Probabilistische Komplexitätsklassen

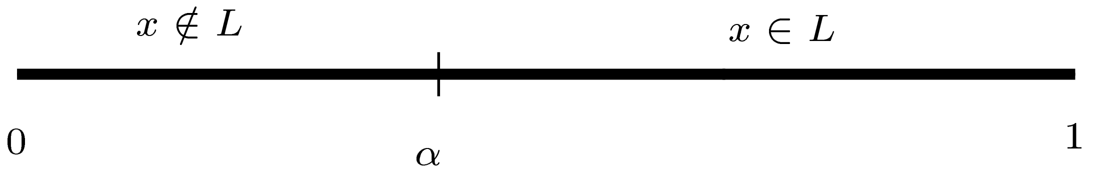
---

1. **Probabilistische Algorithmen**
2. **Wahrscheinlichkeitsverstärkung**
3. **Recycling von Zufallszahlen**

Probabilistische Algorithmen

Definition 1 (Formale Sprache)
Ein Alphabet Σ *sei eine endliche, nicht-leere Menge, deren Elemente* Zeichen *oder* Symbole *genannt werden. Für ein gegebenes Alphabet* Σ *bezeichne* Σ^* *die Menge aller Wörter, die sich durch Hintereinanderschreiben von endlich vielen Symbolen aus* Σ *bilden lassen. Eine formale Sprache* L *ist jede beliebige Teilmenge von* Σ^**. Das Komplement* $\bar{L}$ *ist* $\{x \in \Sigma^* | x \notin L\}$*.*

Definition 2 (Probabilistisches Akzeptieren)
Ein Wort gilt als akzeptiert, wenn ein probabilistischer Algorithmus mit einer Wahrscheinlichkeit größer einem festen Schwellwert $\alpha \in [0, 1]$ *die Entscheidung „Akzeptiert" ausgibt.*

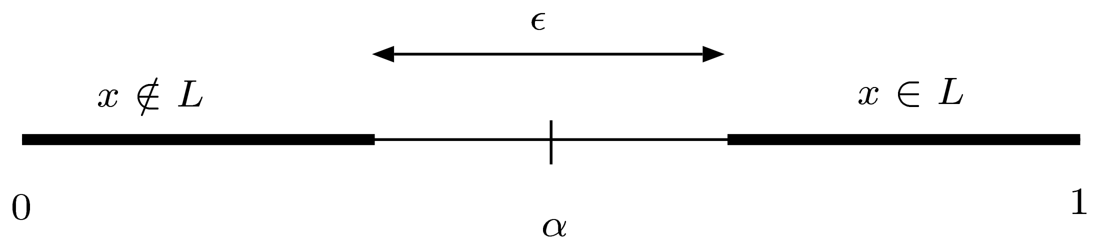


Definition 3 (Komplexitätsklasse PP)
Eine Sprache L *liegt in der Komplexitätsklasse PP (probabilistic polynomial time), falls es einen polynomial zeitbeschränkten Algorithmus und einen Schwellwert* α *gibt, so genau alle Wörter aus* L *akzeptiert werden.*

Probabilistische Algorithmen

Bemerkung 1 (Wahrscheinlichkeitslücke) Akzeptiert ein probabilistischer Algorithmus mit einer Wahrscheinlichkeit von $\geq \alpha + \epsilon/2$ und verwirft mit einer Wahrscheinlichkeit von $\leq \alpha - \epsilon/2$ $(\epsilon > 0)$, so spricht man von einer Wahrscheinlichkeitslücke zwischen den Wörtern $x \in L$ und $x \notin L$.

Definition 4 (Komplexitätsklasse BPP)
Eine Sprache L liegt in der Komplexitätsklasse BPP (bounded-error probabilistic polynomial time), falls es einen polynomial zeitbeschränkten Algorithmus und einen Schwellwert α und eine Wahrscheinlichkeitslücke $\epsilon > 0$ gibt, so genau alle Wörter aus L akzeptiert werden und für kein Eingabewort die Akzeptanzwahrscheinlichkeit im Intervall $[\alpha - \epsilon/2, \alpha + \epsilon/2]$ liegt.



