

# Grundbegriffe der Theoretischen Informatik

Sommersemester 2018 - Thomas Schwentick

Teil D: Komplexitätstheorie

22: Zufallsbasierte Komplexitätsklassen

Version von: 12. Juli 2018 (13:58)

# Zufallsbasierte Komplexitätsklassen: Grundlagen (1/3)

- In diesem Kapitel betrachten wir zufallsbasierte Algorithmen aus der Sicht der Komplexitätstheorie
- Wir definieren Komplexitätsklassen, die Probleme enthalten, die sich effizient mit zufallsbasierten Algorithmen lösen lassen
  - Die Laufzeit wird also (fast) immer polynomiell sein
- Durch verschiedene Anforderungen an die Akzeptier-Wahrscheinlichkeiten werden sich unterschiedliche Klassen ergeben
- Wir betrachten die folgenden Varianten zufallsbasierter Algorithmen:
  - einseitiger Fehler oder zweiseitiger Fehler
  - möglicherweise großer Fehler ( $< \frac{1}{2}$ ) oder kleiner Fehler ( $\leq \frac{1}{4}$ )
- Die Algorithmen für PRIMES und ZEROCIRC haben einseitigen, kleinen Fehler  
(bei genügend häufiger Wiederholung)

## Zufallsbasierte Komplexitätsklassen: Grundlagen (2/3)

- Die erste Frage, die wir beantworten müssen:
  - Wie modellieren wir zufallsbasierte Algorithmen durch Turingmaschinen?

- Zur Beantwortung gehen wir ähnlich vor wie bei der Definition des nicht-deterministischen Akzeptierens

- Wir betrachten Berechnungen einer TM mit Eingabe  $x$  und *Zusatzeingabe*  $y$ , wobei die Zusatzeingabe nur polynomiell lang in  $|x|$  sein darf

- Die Zusatzeingabe repräsentiert die Zufallsbits

- Wir sagen  $M(x, y)$  *akzeptiert*, wenn  $M$  bei Eingabe  $x$  und Zusatzeingabe  $y$  akzeptiert

- Der Einfachheit halber betrachten wir nur Eingaben und Zusatzeingaben über dem Alphabet  $\Sigma = \{0, 1\}$

- Die relative Häufigkeit der Zusatzeingaben  $y$ , die zum Akzeptieren führen, im Verhältnis zu allen Zusatzeingaben, definiert dann gerade die Akzeptierwahrscheinlichkeit:

**Definition (Akzeptierwahrscheinlichkeit,  $p_M$ )**

- Sei  $M$  eine TM mit Zeitschranke  $T$  und  $x \in \Sigma^*$
- Dann ist die Akzeptierwahrscheinlichkeit  $p_M(x)$  von  $M$  bei Eingabe  $x$  definiert durch
$$\frac{|\{y \in \Sigma^{T(|x|)} \mid M(x, y) \text{ akzeptiert}\}|}{2^{T(|x|)}}$$

- Die Zeitschranke hängt dabei wieder nur von  $|x|$  ab, es muss also gelten:
$$t_M((x, y)) \leq T(|x|)$$

# Zufallsbasierte Komplexitätsklassen: Grundlagen (3/3)

- Zur Erinnerung: ein zufallsbasierter  $(f(n), g(n))$ -Algorithmus für eine Sprache  $L$  hat für Eingaben  $x \in L$  Fehlerwahrscheinlichkeit  $\leq f(|x|)$  und für  $x \notin L$  Fehlerwahrscheinlichkeit  $\leq g(|x|)$
- Beobachtung: Zu jeder Sprache gibt es
  - einen polynomiellen  $(\frac{1}{2}, \frac{1}{2})$ -Algorithmus:
    - \* Wähle zufällig und gleichverteilt ein Bit  $b \in \{0, 1\}$  und akzeptiere falls  $b = 1$
  - einen polynomiellen  $(0, 1)$ -Algorithmus:
    - \* Akzeptiere immer
  - einen polynomiellen  $(1, 0)$ -Algorithmus:
    - \* Lehne immer ab
  - einen polynomiellen  $(p, q)$ -Algorithmus, falls  $p + q = 1$  und es für jedes  $n$  ein  $k$  mit  $p = \frac{k}{2^{T(n)}}$  gibt
- Interessant sind also überhaupt nur Klassen, für die die Summe  $p + q$  der beiden Fehler-W-keiten kleiner als 1 ist

