wählen wir δ so, dass $(1-\delta)\cdot E[X] = \frac{k}{2}$ ist.

D.h.: $\frac{k}{2} = (1-\delta)\cdot k \cdot s$

$\Leftrightarrow \quad \frac{1}{2} = (1-\delta)\cdot s = s - \delta s$

$\Leftrightarrow \quad \delta s = s - \frac{1}{2}$

$\Leftrightarrow \quad \delta = 1 - \frac{1}{2s}$.

Beachte: Wegen $s>0$ ist $\delta<1$; wegen $s>\frac{1}{2}$ ist $\delta>0$

Daher können wir die Chernoff-Schranke anwenden und erhalten:

$$\Pr\left[X < \frac{k}{2}\right] = \Pr\left[X < (1-\delta)\cdot E[X]\right] < e^{-\frac{\delta^2}{2}\cdot E[X]} = e^{-\delta^2/2\cdot k\cdot s}$$

Um (**) nachzuweisen, müssen wir zeigen, dass

$$-\frac{\delta^2}{2}\cdot k\cdot s \leq -\ln 2\cdot q(n).$$

D.h. es bleibt zu zeigen, dass

$$k \geq 2\cdot\ln 2\cdot q(n)\cdot\frac{1}{s\cdot\delta^2} \quad \text{ist.}$$

Wegen $k = 2\ln 2\cdot q(n)\cdot p(n)^2$ müssen wir also nur noch zeigen, dass $p(n)^2 \geq \frac{1}{s\cdot\delta^2}$ ist.

Somit bleibt zu zeigen: $s\cdot\delta^2 \geq \left(\frac{1}{p(n)}\right)^2$

Dazu sei ε s.d. $s = \frac{1}{2} + \varepsilon$.

Wegen $\frac{1}{2} + \frac{1}{p(n)} \leq s \leq 1$ gilt: $\frac{1}{p(n)} \leq \varepsilon \leq \frac{1}{2}$.

Wegen $\delta = 1 - \frac{1}{2s}$ gilt:

$$s \cdot \delta^2 = s\left(1 - \frac{1}{2s}\right)^2 = s \cdot \left(1 - \frac{1}{s} + \frac{1}{4s^2}\right) = s - 1 + \frac{1}{4s}$$

$$= \frac{1}{2} + \varepsilon - 1 + \frac{1}{4(\frac{1}{2}+\varepsilon)} = \varepsilon - \frac{1}{2} + \frac{1}{2+4\varepsilon}$$

$$= \frac{2\varepsilon + 4\varepsilon^2 - 1 - 2\varepsilon + 1}{2+4\varepsilon} = \frac{4\varepsilon^2}{2+4\varepsilon}$$

$$\geq \frac{4\varepsilon^2}{4} = \varepsilon^2 \geq \left(\frac{1}{p(n)}\right)^2.$$

$2 + 4\varepsilon \leq 4$, da $\varepsilon \leq \frac{1}{2}$

Insgesamt haben wir damit gezeigt, dass

$$\Pr[M'(x) \neq L(x)] = \Pr\left[X < \frac{k}{2}\right] < e^{-\ln 2 \cdot q(n)} = \frac{1}{2^{q(n)}}.$$

Daher gilt

$$\Pr[M'(x) = L(x)] \geq 1 - \frac{1}{2^{q(n)}}. \quad \square$$

7.3 Derandomisierung: BPP vs P/poly und PH

Theorem 7.19 (Adelman, 1978)

$$BPP \subseteq P/poly$$

Beweis:

Sei $L \in BPP$.

Gemäß Satz 7.15 und dem Wahrscheinlichkeitsverstärkungs-Lemma 7.17 gibt es eine det. TM M und ein Polynom p, s.d. f.a. $x \in \{0,1\}^*$ gilt:

$$\Pr_{r \in \{0,1\}^{p(|x|)}} \left[M(x,r) = L(x) \right] \geq 1 - \frac{1}{2^{|x|+1}}.$$

Also:

$$\Pr_{r \in \{0,1\}^{p(|x|)}} \left[M(x,r) \neq L(x) \right] \leq \frac{1}{2^{|x|+1}}. \quad (*)$$

Wir nennen eine Zusatzeingabe $r \in \{0,1\}^{p(|x|)}$ schlecht für x, falls $M(x,r) \neq L(x)$ gilt; ansonsten heißt r gut für x.