Bemerkung 2 $\mathrm{P} \subseteq \mathrm{BPP} \subseteq \mathrm{PP}$

Probabilistische Algorithmen

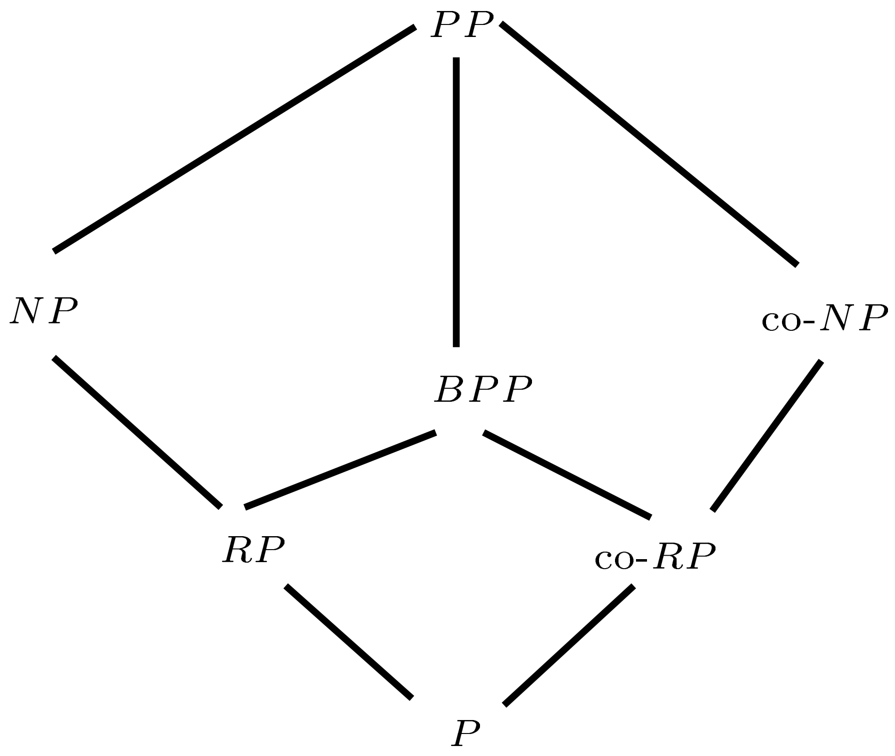
Definition 5 (Komplexitätsklasse RP)
Eine Sprache L liegt in der Komplexitätsklasse RP, falls es einen BPP-Algorithmus für L gibt mit $\epsilon = 2\alpha$*. D.h. die Maschine akzeptiert mit einer Wahrscheinlichkeit* $\geq \epsilon$ *für* $x \in L$ *oder mit Wahrscheinlichkeit* 0 *bei* $x \notin L$*. (Akzeptiert die Maschine mindestens einmal, so ist* $x \in L$*.)*

Definition 6 (Komplexitätsklasse ZPP)
Eine Sprache L liegt in der Komplexitätsklasse ZP (zero error probabilistic polynomial time) genau dann, wenn $L \in RP$ *und* $\bar{L} \in RP$*. (*$ZPP = RP \bigcap co\text{-}RP$*)*

Bemerkung 3 (Zero Error) Für die Komplexitätsklasse ZPP kann eine polynomial zeitbeschränkter Algorithmus angegeben werden, mit einer Wahrscheinlichkeit $p \geq \epsilon$ verwirft bzw. akzeptiert und mit einer Wahrscheinlichkeit $1 - p$ „ungewiß“ ausgibt.

Probabilistische Algorithmen

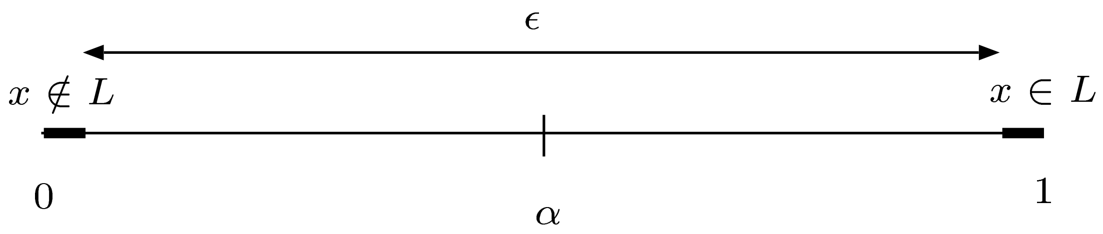
Inklusionsstruktur der Komplexitätsklassen



BEMERKUNG 4 (KOMPLEMENT EINER KOMPLEXITÄTSKLASSE) Ein Sprache L liegt in der Komplexitätsklasse co-K, wenn ihr Komplement $\bar{L} = \{x \in \Sigma^* | x \notin L\}$ in K liegt.