# Inhalt

- ▷ **22.1 Komplexitätsklassen mit einseitigem Fehler**
- 22.2 Komplexitätsklassen mit kleinem zweiseitigen Fehler
- 22.3 Komplexitätsklassen mit großem zweiseitigen Fehler
- 22.4 Komplexitätsklassen-Übersicht

# Zufallsbasierte Komplexitätsklassen: RP und co-RP (1/2)

## Definition (RP, co-RP)

- **RP**  $\stackrel{\text{def}}{=}$  Klasse der Mengen  $L$  mit Poly-Zeit-TM  $M$ , so dass
  - $x \in L \Rightarrow p_M(x) \geq \frac{1}{2}$
  - $x \notin L \Rightarrow p_M(x) = 0$


- **co-RP**  $\stackrel{\text{def}}{=}$  Klasse der Mengen  $L$  mit Poly-Zeit-TM  $M$ , so dass
  - $x \in L \Rightarrow p_M(x) = 1$
  - $x \notin L \Rightarrow p_M(x) \leq \frac{1}{2}$

- Zu beachten: Diese Definitionen verwenden Akzeptierwahrscheinlichkeiten, keine Fehlerwahrscheinlichkeiten

- **RP** umfasst die Probleme mit polynomialen  $(\frac{1}{2}, 0)$ -Algorithmen
- **co-RP** umfasst die Probleme mit polynomialen  $(0, \frac{1}{2})$ -Algorithmen

- Die Algorithmen für PRIMES und ZEROCIRC belegen:

- PRIMES  $\in$  **co-RP**

 Aber, wie wir wissen, gilt sogar:  
PRIMES  $\in$  **P**

- ZEROCIRC  $\in$  **co-RP**

- Nach Definition gilt: **RP**  $\subseteq$  **NP**

## Zufallsbasierte Komplexitätsklassen: RP und co-RP (2/2)

- Die bei den Algorithmen für PRIMES und ZEROCIRC verwendete Technik der *Wahrscheinlichkeitsverstärkung* lässt sich für **RP** und **co-RP** verallgemeinern

### Satz 22.1

- Sei  $L \in \mathbf{RP}$ ,  $k \in \mathbb{N}$
- Dann gibt es eine polynomiell zeitbeschränkte TM  $M$ , so dass
  - $x \in L \Rightarrow p_M(x) \geq 1 - \frac{1}{2^{|x|^k}}$
  - $x \notin L \Rightarrow p_M(x) = 0$

### Beweisidee

- Sei  $M'$  eine der Definition von **RP** entsprechende TM für  $L$
- Die TM  $M$  simuliert  $M'$  hintereinander  $|x|^k$  mal
  - $M$  erwartet eine Zusatzeingabe der Form  $y_1 \cdot \dots \cdot y_{|x|^k}$  und verwendet den String  $y_i$  als Zusatzeingabe für die  $i$ -te Simulation von  $M'$
- $M$  akzeptiert genau dann, wenn mindestens eine dieser Simulationen zum Akzeptieren führt
- Die W-keit, dass  $M'$  für alle diese Zusatzeingaben ablehnt, ist im Falle  $x \in L$  höchstens  $\frac{1}{2^{|x|^k}}$
- Und:  $M$  ist polynomiell zeitbeschränkt
- Ein analoges Resultat gilt für **co-RP**

# Zufallsbasierte Komplexitätsklassen: ZPP (1/5)

## Definition (ZPP)

- $\text{ZPP} \stackrel{\text{def}}{=} \text{RP} \cap \text{co-RP}$

- Probleme in **ZPP** haben also einen polynomiellen  $(\frac{1}{2}, 0)$ -Algorithmus und einen polynomiellen  $(0, \frac{1}{2})$ -Algorithmus
- Durch Kombination dieser beiden Algorithmen lassen sich neue Algorithmen mit sehr günstigen Eigenschaften konstruieren
- Erster Algorithmientyp für **ZPP**-Probleme:
  - polynomielle Laufzeit
  - *Drei Antwortmöglichkeiten:*  
„ja“, „nein“, „weiß-nicht“
  - „ja“, „nein“-Antworten immer richtig
- Zweiter Algorithmientyp für **ZPP**-Probleme:
  - Zwei Antwortmöglichkeiten: „ja“, „nein“
  - Antworten immer richtig
  - im *Durchschnitt* polynomielle Laufzeit