Wegen $(*)$ gibt es für jedes $n \in \mathbb{N}$ und $x \in \{0,1\}^n$ höchstens

$$\frac{2^{p(n)}}{2^{n+1}}$$

Zusatzeingaben, die schlecht für x sind.

Daher gibt es höchstens

$$\sum_{x \in \{0,1\}^n} \frac{2^{p(n)}}{2^{n+1}} = 2^n \cdot \frac{2^{p(n)}}{2^{n+1}} = \frac{2^{p(n)}}{2}$$

Zusatzeingaben, die für mindestens ein $x \in \{0,1\}^n$ schlecht sind.

Da es insgesamt $2^{p(n)} > \frac{2^{p(n)}}{2}$ Zusatzeingaben gibt, muss es mindestens eine Zusatzeingabe $r_n \in \{0,1\}^{p(n)}$ geben, die für <u>jedes</u> $x \in \{0,1\}^n$ gut ist.

D.h.: F.a. $x \in \{0,1\}^n$ ist $M(x, r_n) = L(x)$.

Gemäß Theorem 6.20 $\left(P/poly = \bigcup_{c,d \geq 1} DTIME(n^c)/n^d \right)$

folgt somit: $L \in P/poly$.

$\square$

<u>Bemerkung 7.20</u>

Wegen $BPP \subseteq P/poly$ und

Theorem 6.16 (Falls $NP \subseteq P/poly$, so kollabiert die PH)

ist vermutlich $BPP \neq NP$.

Außerdem gilt: Falls $SAT \in BPP$, so kollabiert die PH

$\boxed{\text{Exkurs:}}$ Begriffe aus der Wahrscheinlichkeitsrechnung

Sei $\mathcal{U}$ eine endliche Menge.

(a) Eine Wahrscheinlichkeitsverteilung über $\mathcal{U}$ ist eine Funktion $\pi : \mathcal{U} \to [0,1]$, so dass gilt:

$$\sum_{u \in \mathcal{U}} \pi(u) = 1.$$

(b) Die Elemente von $\mathcal{U}$ heißen Elementarereignisse.
Teilmengen von $\mathcal{U}$ heißen Ereignisse.

Die Warscheinlichkeit eines Ereignisses $V \subseteq \mathcal{U}$ ist

$$\Pr[V] := \sum_{u \in V} \pi(u)$$

(c) Eine Zufallsvariable über $\mathcal{U}$ ist eine Funktion $X : \mathcal{U} \to \mathbb{R}$.

Der Erwartungswert von X ist

$$E[X] := \sum_{u \in \mathcal{U}} \pi(u) \cdot X(u).$$

Die Varianz von X ist

$$\mathrm{Var}[X] := E\left[X^2 - E[X]^2 \right]$$

Theorem 7.21 (Satz von Sipser und Gacs, 1983)

$$\mathrm{BPP} \subseteq \Sigma_2^P \cap \Pi_2^P$$

Beweis: (Lautemann-Methode)

Wegen $\mathrm{BPP} = \mathrm{coBPP}$ (gemäß der Definition von BPP) genügt es zu zeigen, dass $\mathrm{BPP} \subseteq \Sigma_2^P$ ist.

Sei $L \in \mathrm{BPP}$.