Wahrscheinlichkeitsverstärkung

Bemerkung 5 (Wahrscheinlichkeitsverstärkung) Wahrscheinlichkeitsverstärkung bedeutet, die Wahrscheinlichkeitslücke zwischen den Wörtern $x \in L$ und $x \notin L$ mit polynomialen Zusatzaufwand größer zu machen, d.h. $x \in L$ wird fast immer akzeptiert und $x \notin L$ fast immer verworfen.



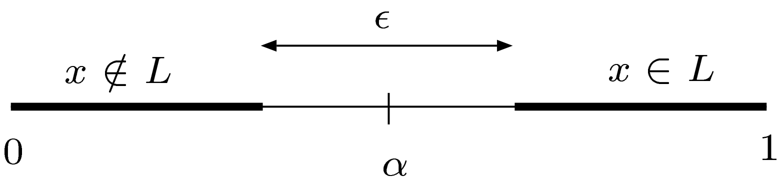
Bei der Komplexitätsklasse RP ist die Ausgabe „Akzeptieren“ immer korrekt, d.h. $x \in L$.

Indem man den Algorithmus mehrere Male auf dieselbe Eingabe anwendet und genau dann akzeptiert, wenn mindestens eins der Ergebnisse „Akzeptieren“ lautet, erhält man einen Algorithmus mit einer viel größeren Wahrscheinlichkeitslücke:

$$
\begin{aligned}
p_t(REJECT|x \in L) &\leq (1-\epsilon)^t && \text{t: Anzahl der Versuche} \\
p_t(ACCEPT|x \in L) &\geq 1-(1-\epsilon)^t
\end{aligned}
$$

Wahrscheinlichkeitsverstärkung

Bei der Komplexitätsklasse BPP sind sowohl die Ausgabe „Akzeptieren“ als auch „Verwerfen“ fehlerbehaftet.



Folgender Algorithmus kann eine Wahrscheinlichkeitsverstärkung bringen:

```
INPUT x;  s := 0;
FOR i := 1 TO t
{
    Simuliere M auf x;
    IF (Ergebnis == ACCEPT) THEN s := s+1
}
IF (s > αt) THEN ACCEPT ELSE REJECT
```

$$
\begin{aligned}
p_M(ACCEPT|x \in L) &\geq \alpha + \epsilon/2 := \gamma \\
p_t(REJECT|x \in L) &\leq \sum_{i=0}^{\alpha t} \binom{t}{i} \gamma^i \cdot (1-\gamma)^{t-i} \leq 2^{-\Omega(t)} \\
p_t(ACCEPT|x \in L) &\geq 1 - 2^{-\Omega(t)}
\end{aligned}
$$

Wahrscheinlichkeitsverstärkung

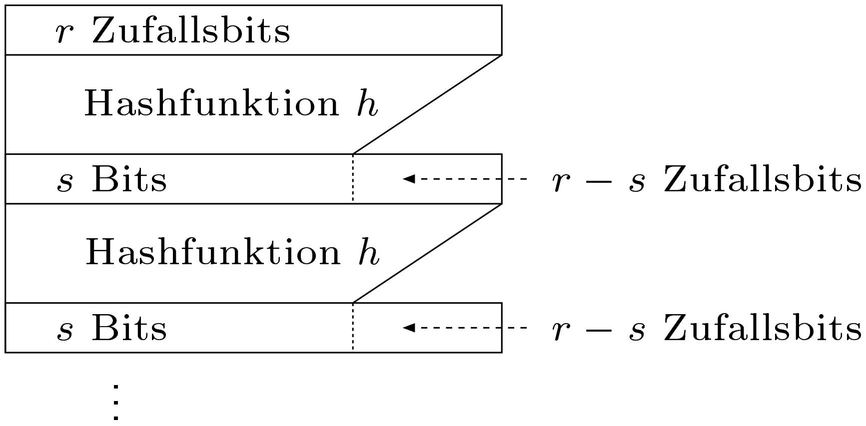
$$\begin{aligned} p_M(ACCEPT|x \notin L) &\leq \alpha - \epsilon/2 := \delta \\ p_t(ACCEPT|x \notin L) &\leq \sum_{i=\alpha t}^{t} \binom{t}{i} \delta^i \cdot (1-\delta)^{t-i} \leq 2^{-\Omega(t)} \end{aligned}$$

