17 Probabilistische Komplexitätsklassen

Eine *probabilistische Turing-Maschine (PTM)* arbeitet wie eine NTM mit zwei Entscheidungsalternativen pro Schritt. Der Unterschied zwischen NTMs und PTMs macht sich technisch erst bei der Definition der von M erkannten Sprache L_M bemerkbar. Eine NTM M akzeptiert einen String $x \in \Sigma^*$ (d.h. $x \in L_M$) gdw M mindestens eine akzeptierende Rechnung auf Eingabe x besitzt. Bei einer PTM hingegen werden wir i.A. fordern, dass es

- auf Eingaben $x \in L_M$ „viele“
- auf Eingaben $x \notin L_M$ „wenige“

akzeptierende Rechnungen gibt. Um die Begriffe „viele“ und „wenige“ zu quantifizieren, sehen wir die Rechnung einer PTM M als zufällig an: in jedem Schritt wird eine der beiden Handlungsalternativen mit Wahrscheinlichkeit $1/2$ gewählt. Wie schon bei NTMs gibt es bei PTMs ein „online“- und ein „offline“-Modell:

online-Nichtdeterminismus In jedem Schritt wählt M nicht-deterministisch eine von (oBdA) zwei Handlungsalternativen aus, sagen wir „Aktion 0“ oder „Aktion 1“.

online-Randomisierung In jedem Schritt bestimmt M ein perfektes Zufallsbit $b \in \{0,1\}$ (etwa durch den Wurf einer perfekten Münze) und wählt dann Aktion b.

offline-Nichtdeterminismus Für eine $T(n)$-zeitbeschränkte NTM M nehmen wir die $T(n)$ nicht-deterministischen Entscheidungen vorweg, indem wir die Eingabe x um einen (nicht-deterministisch gewählten) Ratestring $y \in \{0,1\}^{T(n)}$ ergänzen und M deterministisch auf $\langle y, x\rangle$ rechnen lassen, wobei im i-ten Schritt die Aktion y_i gewählt wird (Rate-Verifikationsschema).

offline-Randomisierung Wir stellen uns vor, die $T(n)$-zeitbeschränkte PTM M hätte Zugriff auf einen (bezüglich der uniformen Verteilung) zufälligen String $y \in \{0,1\}^{T(n)}$. Auf $\langle y, x\rangle$ rechnet M dann deterministisch, wobei (in Analogie zu NTMs) im i-ten Schritt die Aktion y_i gewählt wird.

Wir wählen im Folgenden das Modell der offline-Randomisierung und setzen zudem oBdA voraus, dass eine $T(n)$-zeitbeschränkte PTM M auf jeder Eingabe $\langle y, x\rangle$ mit $x \in \Sigma^n$ und $y \in \{0,1\}^{T(n)}$ exakt $T(n)$ Schritte rechnet. Da y bezüglich der uniformen Verteilung auf $\{0,1\}^{T(n)}$ ausgewählt wurde, lassen sich die Wahrscheinlichkeiten zum Akzeptieren bzw. Verwerfen von x durch „Zählen“ ermitteln (Anzahl der günstigen Fälle dividiert durch die Anzahl aller Fälle):

$$\begin{aligned}
\Pr_y[M(y,x)=1] &= \frac{|\{y \in \{0,1\}^{T(n)} | M(y,x)=1\}|}{2^{T(n)}} \\
\Pr_y[M(y,x)=0] &= \frac{|\{y \in \{0,1\}^{T(n)} | M(y,x)=0\}|}{2^{T(n)}}
\end{aligned}$$

Wir merken kurz an, dass alle auftretenden Wahrscheinlichkeiten Vielfache von $2^{-T(n)}$ sind. Die Schreibweise „$\Pr_y$“ soll stets implizieren, dass y bezüglich der uniformen Verteilung zufällig aus $\{0,1\}^{T(n)}$ ausgewählt wurde (so dass einer $T(n)$-zeitbeschränkten PTM $T(n)$ unabhängige perfekte Zufallsbits zur Verfügung stehen).

Beim Erkennen einer Sprache L kann M folgende Fehler begehen:

Fehler 1. Art M akzeptiert x, obwohl $x \notin L$.

Fehler 2. Art M verwirft x, obwohl $x \in L$.