## Zufallsbasierte Komplexitätsklassen: ZPP (2/5)

- Für den ersten Typ definieren wir das folgende TM-Modell

### Definition (Las-Vegas-TM)


- Eine Las-Vegas-TM für eine Sprache  $L$  hat folgende Eigenschaften:
  - Sie hat außer „ja“ und „nein“ einen weiteren Endzustand „weiß-nicht“
  - \* Für  $x \in L$  endet jede Berechnung (für jede Zusatzeingabe  $y$ ) in „ja“ oder „weiß-nicht“
  - \* Für  $x \notin L$  endet jede Berechnung in „nein“ oder „weiß-nicht“
  - Die Antwort „ja“ oder „nein“ ist also immer richtig
  - Die W-keit, dass  $M$  bei Eingabe  $x$  im Zustand „weiß-nicht“ endet, bezeichnen wir mit  $p_{M,?}(x)$

- Für den zweiten Typ betrachten wir eine andere Art von „Zeitschranke“

### Definition (Polynomiell erwartete Laufzeit)

- Eine TM  $M$  entscheidet eine Sprache  $L$  mit polynomiell erwarteter Laufzeit, falls es  $c, d$  gibt, so dass für alle  $x$  gelten:
  - Für jede Zusatzeingabe  $y$  der Länge  $|x|^c$  gibt  $M$  die richtige Antwort („ja“, falls  $x \in L$ , „nein“, falls  $x \notin L$ )

$$- \frac{1}{2^{|x|^c}} \sum_{y \in \Sigma^{|x|^c}} t_M(x, y) \leq |x|^d$$

 Die durchschnittliche Laufzeit (gemittelt über die Zusatzeingaben  $y$ ) ist also polynomiell beschränkt

- Zu beachten:
  - Die Laufzeit von  $M$  kann für einzelne Zusatzeingaben größer als  $|x|^d$  sein

# Zufallsbasierte Komplexitätsklassen: ZPP (3/5)

## Satz 22.2

- Für eine Sprache  $L$  sind äquivalent:
  - (a)  $L \in \mathbf{ZPP}$
  - (b) Es gibt für  $L$  eine Las-Vegas-TM  $M_1$  mit Poly-Laufzeit, so dass für alle  $x$  gilt:  $p_{M,?}(x) \leq \frac{1}{2}$
  - (c) Es gibt für  $L$  eine zufallsbasierte TM  $M_2$  mit polynomiell erwarteter Laufzeit

## Beweisskizze „(a) $\Rightarrow$ (b)“

- Sei  $L \in \mathbf{ZPP}$
- Sei  $A^+$  ein  $(\frac{1}{2}, 0)$ -Algorithmus für  $L$  (**RP**) und  $A^-$  ein  $(0, \frac{1}{2})$ -Algorithmus für  $L$  (**co-RP**)
- Sei  $A$  folgender Algorithmus (bei Eingabe  $x$ ):
  - Simuliere  $A^+$  bei Eingabe  $x$
  - Falls  $A^+$  akzeptiert, Ausgabe „ja“
  - Simuliere  $A^-$  bei Eingabe  $x$
  - Falls  $A^-$  ablehnt, Ausgabe „nein“
  - Andernfalls Ausgabe: „weiß-nicht“

## Beweisskizze (Forts.)

- $A^+$  akzeptiert nur, falls  $x \in L$ 
  - ➔ die Ausgabe „ja“ von  $A$  ist immer richtig
- Analog:  $A^-$  lehnt nur ab, falls  $x \notin L$ 
  - ➔ die Ausgabe „nein“ von  $A$  ist immer richtig
- Schließlich:
  - Falls  $x \in L$ , gibt  $A^+$  die Antwort „ja“ mit W-keit  $\geq \frac{1}{2}$ 
    - ➔  $p_{M,?}(x) \leq \frac{1}{2}$
  - Falls  $x \notin L$ , gibt  $A^-$  die Antwort „nein“ mit W-keit  $\geq \frac{1}{2}$ 
    - ➔  $p_{M,?}(x) \leq \frac{1}{2}$
- ➔ Aus  $A$  lässt sich eine Las Vegas-TM  $M_1$  wie in (b) konstruieren