Bemerkung 6 (BPP-Algorithmus mit kleiner Fehlerwahrscheinlichkeit) Für jede Sprache $L \in BPP$ gibt es einen BPP-Algorithmus für L, der eine Fehlerwahrscheinlichkeit von $2^{-\Omega(t)}$ hat. Wählt man t in Abhängigkeit von der Länge der Eingabe $|x|$, so kann man für jedes Polynom p eine Fehlerwahrscheinlichkeit von $2^{-p(|x|)}$ erreichen.

Recycling von Zufallszahlen

Um eine Wahrscheinlichkeitslücke von $\epsilon > 0$ aus $2^{-\Omega(t)}$ auszuweiten, muß man den probabilistischen Algorithmus $O(k)$ mal wiederholen. Falls der zugrundeliegende Algorithmus r Zufallsbits benötigt, so braucht der wahrscheinlichkeitsverstärkte Algorithmus $O(rk)$ Zufallsbits.

Ansatz 1 *Aus den ersten r Zufallsbits werden durch eine Hashfunktion $s < r$ neue erzeugt und mit $r - s$ „frischen“ Zufallsbits ergänzt. Als Komplexität erhalten wir: $r + (r - s)(k - 1)$.*



Kann $r - s = O(k)$ gewählt werden, so ergibt dies einen Aufwand von $O(r + k^2)$, der sich durch andere Methoden auf $O(r + k)$ herunterdrücken läßt.

Recycling von Zufallszahlen: Universalität von Hashfunktionen

Bemerkung 7 (Kollisionswahrscheinlichkeit) Die Kollisionswahrscheinlichkeit einer Verteilung D ist die Wahrscheinlichkeit, daß zwei Elemente x und y, die unabhängig unter D gezogen wurden, gleich sind.

Definition 7 (Universalität von Hashfunktionen)
Eine Klasse von Hashfunktionen von $\{0,1\}^r$ *nach* $\{0,1\}^s$ *heißt* universal, *falls für alle* $x, y \in \{0,1\}^r, x \neq y$ *gilt:* $P[h(x) = h(y)] = 1/2^s$.

Eine Klasse von Hashfunktionen heißt fast universal, *wenn unter obigen Voraussetzungen gilt:* $P[h(x) = h(y)] \leq 1/2^s + 1/2^r$.

Sei folgende Klasse von Hashfunktionen gegeben:

$$h_{a,b,p}(x) = [(ax+b) \mod p] \mod 2^s \qquad p \text{ Primzahl}, \ p \geq 2^r \qquad a, b \in [0, p-1]$$

Für beliebige x und zufällige a, b-Werte ist $[(ax+b) \mod p]$ gleichverteilt in $[0, p-1]$, da $GF(p)$ ein Körper ist. Diese Gleichverteilung (und Universalität) geht durch Beschränkung auf die letzten s Bits verloren.

Recycling von Zufallszahlen: Universalität von Hashfunktionen

Behauptung 1
Die hier definierte Hashklasse ist fast universal.