Gemäß Satz 7.15 und dem Wahrscheinlichkeitsverstärkungs-Lemma 7.17 gibt es eine det. Polynomialzeit TM M und ein Polynom p, s.d. f.a. $x \in \{0,1\}^*$ gilt:

$$\Pr_{r \in \{0,1\}^{p(|x|)}} \left[M(x,r) = L(x) \right] \geq 1 - \frac{1}{2^{|x|}}$$

D.h.: F.a. $n \in \mathbb{N}$ und f.a. $x \in \{0,1\}^n$ gilt:

- Falls $x \in L$, so $\dfrac{|\{r \in \{0,1\}^{p(n)} : M(x,r) = 1\}|}{2^{p(n)}} \geq 1 - \dfrac{1}{2^n}$

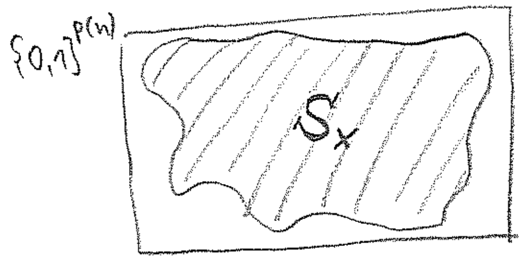
- Falls $x \notin L$, so $\dfrac{|\{r \in \{0,1\}^{p(n)} : M(x,r) = 1\}|}{2^{p(n)}} \leq \dfrac{1}{2^n}$.

Sei $S_x := \{r \in \{0,1\}^{p(n)} : M(x,r) = 1\}$. Somit gilt:

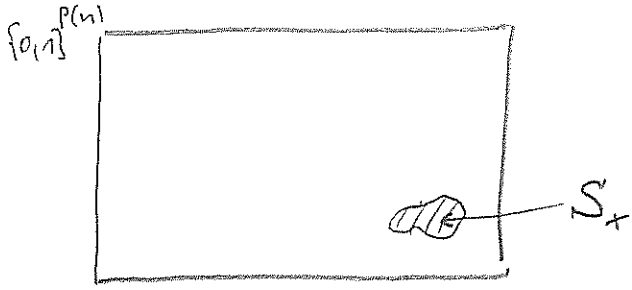
$$\left.\begin{array}{l} \bullet \text{ Falls } x \in L, \text{ so } |S_x| \geq \left(1 - \frac{1}{2^n}\right) \cdot 2^{p(n)} \\ \bullet \text{ Falls } x \notin L, \text{ so } |S_x| \leq \frac{1}{2^n} \cdot 2^{p(n)} \end{array}\right\} \circledast$$

<u>Skizze:</u>

$x \in L \Rightarrow$



$x \notin L \Rightarrow$



<u>Notation:</u>

Für $u, v \in \{0,1\}^{p(n)}$ sei

$$u+v := (u_1+v_1 \bmod 2,\ u_2+v_2 \bmod 2, \ldots, u_{p(n)}+v_{p(n)} \bmod 2)$$

Für $u \in \{0,1\}^{p(n)}$ und $S \subseteq \{0,1\}^{p(n)}$ sei

$S+u := \{ v+u : v \in S \}$ die "Verschiebung" von S um u.

Beachte: $u+v = u-v$ und $S+u = S-u$.

Sei $k(n) := \left\lceil \frac{p(n)}{n} + 1 \right\rceil$

<u>Behauptung 1</u>: F.a. hinreichend großen $n \in \mathbb{N}$ gilt:

Für jedes $S \subseteq \{0,1\}^{p(n)}$ mit $|S| \leq \frac{2^{p(n)}}{2^n}$ und für alle $u^{(1)}, \ldots, u^{(k(n))} \in \{0,1\}^{p(n)}$ gilt:

$$\bigcup_{i=1}^{k(n)} (S+u^{(i)}) \neq \{0,1\}^{p(n)}$$

<u>Beweis:</u>

$$\left| \bigcup_{i=1}^{k(n)} (S+u^{(i)}) \right| \leq \sum_{i=1}^{k(n)} |S+u^{(i)}| = k(n) \cdot |S| \leq \left\lceil \frac{p(n)}{n} + 1 \right\rceil \cdot \frac{2^{p(n)}}{2^n}$$

$$\underset{\substack{\uparrow \\ \text{für hinreichend großes } n}}{<} 2^{p(n)} = |\{0,1\}^{p(n)}|.$$

□ Beh 1

Behauptung 2: Für jedes $n \geq 1$ gilt:

Für jedes $S \subseteq \{0,1\}^{p(n)}$ mit $|S| \geq \left(1-\frac{1}{2^n}\right)\cdot 2^{p(n)}$

gibt es $u^{(1)},\dots,u^{(k(n))} \in \{0,1\}^{p(n)}$, so dass

$$\bigcup_{i=1}^{k(n)} \left(S+u^{(i)}\right) = \{0,1\}^{p(n)}$$

Bevor wir Behauptung 2 beweisen, zeigen wir zunächst, wie man aus ⊛, Beh 1 und Beh 2 erhält, dass $L \in \Sigma_2^P$ ist:

Für jedes hinreichend große n und jedes $x \in \{0,1\}^n$ gilt gemäß ⊛, Beh 1 und Beh 2:

$$x \in L$$

$$\iff \exists u^{(1)},\dots,u^{(k(n))} \in \{0,1\}^{p(n)} : \forall r \in \{0,1\}^{p(n)} : \underbrace{r \in \bigcup_{i=1}^{k(n)} \left(S_x + u^{(i)}\right)}_{\iff \bigvee_{i=1}^{k(n)} \left(r+u^{(i)} \in S_x\right)}$$

$$\iff \exists u^{(1)},\dots,u^{(k(n))} \in \{0,1\}^{p(n)} : \forall r \in \{0,1\}^{p(n)} : \underbrace{\bigvee_{i=1}^{k(n)} M(\langle x, r+u^{(i)}\rangle) = 1}$$

Wegen $k(n) = \text{poly}(n)$ kann dies bei gegebenem $x, u^{(1)},\dots,u^{(k(n))}$ von einer det. Polynomialzeit-TM überprüft werden.

Insgesamt erhalten wir, dass $L \in \Sigma_2^P$ ist.

Zum Abschluss des Beweises von Theorem 7.21 genügt es also, Behauptung 2 zu beweisen.

<u>Beweis von Behauptung 2:</u>

Wir nutzen die sog. probabilistische Methode:

Wähle $u^{(1)}, \dots, u^{k(n)}$ zufällig und unabhängig voneinander aus $\{0,1\}^{p(n)}$.

Für $i \in \{1, \dots, k(n)\}$ und $r \in \{0,1\}^{p(n)}$ sei

- $B_{i,r}$ das Ereignis, dass $r \notin S + u^{(i)}$ (d.h.: $r + u^{(i)} \notin S$)
- B_r das Ereignis, dass $r \notin \bigcup_{i=1}^{k(n)} (S + u^{(i)})$

D.h.
- $B_r \iff B_{1,r}$ und $B_{2,r}$ und $\dots$ und $B_{k(n),r}$ und
- $\bigcup_{i=1}^{k(n)} (S + u^{(i)}) \neq \{0,1\}^{p(n)} \iff$ ex $r \in \{0,1\}^{p(n)}$ s.d. das Ereignis B_r eintritt

Da die $u^{(i)}$ unabhängig voneinander gewählt werden, gilt

$$\Pr[B_r] = \prod_{i=1}^{k(n)} \Pr[B_{i,r}]$$

Ereignis $B_{i,r}$ tritt genau dann ein, wenn $r + u^{(i)} \notin S$.

Da $u^{(i)}$ zufällig gewählt wird gilt:

$$\Pr[B_{i,r}] = \frac{|\{0,1\}^{p(n)} \setminus S|}{|\{0,1\}^{p(n)}|}$$

Wegen $|S| \geq \left(1 - \frac{1}{2^n}\right) \cdot 2^{p(n)}$ ist $|\{0,1\}^{p(n)} \setminus S| \leq \frac{1}{2^n} \cdot 2^{p(n)}$.