Die diversen probabilistischen Komplexitätsklassen unterscheiden sich darin, welche Fehlerwahrscheinlichkeiten toleriert werden. Wir führen nun eine allgemeine generische Notation ein, aus welcher sich die gängigen Klassen leicht ableiten lassen:

Definition 17.1 *Seien $0 \leq \alpha < \beta \leq 1$, $\alpha, \beta \in \mathbb{Q}$ und $L \subseteq \{0,1\}^*$. Wir sagen L gehört zur Klasse $R_{\leq\alpha,\geq\beta}$ gdw eine PTM M mit polynomieller Zeitschranke $T(n)$ existiert, so dass für alle $n \geq 0$ und alle $x \in \{0,1\}^n$ folgende Bedingungen gelten:*

$$\begin{aligned} x \notin L &\implies \Pr_y[M(y,x)=1] \leq \alpha \\ x \in L &\implies \Pr_y[M(y,x)=1] \geq \beta \end{aligned}$$

Die Klassen $R_{<\alpha,\geq\beta}$, $R_{\leq\alpha,>\beta}$ und $R_{<\alpha,>\beta}$ sind analog definiert. Für diese Klassen ist auch der Fall $\alpha = \beta$ (mehr oder weniger) sinnvoll.

Im Folgenden nennen wir eine PTM mit einer polynomiellen Zeitschranke kurz eine PPTM.

Dass sich die uns vertrauten Klassen P und NP als Grenzfälle von probabilistischen Komplexitätsklassen darstellen lassen, lehrt folgendes

Beispiel 17.2 *$P = R_{\leq 0,\geq 1}$ und $NP = R_{\leq 0,>0}$.*

Weiterhin merken wir kurz an:

Lemma 17.3 *Alle von Definition 17.1 induzierten probabilistischen Komplexitätsklassen sind Teilklassen von PSpace.*

Beweis Eine Sprache $L \in R_{\leq\alpha,\geq\beta}$ mit einer PPTM M (polynomielle Zeitschranke T) als Akzeptor kann deterministisch mit polynomiell beschränktem Platzverbrauch erkannt werden wie folgt:

1. Gegeben Eingabe $x \in \{0,1\}^n$, starte M auf $\langle y, x\rangle$ für jedes $y \in \{0,1\}^{T(n)}$, benutze aber jeweils das gleiche Bandsegment.
2. Zähle nebenbei die Anzahl N_+ der akzeptierenden Rechnungen.
3. Akzeptiere schließlich x gdw $N_+ \geq \beta 2^{T(n)}$.

Für die Klassen $R_{<\alpha,\geq\beta}$, $R_{\leq\alpha,>\beta}$ und $R_{<\alpha,>\beta}$ ist die Argumentation völlig analog. **qed.**

In den folgenden Abschnitten diskutieren wir einige Standardbeispiele für probabilistische Komplexitätsklassen:

$$\begin{aligned} PP &= R_{<1/2,>1/2} \\ BPP &= R_{\leq 1/3,\geq 2/3} \\ RP &= R_{\leq 0,\geq 1/2} \\ ZPP &= RP \cap \text{co-}RP \end{aligned}$$

Heutzutage gilt BPP (mehr noch als die Klasse P) als die Klasse der „praktisch lösbaren" Probleme (unter Einsatz von Randomisierung). Algorithmen, die die Mitgliedschaft eines Problems in der Klasse RP (bzw. ZPP) bezeugen, sind auch unter dem Namen „Monte-Carlo Algorithmen" (bzw. „Las-Vegas Algorithmen") bekannt.

17.1 Die Klasse PP

Es ist nicht sinnvoll, für den Fehler 1. und 2. Art jeweils eine Fehlerwahrscheinlichkeit von 1/2 zuzulassen: dann könnten wir nämlich die Frage

$$x \in L \ ?$$

mit dem Wurf einer perfekten Münze entscheiden.[1] Es ist daher eine Art Minimalforderung, dass die Fehlerwahrscheinlichkeit für den Fehler 1. und 2. Art zumindest kleiner als 1/2 sein sollte. Dies führt zur Definition der Klasse PP (Probabilistic Polynomial Time):

$$PP = R_{<1/2,>1/2}$$

Die Minimalforderungen, die wir an die Klasse PP richten, sind allerdings zu schwach. Der Hauptgrund hierfür ist, dass eine Maschine, die einen lediglich „exponentiell kleinen Vorsprung vor zufälligem Raten" besitzt, in ihrem statistischen Ein/Ausgabeverhalten nicht effizient von zufälligem Raten unterschieden werden kann.[2] Ein weiteres Indiz dafür, dass PP keine Klasse mit praktikablen Erkennungsalgorithmen ist, werden wir im Satz 17.5 liefern, welcher besagt, dass NP eine Teilklasse von PP ist.