Beweis:

$$\begin{aligned}
P[h(x) = h(y)] &= \sum_{z \in \{0,1\}^s} P[h(x) = z \wedge h(y) = z] \\
&= \sum_{u \in [0,p-1]} \sum_{\substack{u' \in [0, p-1] \\ u' \equiv u \mod 2^s}} P[(ax+b) \mod p = u \wedge (ay+b) \mod p = u'] \\
&\leq p \cdot \lceil p/2^s \rceil \cdot \frac{1}{p^2} \\
&\leq (p/2^s + 1) \cdot \frac{1}{p} \\
&= 1/2^s + 1/p \leq 1/2^s + 1/2^r
\end{aligned}$$

Definition 8 (Ähnlichkeit von Verteilungen)
Seien D, D' Wahrscheinlichkeitsverteilungen auf einer endlichen Menge S. D und D' heißen ähnlich bis auf eine Schranke ϵ, wenn für jede Teilmenge $X \subseteq S$ gilt: $|D(X) - D'(X)| \leq \epsilon$ mit $D(x) = \sum_{x \in X} P_D(x)$.

Recycling von Zufallszahlen: Leftover Hash Lemma

LEMMA 1 (LEFTOVER HASH LEMMA) *Sei X eine beliebige Teilmenge von $\{0,1\}^r$ mit $|X| \geq 2^l$. Sei H eine fast universelle Klasse von Hashfunktionen von $\{0,1\}^r$ nach $\{0,1\}^s$. Es gelte $r \geq l > s$ und es sei $e = (l-s)/2$. Dann ist die Verteilung von $(h, h(x))$ fast gleichverteilt (auf der Menge $H \times \{0,1\}^s$) bis auf die Schranke $1/2^e$.*

BEWEIS 1
1. Die Kollisionswahrscheinlichkeit der Verteilung $(h, h(x))$ ist höchstens $\frac{1+2/2^{2e}}{|H|2^s}$.

$$
\begin{aligned}
P[(h_1, h_1(x_1)) = (h_2, h_2(x_2))] &= P[h_1 = h_2 \wedge h_1(x_1) = h_2(x_2)] \\
&= \frac{1}{|H|} P[h_1(x_1) = h_2(x_2)] \\
P[h_1(x_1) = h_2(x_2)] &= P(x_1 = x_2) + P[h_1(x_1) = h_2(x_2) | x_1 \neq x_2] \\
&= 1/2^l + 1/2^s + 1/2^r \\
P[(h_1, h_1(x_1)) = (h_2, h_2(x_2))] &= \frac{1}{|H|2^s}(1/2^{l-s} + 1 + 1/2^{r-s}) \\
P[(h_1, h_1(x_1)) = (h_2, h_2(x_2))] &\leq \frac{1}{|H|2^s}(1 + 2/2^{l-s}) = \frac{1}{|H|2^s}(1 + 2/2^{2e})
\end{aligned}
$$

2. Wenn eine Verteilung D auf einer endlichen Menge S eine Kollisionswahrscheinlichkeit von höchstens $(1 + 4\delta^2)/|S|$ hat, dann ist D fast gleichverteilt mit der Schranke δ.

Recycling von Zufallszahlen: Leftover Hash Lemma

Beweis 2 (Indirekter Beweis)
Angenommen, D ist nicht *fast gleichverteilt mit der Schranke δ – dann gibt es eine Teilmenge Y von S mit $D(Y) > |Y|/|S| + \delta$. Sei $\beta > \delta$, so daß $D(Y) = |Y|/|S| + \beta$. Wir zerlegen den Wahrscheinlichkeitsraum an Hand von Y: Die Wahrscheinlichkeit für $d_1 = d_2$, unter der Bedingung $d_1, d_2 \in Y$ ist mindestens $1/|Y|$ – und die Wahrscheinlichkeit für $d_1 = d_2$ unter der Bedingung $d_1, d_2 \notin Y$ ist mindestens $1/(|S| - |Y|)$. Deshalb ist die Kollisionswahrscheinlichkeit mindestens:*

$$\begin{aligned}
\frac{D(Y)^2}{|Y|} + \frac{(1-D(Y))^2}{|S|-|Y|} &= \frac{1}{|S|} + \frac{\beta^2}{|Y|} + \frac{\beta^2}{|S|-|Y|} \\
&= \frac{1}{|S|} + \frac{\beta^2|S|}{|Y|\cdot|S|-|Y|} \\
\min_{|Y|}(\ldots) &= \frac{1+4\beta^2}{|S|} \quad \textit{für} \quad |Y| = |S|/2
\end{aligned}$$

Es ergibt sich eine Kollisionswahrscheinlichkeit von $(1+4\beta^2)/|S| > (1+4\delta^2)/|S|$, was im Widerspruch zur Annahme oben steht. $\square$

Recycling von Zufallszahlen

Frage: Inwiefern ändert die Verteilung der Ergebnisse eines probabilistischen Algorithmen, wenn statt richtigen Zufallszahlen „recycelte Zufallszahlen" genutzt werden ?