# Zufallsbasierte Komplexitätsklassen: ZPP (4/5)

## Beweisskizze „(b) $\Rightarrow$ (c)“

- Sei  $M_1$  Las-Vegas-TM für  $L$  gemäß (b) mit Zeitschranke  $n^j$
- Sei ferner  $M$  eine TM, die  $L$  in Zeit  $2^{n^k}$  entscheidet, für ein  $k \in \mathbb{N}$   
👉 **ZPP  $\subseteq$  RP  $\subseteq$  NP  $\subseteq$  EXPTIME**
  - Idee für  $M$ : Simuliere  $M_1(x, y)$ , für alle Zusatzeingaben  $y$  der Länge  $|x|^j$
- $M_2$  arbeitet bei Eingabe  $x$  wie folgt:
  - Simuliere  $|x|^k$  mal  $M_1$
  - Falls eine dieser Simulationen „ja“ ausgibt, so akzeptiere
  - Falls eine dieser Simulationen „nein“ ausgibt, so lehne ab
  - Falls alle Simulationen „weiß-nicht“ ausgeben, simuliere  $M$  bei Eingabe  $x$ , und gib die Antwort, die  $M$  geben würde

## Beweisskizze (Forts.)

- Klar:  $M_2$  terminiert immer und gibt immer die korrekte Antwort
- Die W-keit, dass alle Simulationen von  $M_1$  die Antwort „weiß-nicht“ haben ist  $\leq \frac{1}{2^{|x|^k}}$   
➡ Also ist die erwartete Laufzeit  $\leq |x|^k |x|^j + \frac{1}{2^{|x|^k}} 2^{|x|^k} = |x|^{j+k} + 1$

# Zufallsbasierte Komplexitätsklassen: ZPP (5/5)

## Beweisskizze „(c) $\Rightarrow$ (a)“

- Sei  $M_2$  eine TM für  $L$  mit polynomieller erwarteter Laufzeit
- Seien  $c, d$  so gewählt, dass für jedes  $x$  gilt:
  - Für jede Zusatzeingabe  $y$  der Länge  $|x|^c$  gibt  $M_2$  die richtige Antwort  
☞ „ja“, falls  $x \in L$ , „nein“, falls  $x \notin L$
  - $$\frac{1}{2^{|x|^c}} \sum_{y \in \Sigma^{|x|^c}} t_{M_2}(x, y) \leq |x|^d$$
- Da die mittlere Laufzeit  $\leq |x|^d$  ist, ist die W-keit kleiner als  $\frac{1}{2}$ , dass für ein zufällig gewähltes  $y$  die Laufzeit größer als  $2|x|^d$  ist

## Beweisskizze (Forts.)

- Wir konstruieren eine TM  $M^+$  zum Nachweis, dass  $L \in \mathbf{RP}$
  - $M^+$  arbeitet wie folgt (bei Eingabe  $x$ ):
    - Simuliere  $M_2$  bei Eingabe  $x$  für  $2|x|^d$  Schritte
    - Akzeptiere, falls  $M_2$  in dieser Zeit akzeptiert
    - Andernfalls lehne ab
  - Da  $M_2$  bei jeder Zusatzeingabe die richtige Ausgabe hat, und die W-keit, dass  $M_2$  in Zeit  $2|x|^d$  anhält,  $\geq \frac{1}{2}$  ist, gilt:
    - Falls  $x \in L$  ist  $p_{M^+}(x) \geq \frac{1}{2}$
    - Falls  $x \notin L$  ist  $p_{M^+}(x) = 0$
- ➡  $L \in \mathbf{RP}$
- Die Konstruktion einer TM  $M^-$  zum Nachweis, dass  $L \in \mathbf{co-RP}$ , ist völlig analog

# Inhalt

22.1 Komplexitätsklassen mit einseitigem Fehler

▷ **22.2 Komplexitätsklassen mit kleinem zweiseitigen Fehler**

22.3 Komplexitätsklassen mit großem zweiseitigen Fehler

22.4 Komplexitätsklassen-Übersicht

# Probabilistische Klassen: kleiner, zweiseitiger Fehler

## Definition (BPP)

- **BPP** sei die Klasse der Mengen  $L$ , für die es eine polynomiell zeitbeschränkte TM  $M$  gibt, so dass:

$$\begin{aligned} - x \in L &\Rightarrow p_M(x) \geq \frac{3}{4} \\ - x \notin L &\Rightarrow p_M(x) \leq \frac{1}{4} \end{aligned}$$