Somit ist $\Pr[B_{i,r}] \leq \dfrac{\frac{1}{2^n} \cdot 2^{p(n)}}{2^{p(n)}} = \dfrac{1}{2^n}$

und $\Pr[B_r] = \prod_{i=1}^{k(n)} \Pr[B_{i,r}] \leq \left(\dfrac{1}{2^n}\right)^{k(n)} = \dfrac{1}{2^{n \cdot k(n)}}$

Wegen $k(n) = \frac{p(n)}{n} + 1$ ist $n \cdot k(n) \geq p(n) + n > p(n)$

Somit ist $Pr[B_r] < \frac{1}{2^{p(n)}}$ und

$$Pr\left[\begin{matrix}\text{ex. } r \in \{0,1\}^{p(n)} \text{ s.d.} \\ \text{das Ereignis } B_r \text{ eintritt}\end{matrix}\right] \leq \sum_{r \in \{0,1\}^{p(n)}} Pr[B_r] < 2^{p(n)} \cdot \frac{1}{2^{p(n)}} = 1$$

$$\shortparallel$$

$$Pr\left[\bigcup_{i=1}^{k(n)} (S + u^{(i)}) \neq \{0,1\}^{p(n)}\right]$$

Wir haben also folgendes gezeigt:
Wenn wir $u^{(1)}, \ldots, u^{(k(n))}$ zufällig aus $\{0,1\}^{p(n)}$ wählen, so ist die Wahrscheinlichkeit, dass $\bigcup_{i=1}^{k(n)} (S + u^{(i)}) \neq \{0,1\}^{p(n)}$ ist, echt kleiner als 1.

Somit muss es eine Möglichkeit geben, $u^{(1)}, \ldots, u^{(k(n))}$ zu wählen, so dass $\bigcup_{i=1}^{k(n)} (S + u^{(i)}) = \{0,1\}^{p(n)}$ ist.

$\square$ Beh 2

Dies schließt den Beweis von Theorem 7.21 ab.

$\square$

Wegen $BPP \subseteq \Sigma_2^P$ folgt insbesondere:
Falls $P = NP$ ist, so ist $PH = P$ und $BPP = P$.

7.4 Vollständige Probleme bzw Hierarchiesätze für BPP?

Es sind keine vollständigen Probleme für BPP bekannt.

Ein naheliegender Versuch, ein vollständiges Problem zu definieren ist, die Sprache

$$L' := \{ \langle M, x, 1^t \rangle : \text{ M ist eine PTM, die bei Eingabe x nach t Berechnungsschritten mit Wahrscheinlichkeit} \geq 2/3 \text{ den Wert } \underline{1} \text{ ausgibt} \}$$

Man sieht leicht, dass L' BPP-hart ist bzgl. Polynomialzeitreduktionen — dh f.a. $L \in BPP$ ist $L \leq_p L'$.

Aber es ist unklar, ob $L' \in BPP$ ist. Denn:
Für $\langle M, x, 1^t \rangle \notin L'$ könnte es z.B. sein, dass $\Pr[M(x) = 0] = \frac{1}{2} < \frac{2}{3}$ ist — und somit wäre M keine im Sinne von BPP geeignete PTM.

In der Tat liegt L' vermutlich nicht in BPP, da bekannt ist, dass L' vollständig ist für eine Komplexitätsklasse, die vermutlich deutlich größer als BPP ist.

Andererseits: Falls die Vermutung "BPP = P" tatsächlich wahr ist, so besitzt BPP natürlich vollständige Probleme.

Es sind keine Zeithierarchiesätze für BPP bekannt.
So ist z.B. unklar, ob

- $BPTIME(n) \neq BPTIME(n^2)$ oder ob
- $BPTIME(n) \neq BPTIME(n^{((\log n)^{10})})$

ist.

Die in Kapitel 3 betrachteten Diagonalisierungsmethoden funktionieren hier nicht, da wir für eine gegebene PTM M nicht entscheiden können, ob f.a. Eingaben $x \in \{0,1\}^*$ gilt:
entweder $Pr[M(x)=1] \geq 2/3$ oder $Pr[M(x)=1] \leq 1/3$.