Sei M ein probabilistischer Algorithmus für eine Sprache L vom BPP-Typ, der zur Verarbeitung einer Eingabe genau r Zufallsbits benötigt. Es gelte:

$$x \in L \Rightarrow 2/3 \leq P[M(x) = 1] \leq 3/4$$
$$x \notin L \Rightarrow 1/4 \leq P[M(x) = 1] \leq 1/3$$

Für eine feste Eingabe $x_0 \in L$ sei $P[M(x_0) = 1] = p,\ (2/3 \leq p \leq 3/4)$ gegeben. Weiterhin seinen $z_1, z_2, \ldots, z_k$ „recycelte Zufallszahlen". Es soll untersucht werden, ob die Verteilung der Bitfolge $b(z_1), b(z_2), \ldots, b(z_k)$ mit

$$b(z_i) = \begin{cases} 1 & \text{wenn M bei Zufallszahl } z_i \text{ akzeptiert} \\ 0 & \text{wenn M bei Zufallszahl } z_i \text{ verwirft} \end{cases}$$

annähernd der Verteilung $b_1, b_2, \ldots, b_k$ gleichkommt, die das Ergebnis der k-maligen Simulation Algorithmus M auf derselben Eingabe x_0 mit „richtigen Zufallszahlen" ist.

Behauptung 2 (Ähnlichkeit der Verteilungen der Ergebnisse)
Für alle $i \geq 0$ ähnelt die Verteilung von $b(z_1) \ldots b(z_i)\, h\, z_{i+1}$ (z_i = „recycelte Zufallszahlen") der folgenden Verteilung bis auf eine Schranke ϵ_i: Die ersten i Bits sind mit einer Wahrscheinlichkeit von p als „1" und mit einer Wahrscheinlichkeit von $1 - p$ als „0" gewählt, h wird unter Gleichverteilung aus H gezogen, dann folgt ein Zufallswort, daß unter Gleichverteilung aus $\{0, 1\}^r$ gezogen wird.

Recycling von Zufallszahlen

Beweis 3 (Vollständige Induktion)
Im Fall $i = 0$ ist $h\, z_1$ echt gleichverteilt, also ist $\epsilon_0 = 0$.

Im Fall $i = 1$ ist $b(z_1)h\, z_2 = b(z_1)h\, h(z_1)$. Es ist nachzuweisen, daß dessen Verteilung folgender ähnelt: Das erste Bit wird aus einer $(p, 1-p)$-Verteilung gezogen, h zufällig aus H und die letzten s Bits zufällig aus $\{0,1\}^s$.

Da z_1 zufällig ist, ist $b(z_1)$ gemäß $(p, 1-p)$ verteilt. Unter der Bedingung, daß $b(z_1) = a = const, a \in \{0,1\}$ ist zu zeigen, daß $h\, h(z_1)$ annähernd auf $H \times \{0,1\}^s$ gleichverteilt ist - durch das Leftover Hash Lemma. Es stehen noch $p \cdot 2^r$ bzw. $(1-p) \cdot 2^r$, also mindestens $2^r/4$ Wahlmöglichkeiten zur Verfügung, also ist $l = r - 2$. Damit ergibt sich:

$$\epsilon_1 = 2^{-e} = 2^{-(l-s)/2} = 2^{-(r-s-2)/2}$$

Induktionsschritt: Nach Induktionsvoraussetzung gelte die Behauptung für $i-1$, d.h. die Verteilung $b(z_1)\ldots b(z_{i-1})h\, z_i$ ähnelt einer konkreten Verteilung bis auf die Schranke ϵ_{i-1}. Es ist zu beweisen, daß $b(z_1)\ldots b(z_i)h\, z_{i+1} = b(z_1)\ldots b(z_i)h\, h(z_i)$ der Verteilung $b_1 \ldots b_i h\, v$ ähnelt, wobei die b_j $(p, 1-p)$-verteilt sind und $h\, v$ gleichverteilt auf $H \times \{0,1\}^s$.