- **BPP** umfasst also alle Probleme, die einen polynomiellen  $(\frac{1}{4}, \frac{1}{4})$ -Algorithmus haben
- Im Falle von **RP** und **coRP** lässt sich daraus, dass zwei Berechnungen für eine Eingabe  $x$  einmal „ja“ und einmal „nein“ ergeben, jeweils ein eindeutiger Schluss ziehen
  - Das war dort die Grundlage für die Wahrscheinlichkeitsverstärkung
- Für **BPP** können wir nicht so vorgehen, da bei einer „**BPP**-TM“ vorkommen kann,
  - dass sie für  $x \in L$  „nein“ sagt, und
  - dass sie für  $x \notin L$  „ja“ sagt

- Aber auch hier lässt sich auf einfache Weise eine W-Verstärkung erreichen:
  - Wiederhole den Algorithmus (mit mehreren Zusatzeingaben) und akzeptiere genau dann, wenn die *Mehrheit der Berechnungen* akzeptierend ist

## Satz 22.3

- Ist  $L \in \mathbf{BPP}$ ,  $k \in \mathbb{N}$ , so gibt es eine polynomiell zeitbeschränkte TM  $M$ , so dass gilt:
  - (a)  $x \in L \Rightarrow p_M(x) \geq 1 - \frac{1}{2^{|x|^k}}$
  - (b)  $x \notin L \Rightarrow p_M(x) \leq \frac{1}{2^{|x|^k}}$
- Die Fehlerwahrscheinlichkeit kann also nicht nur (in Poly-Zeit) unter jede beliebige feste Zahl  $\epsilon > 0$  gesenkt werden
- Sondern sie kann sich für große  $n$  exponentiell schnell an 0 annähern

# Ein hilfreiches Resultat aus der Wahrscheinlichkeitstheorie

- Um Satz 22.3 zu beweisen brauchen wir etwas Wahrscheinlichkeitstheorie
- Wir hatten eben schon verwendet, dass für Zufallsvariable  $X$ , die keine negativen Werte annehmen, und den Erwartungswert  $E(X) = p$  haben, gilt:
  - Die Wahrscheinlichkeit, dass der Wert von  $X$  größer als  $2p = 2E(X)$  ist, ist kleiner als  $\frac{1}{2}$ :  
\*  $P(X \geq 2p) \leq \frac{1}{2}$
  - Zum Beweis von Satz 22.3 benötigen wir jedoch eine bessere Abschätzung, die uns das folgende Lemma liefert

## Lemma 22.4 [Chernoff-Schranke]

- Seien  $X_1, \dots, X_n$  unabhängige, 0-1-wertige Zufallsvariable mit  $P(X_i = 1) \leq p$  für alle  $i$

- Sei  $X \stackrel{\text{def}}{=} \sum_{i=1}^n X_i$

- Dann gilt für alle  $\theta$  mit  $0 \leq \theta \leq 1$ :

$$P(X \geq (1 + \theta)pn) \leq e^{-\frac{\theta^2}{3}pn}$$

- Die W-Keit, dass in 100 Münzwürfen öfter als 75 mal „Kopf“ kommt, ist z.B. kleiner als 0,02:
  - $n = 100, p = \frac{1}{2}, \theta = 0,5$
- Wir verwenden Lemma 22.4 für  $\theta = 1$  und  $p = \frac{1}{4}$  und erhalten:
  - Wenn bei  $n$  Experimenten jeweils mit W-Keit  $\leq \frac{1}{4}$  das Ergebnis 1 und mit W-keit  $\geq \frac{3}{4}$  das Ergebnis 0 ist,
  - dann ist die W-keit, dass die Summe der Ergebnisse größer als  $n/2$  ist, höchstens  $e^{-\frac{1}{12}n}$