Wir beweisen als kleine Aufwärmübung zunächst das folgende

Lemma 17.4 $PP := R_{<1/2,>1/2} = R_{\leq 1/2,>1/2}$.

[1] Deshalb hatten wir die Klasse $R_{\leq\alpha,\geq\beta}$ nur im Falle $\alpha < \beta$ zugelassen. $R_{\leq 1/2,\geq 1/2}$ wäre die Klasse aller Sprachen!

[2] Wir verzichten an dieser Stelle auf eine Konkretisierung und einen mathematischen Beweis dieser Aussage. Im Prinzip läuft die Argmentation darauf hinaus, dass man viel zu viele Experimente machen müsste, um eine infinitesimal unfaire Münze zuverlässig von einer fairen Münze zu unterscheiden.

Beweis Die Inklusion $R_{<1/2,>1/2} \subseteq R_{\leq 1/2,>1/2}$ ist trivial. Wir zeigen, dass auch die Inklusion

$$R_{\leq 1/2,>1/2} \subseteq R_{<1/2,>1/2}$$

gültig ist. Es sei $L \in R_{\leq 1/2,>1/2}$ und M eine entsprechende PPTM mit Zeitschranke $T(n)$, wobei T ein Polynom mit Koeffizienten aus $\mathbb{N}$ bezeichnet. Da die Wahrscheinlichkeit für das Ereignis „M akzeptiert Eingabe $x \in \{0,1\}^n$“ ein Vielfaches von $2^{-T(n)}$ ist, folgern wir

$$\Pr_y[M(y,x)=1] > \frac{1}{2} \Leftrightarrow \Pr_y[M(y,x)=1] \geq \frac{1}{2} + 2^{-T(n)} \ . \tag{1}$$

Wir erweitern M zu einer PPTM M', die zusätzliche Zufallsbits $b_0, b_1, \ldots, b_{T(n)}$ verwendet und arbeitet wie folgt:

1. Falls $b_0 = b_1 = \cdots = b_{T(n)} = 0$ (ein Ereignis der Wahrscheinlichkeit $2^{-(1+T(n))}$)) dann verwerfe x und stoppe. Andernfalls mache weiter.
2. Starte M auf $\langle y, x\rangle$ und entscheide wie M.

Wenn M die Eingabe x verwirft, so auch M'. Es gibt aber auch eine positive Wahrscheinlichkeit dafür, dass M Eingabe x akzeptiert, aber M' sie verwirft, weil alle b-Bits Nullen sind. Folglich gilt für alle $x \notin L$:

$$\Pr_{yb}[M'(yb,x)=1] < \Pr_y[M(y,x)=1] \leq \frac{1}{2} \ .$$

Weiterhin gilt (wegen der Subadditivität von Wahrscheinlichkeitsmaßen) für alle $x \in L$:

$$\begin{aligned} \Pr_{yb}[M(yb,x)=0] &\leq \Pr_y[M(y,x)=0] + 2^{-(1+T(n))} \\ &\overset{(1)}{\leq} \frac{1}{2} - 2^{-T(n)} + 2^{-(1+T(n))} \\ &= \frac{1}{2} - 2^{-(1+T(n))} < \frac{1}{2} \end{aligned}$$

und somit

$$\Pr_{yb}[M(yb,x)=1] > \frac{1}{2} \ .$$

PPTM M' bezeugt, dass $L \in R_{<1/2,>1/2}$. **qed.**

Wir beschließen diesen Abschnitt mit dem folgenden

Satz 17.5 *$NP \subseteq PP$.*

Beweis Wegen Lemma 17.4 genügt es

$$NP \subseteq R_{\leq 1/2,>1/2}$$

nachzuweisen. Es sei L eine beliebige aber fest ausgewählte Sprache aus $NP = R_{\leq 0,>0}$ und M eine entsprechende PPTM. Wir erweitern M zu einer PPTM M', die ein zusätzliches Zufallsbit b verwendet und arbeitet wie folgt:

1. Falls $b = 1$ (ein Ereignis der Wahrscheinlichkeit 1/2), dann akzeptiere und stoppe. Andernfalls mache weiter.

2. Starte M auf $\langle y, x\rangle$ und entscheide wie M.

Offensichtlich gilt für alle $x \notin L$ (welche von M niemals akzeptiert werden):

$$\Pr_{yb}[M(yb, x) = 1] = \frac{1}{2}$$

Unter Verwendung des „Satzes der bedingten Wahrscheinlichkeiten“ erhalten wir für alle $x \in L$ (die von M mit einer positiven, obschon evtl. sehr kleinen, Wahrscheinlichkeit akzeptiert werden):

$$\begin{aligned}\Pr_{yb}[M(yb, x) = 1] &= \frac{1}{2}\Pr_{yb}[M(yb, x) = 1|b = 1] + \frac{1}{2}\Pr_{yb}[M(yb, x) = 1|b = 0] \\ &= \frac{1}{2} \cdot 1 + \frac{1}{2}\underbrace{\Pr_{y}[M(y, x) = 1]}_{>0} > \frac{1}{2}\end{aligned}$$