Hierbei hilft folgender Satz:

Recycling von Zufallszahlen

Lemma 2 *Seien F_1, F_2, H Zufallsvariablen, so daß die Verteilung von F_1 derjenigen von F_2 bis auf eine Schranke δ_1 ähnelt. Ferner sei t eine Transformation, so daß Verteilung von $t(F_2)$ derjenigen von H bis auf eine Schranke δ_2 ähnelt. Dann ähneln sich die Verteilungen von $t(F_1)$ und H bis auf die Schranke $\delta_1 + \delta_2$.*

$$\begin{tikzcd}
t(F_1) \arrow[r, dash, dashed] & H \arrow[r, dash, dashed, "\delta_2"] & t(F_2) \\
F_1 \arrow[u] \arrow[rr, dash, dashed, "\delta_1"] & & F_2 \arrow[u]
\end{tikzcd}$$

$$\begin{aligned}
F_1 &= b(z_1)\cdots b(z_{i-1})\,h\,z_i \\
F_2 &= b_1 \cdots b_{i-1}\,h\,u \\
H &= b_1 \cdots b_{i-1} b_i\,h\,v \\
t(b_1 \cdots b_{i-1}\,h\,u) &= b_1 \cdots b_{i-1} b(u)\,h\,h(u)
\end{aligned}$$

Damit ergibt sich:

$$\begin{aligned}
t(F_1) &= b(z_1)\cdots b(z_{i-1}) b(z_i)\,h\,h(z_i) \\
t(F_2) &= b_1 \cdots b_{i-1} b(u)\,h\,h(u)
\end{aligned}$$

F_1 und F_2 sind sich nach Induktionsvoraussetzung bis auf Die Schranke ϵ_{i-1} ähnlich. Ferner sind sich $t(F_2)$ und H bis auf die Schranke ϵ_1 ähnlich. Es folgt, daß sich H und F_1 bis auf $\epsilon_{i-1} + \epsilon_1$ ähnlich sind, d.h. $\epsilon_i = \epsilon_{i-1} + \epsilon_1 = i\epsilon_1$. □

Recycling von Zufallszahlen

Berechnung von ϵ_k:

$$
\begin{aligned}
\epsilon_1 &= 2^{-e} = 2^{-(l-s)/2} = 2^{-(r-s-2)/2} \\
\epsilon_k &= k\epsilon_1 = k2^{-(r-s-2)/2} \\
\epsilon_k &= k2^{-k} \quad \text{wenn } s \text{ so gewählt, daß} \quad k = (r-s-2)/2 \\
\epsilon_k &= k2^{-k} = 2^{-(k - \operatorname{ld} k)} = 2^{-\Omega(k)}
\end{aligned}
$$

Behauptung 3 (Fehlerschranke bei „recycelten Zufallszahlen“)
Bei Verwenden von echten Zufallszahlen hat der wahrscheinlichkeitsverstärkte BPP-Algorithmus eine Fehlerwahrscheinlichkeit von $2^{-\Omega(k)}$. Ähnelt die Verteilung der Ergebnisse bei Verwendung von „recycelten Zufallszahlen“ der Verteilung der Ergebnisse bei echten Zufallszahlen bis auf die Schranke $\epsilon_k = 2^{-\Omega(k)}$, so ist der Fehler hier ebenfalls nur $2^{-\Omega(k)}$.

Beweis 4
Sei X die Menge der 0-1-Folgen mit nicht mehr als $k/2$ Einsen in der Folge. Sie D eine Verteilung, die der $(p, 1-p)$-Verteilung der Länge k ähnlich ist bis auf die Schranke ϵ_k und sei E die „korrekte“ Verteilung.

$$
\begin{aligned}
P_D(X) &\leq P_E(X) + \epsilon_k \\
&= \sum_{i=0}^{k/2} \binom{k}{i} p^i \cdot (1-p)^{t-i} + \epsilon_k \leq 2^{-\Omega(k)} + \epsilon_k
\end{aligned}
$$

Wegen $\epsilon_k = 2^{-\Omega(k)}$ gilt obige Behauptung. □