# Wahrscheinlichkeitsverstärkung für BPP

## Beweisskizze von Satz 22.3

- Sei  $L \in \mathbf{BPP}$  und sei  $M$  eine TM mit Zeitschranke  $n^l$ , für die gilt:
  - $x \in L \Rightarrow p_M(x) \geq \frac{3}{4}$
  - $x \notin L \Rightarrow p_M(x) \leq \frac{1}{4}$
- Wir konstruieren eine TM  $M'$  mit Zeitschranke  $\sim 24n^{k+l}$
- $M'$  arbeitet wie folgt (bei Eingabe  $x$ ):
  - $M'$  simuliert  $24|x|^k$  mal  $M$  bei Eingabe  $x$  (und  $24|x|^k$  Zusatzeingaben) und zählt die Anzahl  $m$  der akzeptierenden Berechnungen
  - $M'$  hat Ausgabe „ja“, falls
$$m \geq 12|x|^k$$
  - Andernfalls Ausgabe „nein“

## Beweisskizze (Forts.)

- Klar:  $M'$  hat polynomielle Laufzeit
- Wir zeigen nun: (b):
$$x \notin L \Rightarrow p_{M'}(x) \leq \frac{1}{2^{|x|^k}}$$
- Sei dazu  $x \notin L$
- Für  $i \leq 24|x|^k$  sei  $X_i$  die Zufallsvariable mit Wert
  - 1, falls die  $i$ -te Simulation akzeptiert
  - 0, falls die  $i$ -te Simulation ablehnt
- Also:  $P(X_i = 1) \leq \frac{1}{4}$ , für alle  $i$
- Sei  $X \stackrel{\text{def}}{=} \sum_{i=1}^{24|x|^k} X_i$  und  $\theta = 1$
- ➔  $P(X \geq 12|x|^k) \leq e^{-2|x|^k} \leq 2^{-|x|^k}$
- (a) kann analog gezeigt werden

➔ Behauptung



# Zufallsbasierte Algorithmen für 3-SAT: Grenzen

- Interessante Frage: gibt es für 3-SAT einen zufallsbasierten Algorithmus mit polynomieller Laufzeit?
- Als Konsequenz ergäbe sich:
  - **NP**  $\subseteq$  **BPP** oder sogar
  - **NP**  $\subseteq$  **RP**
- Das wird als unwahrscheinlich erachtet, insbesondere angesichts der verbreiteten Vermutung, dass **BPP** = **P** sein könnte:
  - Denn dann würde **NP** = **P** folgen
- Trotzdem sind Algorithmen wie der von Schönig äußerst nützlich, wie bereits im letzten Kapitel besprochen

# Inhalt

22.1 Komplexitätsklassen mit einseitigem Fehler

22.2 Komplexitätsklassen mit kleinem zweiseitigen Fehler

▷ **22.3 Komplexitätsklassen mit großem zweiseitigen Fehler**

22.4 Komplexitätsklassen-Übersicht

# Probabilistische Klassen: großer, zweiseitiger Fehler

## Definition (PP)

- **PP** sei die Klasse der Mengen  $L$ , für die es eine polynomiell zeitbeschränkte TM  $M$  gibt, so dass:
  - $x \in L \Rightarrow p_M(x) > \frac{1}{2}$
  - $x \notin L \Rightarrow p_M(x) \leq \frac{1}{2}$

## Proposition 22.5

- (a) **ZPP**  $\subseteq$  **RP**  $\subseteq$  **BPP**  $\subseteq$  **PP**
- (b) **ZPP**  $\subseteq$  **co-RP**  $\subseteq$  **BPP**  $\subseteq$  **PP**
- (c) **NP**  $\subseteq$  **PP**

- (a,b) folgen direkt aus den Definitionen

## Beweisskizze für (c)

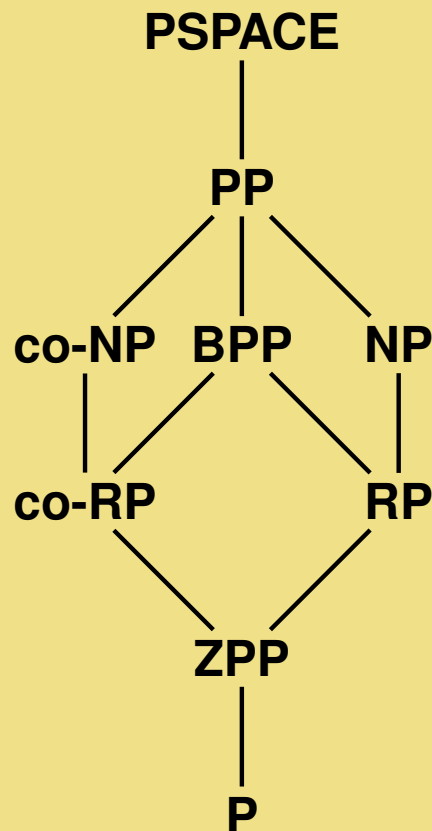
- Sei  $L \in \mathbf{NP}$  und  $M$  eine TM, die  $L$  nichtdeterministisch entscheidet
- Idee: Konstruiere TM  $M'$ , die immer mit W-keit  $\geq \frac{1}{2}$  akzeptiert