7.5 Randomisierte Platzbeschränkte Berechnungen

Ähnlich wie BPP und RP randomisierte Varianten von P sind, können wir randomisierte Varianten der Klasse L aller auf logarithmischem Platz berechenbaren Probleme definieren:

Definition 7.22 (BPL und RL)

(a) Die Klasse BPL besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, für die es eine $O(\log n)$-platz-beschränkte PTM M gibt, s.d. f.a. $x \in \{0,1\}^*$ gilt:
$$Pr[M(x) = L(x)] \geq 2/3.$$

(b) Die Klasse RL besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, für die es eine $O(\log n)$-platz-beschränkte PTM M gibt, s.d. f.a. $x \in \{0,1\}^*$ gilt:

- falls $x \in L$, so ist $\Pr[M(x)=1] \geq 2/3$
- falls $x \notin L$, so ist $\Pr[M(x)=0] = 1$.

Bemerkung 7.23

(a) Die Wahrscheinlichkeitsverstärkungs-Lemmata 7.7 und 7.17 gelten analog auch für RL und BPL:

Man kann sich leicht davon überzeugen, dass die im Beweis von Lemma 7.7 und 7.17 konstruierten PTMen M' nur logarithmisch mehr Platz benutzen als die gegebenen PTMen M

(b) Offensichtlich gilt: $L \subseteq RL \subseteq NL \subseteq P$

(c) Man kann zeigen, dass $BPL \subseteq P$ gilt (Übung!)

Theorem 7.24

$UPATH \in RL$, wobei

$UPATH := \{ \langle G,s,t \rangle :$ G ist ein endlicher ungerichteter Graph, in dem es einen Weg von Knoten s zu Knoten t gibt $\}$

Beweisidee:

Konstruiere eine PTM M, die bei Eingabe von $\langle G,s,t\rangle$ folgenden randomisierten Algorithmus ausführt:

Random Walk on $\langle G,s,t\rangle$

$n := |V(G)|$

$i := 0$

$u := s$

while $u \neq t$ and $i < 100 \cdot n^4$ do

  sei v ein zufällig gewählter Nachbarknoten von u (d.h.: $\{u,v\} \in E(G)$).

  $i := i+1$

  $u := v$

Falls $u = t$, so STOPP mit Ausgabe "1"

Sonst: STOPP mit Ausgabe "0".

Dieser Algorithmus lässt sich leicht durch eine $O(\log n)$-platzbeschränkte PTM M realisieren.

Falls t von s aus nicht erreichbar ist, so gilt offensichtlicherweise: $Pr[M(\langle G,s,t\rangle)=0] = 1$.

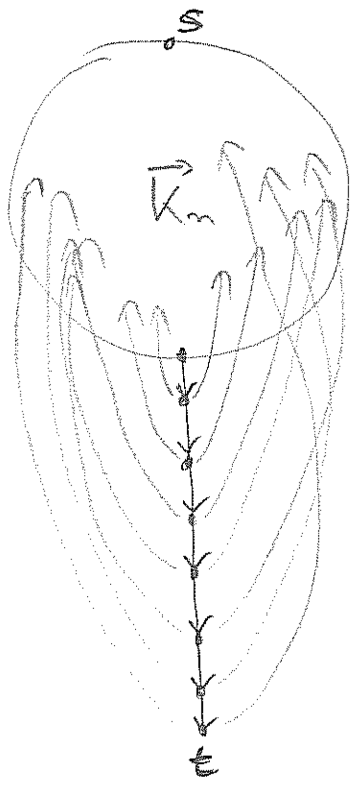
Unter Verwendung der Theorie der Markov-Ketten kann man folgendes beweisen (hier ohne Beweis):

Falls t von s aus erreichbar ist, so ist die erwartete Anzahl von Schritten im "random walk" bis Knoten t erreicht ist $\leq 10\,n^4$, und daher ist die Wahrscheinlichkeit, dass t innerhalb von $100 \cdot n^4$ Schritten erreicht wird, $\geq 2/3$.

$\square$

<u>Bemerkung 7.25</u>

Auf <u>gerichteten</u> Graphen funktioniert die Random-Walk-Methode nicht so gut. Betrachtet man z.B. einen Graphen der Form,



($\vec{K}_n$: gerichteter vollständiger Graph auf n Knoten mit Kantenmenge $V \times V$)

So kann man nachweisen, dass ein in s startender Random Walk Erwartungswert deutlich mehr als $10\,n^4$ Schritte braucht, um Knoten t zu erreichen.