## Beweisskizze (Forts.)

- Sei  $M'$  die folgende TM (Eingabe  $x$ ):
  - Falls das erste Zeichen der Zusatzeingabe 1 ist, so akzeptiert  $M'$
  - Falls das erste Zeichen der Zusatzeingabe 0 ist, so simuliert  $M'$  die TM  $M$  bei Eingabe  $x$  mit dem Rest der Zusatzeingabe, und akzeptiert genau dann, wenn  $M$  akzeptiert
- Falls  $x \in L$  ist die W-keit, dass  $M'$  akzeptiert  $> \frac{1}{2}$ :
  - In der Hälfte aller Fälle akzeptiert  $M'$ , weil das erste Bit der Zusatzeingabe 1 ist
  - Es gibt aber auch mindestens eine Zusatzeingabe  $y$ , für die  $M(x, y)$  akzeptiert
  - ➡  $M'$  akzeptiert bei Zusatzeingabe  $0y$
  - ➡  $p_{M'}(x) > \frac{1}{2}$
- Klar: Falls  $x \notin L$ , ist  $p_{M'}(x) = \frac{1}{2}$

# Verhältnis der betrachteten Komplexitätsklassen

- Das folgende Diagramm illustriert die Inklusionsstruktur der betrachteten Klassen:



- Welche Komplexitätsklasse entspricht nun dem intuitiven Begriff des effizient berechenbaren am besten?
- P**? Ist ein Problem in **P** lösbar, wissen wir, dass wir nach polynomieller Zeit die richtige Antwort bekommen  
 und das Problem, dass Polynome untragbar groß sein können, haben wir ja schon besprochen
- ZPP**? Ist ein Problem in **ZPP** lösbar, wissen wir, dass wir immer die richtige Antwort bekommen und dies mit großer W-keit nach polynomieller Zeit passiert
- BPP**? Ist ein Problem in **BPP** lösbar, können wir zwar nicht sicher sein, dass die Antwort des Algorithmus korrekt ist, aber die Fehler-W-keit kann beliebig klein gemacht werden
- Für jede der drei Möglichkeiten gibt es gute Gründe
- Seit einigen Jahren wird von vielen vermutet, dass die Diskussion überflüssig ist, und **P** = **BPP** gilt

# Fehler-W-Keit der betrachteten Komplexitätsklassen

Klasse	max. Fehler $x \in L$	max. Fehler $x \notin L$
<b>P</b>	0	0
<b>NP</b>	$< 1$	0
<b>RP</b>	$\leq \frac{1}{2}$	0
<b>co-RP</b>	0	$\leq \frac{1}{2}$
<b>BPP</b>	$\leq \frac{1}{4}$	$\leq \frac{1}{4}$
<b>PP</b>	$< \frac{1}{2}$	$\leq \frac{1}{2}$

- Bei **RP** und **co-RP** kann  $\frac{1}{2}$  durch jede beliebige Konstante  $c$ ,  $0 < c < 1$ , ersetzt werden
- Bei **BPP** kann  $\frac{1}{4}$  durch jede Konstante  $c$ ,  $0 < c < \frac{1}{2}$  ersetzt werden
- **NP** kann also auch als eine probabilistische Komplexitätsklasse aufgefasst werden:
  - Ist  $x \in L$  wird dies mit W-keit  $> 0$  erkannt
  - Ist  $x \notin L$  wird dies mit W-keit 1 erkannt

# Inhalt

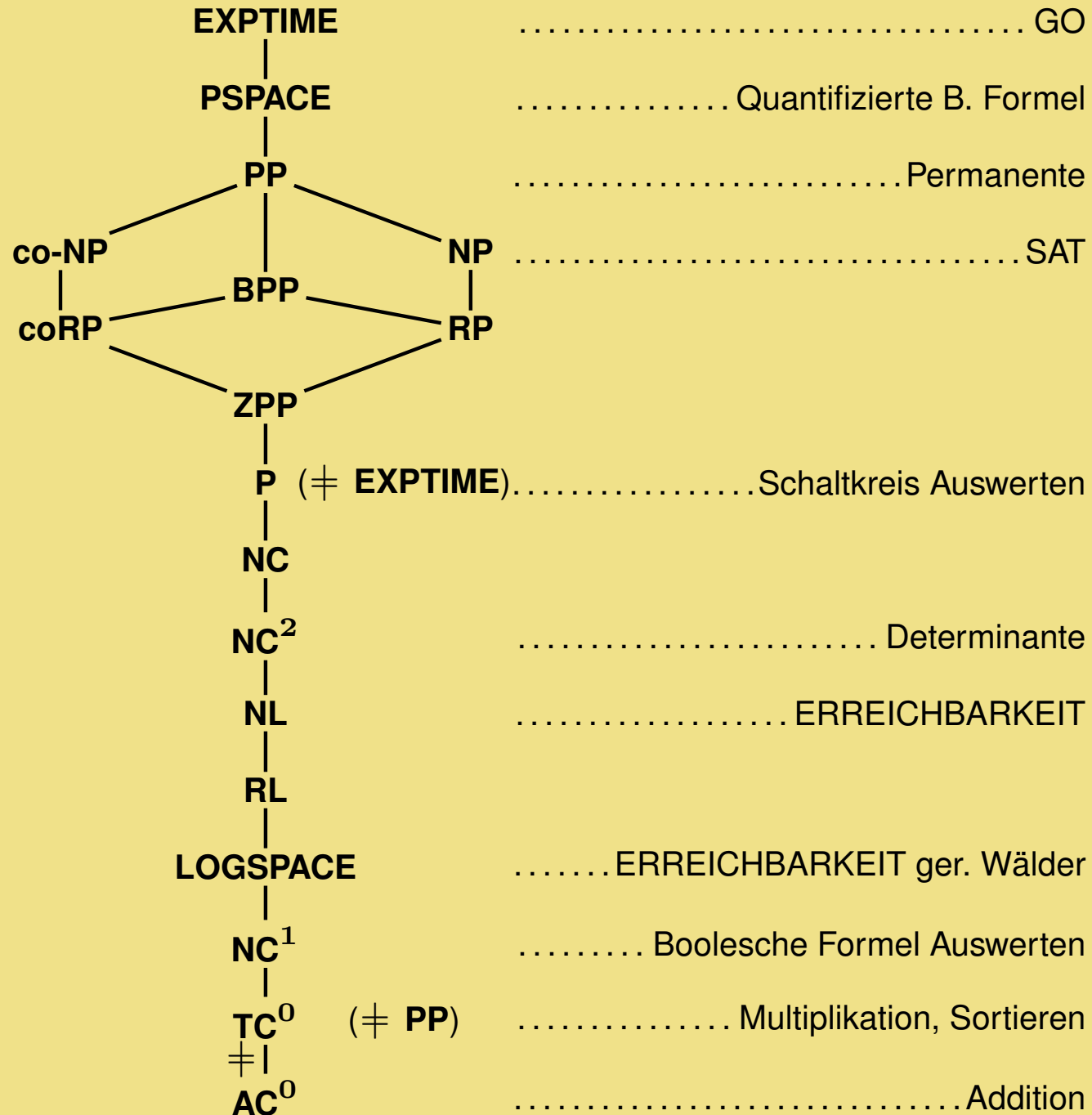
22.1 Komplexitätsklassen mit einseitigem Fehler

22.2 Komplexitätsklassen mit kleinem zweiseitigen Fehler

22.3 Komplexitätsklassen mit großem zweiseitigen Fehler

▷ **22.4 Komplexitätsklassen-Übersicht**

# Es gibt noch viel mehr Komplexitätsklassen...



# Zusammenfassung

- Es gibt Klassen mit einseitigem oder zweiseitigem Fehler, sowie kleinem oder großem Fehler
- Die Probleme in **ZPP**, **RP**, **co-RP**, **BPP** können durchaus als effizient berechenbar gelten



## Literaturhinweise

- Christos M. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994