PPTM M' bezeugt, dass $L \in R_{\leq 1/2, >1/2}$. **qed.**

17.2 Die Klasse BPP

Die Lehre aus dem Abschnitt 17.1 ist, dass die Wahrscheinlichkeiten für den Fehler 1. und 2. Art „deutlich“ kleiner als 1/2 sein sollten, damit die randomisierten Entscheidungen (Akzeptieren versus Verwerfen) sich signifikant von zufälligem Raten unterscheiden. Dies ist bei der Definition der Klasse BPP (Bounded-away from 1/2 Probabilistic Polynomial Time) berücksichtigt:

$$BPP = R_{\leq 1/3, \geq 2/3}$$

Um mit dieser technischen Definition vertraut zu werden, zeigen wir zunächst, dass die Auswahl der Konstanten $1/3, 2/3$ willkürlich ist und es nur darauf ankommt die Fehlerwahrscheinlichkeit „deutlich“ von 1/2 abzugrenzen. Anschließend zeigen wir, dass BPP auf dem 2. Level der polynomiellen Hierarchie (oder noch tiefer)[3] angesiedelt ist.

Um die Grenzen der Fehlerwahrscheinlichkeiten auszuloten, welche wir an die Stelle der Konstanten $1/3, 2/3$ setzen können, benötigen wir folgende natürliche Verallgemeinerung von Definition 17.1:

Definition 17.6 *Es seien $\alpha = (\alpha_n)_{n\geq 0}$ und $\beta = (\beta_n)_{n\geq 0}$ zwei Folgen mit $0 \leq \alpha_n < \beta_n \leq 1$ und $\alpha_n, \beta_n \in \mathbb{Q}$ für alle $n \geq 0$. Wir sagen L gehört zur Klasse $R_{\leq\alpha,\geq\beta}$ gdw eine PTM M*

[3]Sogar $BPP = P$ lässt sich beim derzeitigen Stand des Wissens nicht ausschließen.

mit polynomieller Zeitschranke $T(n)$ (also eine PPTM) existiert, so dass für alle $n \geq 0$ und alle $x \in \{0,1\}^n$ folgende Bedingungen gelten:

$$\begin{aligned} x \notin L &\implies \Pr_y[M(y,x)=1] \leq \alpha_n \\ x \in L &\implies \Pr_y[M(y,x)=1] \geq \beta_n \end{aligned}$$

Die Klassen $R_{<\alpha,\geq\beta}$, $R_{\leq\alpha,>\beta}$ und $R_{<\alpha,>\beta}$ sind analog definiert.

Wir zeichnen zwei Nullfolgen $\epsilon = (\epsilon_n)$ und $\delta = (\delta_n)$ aus:

$$\epsilon_n = 2^{-n^l} \text{ und } \delta_n = n^{-k} \tag{2}$$

Hierbei seien $k, l \in \mathbb{N}$ natürlich-zahlige Konstanten. Offensichtlich gilt

$$R_{\leq\epsilon,\geq 1-\epsilon} \subseteq \overbrace{R_{\leq 1/3,\geq 2/3}}^{=BPP} \subseteq R_{\leq 1/2-\delta,\geq 1/2+\delta} \ . \tag{3}$$

ϵ_n ist eine phantastisch kleine Fehlerrate, die mit exponentieller Geschwindigkeit gegen Null konvergiert. $1/2 - \delta_n$ hingegen kommt verdächtig nahe an die Fehlerrate $1/2$ von zufälligem Raten heran.[4] Obwohl bei oberflächlicher Betrachtung eine „Galaxie“ zwischen den Fehlerraten ϵ_n und $1/2 - \delta_n$ zu liegen scheint, werden wir jetzt nachweisen, dass alle in (3) aufgeführten Klassen zu *BPP* identisch sind. Hierzu genügt es freilich, die Inklusion

$$R_{\leq 1/2-\delta,\geq 1/2+\delta} \subseteq R_{\leq\epsilon,\geq 1-\epsilon}$$

nachzuweisen. In Worten: Eine PPTM M, die lediglich einen polynomiellen Vorteil über zufälliges Raten erzielt, lässt sich in eine PPTM M' mit einer exponentiell kleinen Fehlerrate transformieren. Wir beweisen zu diesem Zweck die folgende (etwas allgemeinere) Aussage:

Satz 17.7 *Es sei $\tau = (\tau_n)_{n\geq 0}$ eine in poly(n) Schritten berechenbare Folge rationaler Zahlen im Intervall von 0 bis 1.*[5] *Weiter sei M eine PPTM, die für alle $n \geq 0$ und alle $x \in \{0,1\}^n$ die folgenden Bedingungen erfüllt:*

$$\begin{aligned} x \notin L &\implies \Pr_y[M(y,x)=1] \leq \tau_n - \delta_n \\ x \in L &\implies \Pr_y[M(y,x)=1] \geq \tau_n + \delta_n \end{aligned}$$

Dann existiert eine PPTM M' für L mit Fehlerrate ϵ (was $L \in R_{\leq\epsilon,\geq 1-\epsilon}$ bezeugt).

Beweis PPTM M' gehe auf Eingabe $x \in \{0,1\}^n$ vor wie folgt:

1. Berechne τ_n aus x. (Hierfür spielt nur die Länge n von x eine Rolle.)

[4] Da dieser Term jedoch noch um den Kehrwert eines Polynoms von $1/2$ weg-separiert ist, spricht man von „polynomiellem Vorteil über zufälliges Raten“).

[5] D.h., Aus 0^n (der unären Kodierung von n) lässt sich τ_n in poly(n) Schritten berechnen.

2. Für eine hinreichend große (Präzisierung erfolgt aus didaktischen Gründen später) natürliche Zahl R lasse M R-mal auf Eingabe x laufen (mit jeweils neuen, unabhängigen Zufallsbits) und bestimme dabei die absolute Häufigkeit R_+, mit welcher x von M akzeptiert wird.

3. Akzeptiere x gdw $R_+ \geq \tau_n R$.

Wir machen folgende Beobachtungen:

- R_+ ist eine binomial verteilte Zufallsvariable (Anzahl der Erfolge bei R unabhängig durchgeführten Bernoulli-Experimenten).
- Falls $x \in L$ (und M daher mit einer Wahrscheinlichkeit von mindestens $\tau_n + \delta_n$ akzeptiert), dann gilt
$$\mathbb{E}[R_+] \geq (\tau_n + \delta_n)R \ .$$
- Falls $x \notin L$ (und M daher mit einer Wahrscheinlichkeit von höchstens $\tau_n - \delta_n$ akzeptiert), dann gilt
$$\mathbb{E}[R_+] \leq (\tau_n - \delta_n)R \ .$$
- M' begeht einen Fehler höchstens dann, wenn die Zufallsvariable R_+ von ihrem Erwartungswert um mindestens $\delta_n R$ abweicht.

Mit Hilfe der (aus der Statistik bekannten)[6] Chernov-Schranken folgt, dass die Abweichung einer binomial-verteilten Zufallsvariable von ihrem Erwartungswert (bei hinreichend großer Anzahl von Versuchen) ein eher unwahrscheinliches Ereignis ist. Genauer:

$$\Pr[|R_+ - \mathbb{E}[R_+]| \geq \delta_n R] \leq 2e^{-2\delta_n^2 R}$$

Eine kleine Rechnung ergibt

$$2e^{-2\delta_n^2 R} \Leftrightarrow R \geq \frac{\ln(2/\epsilon_n)}{2\delta_n^2} \ .$$

Wegen $\delta_n = n^{-k}$ und $\epsilon_n = 2^{-n^l}$ ergibt sich nun, dass

$$R := n^{2k+l+1}$$

hinreichend groß ist, um für M' die Fehlerschranke ϵ_n zu garantieren. **qed.**

Der Spezialfall $\tau_n = 1/2$ liefert nun die

Folgerung 17.8 *Die Klassen $R_{\leq\epsilon,\geq 1-\epsilon}$ und $R_{\leq 1/2-\delta,\geq 1/2+\delta}$ stimmen beide mit der Klasse $BPP = R_{\leq 1/3,\geq 2/3}$ überein.*

Die Technik, eine hohe Fehlerrate durch wiederholte Anwendung eines Zufallsexperimentes signifikant zu verkleinern, wird „Boosting“ genannt. Bei PPTMs mit beidseitigem Fehler haben wir dabei eine Art „Majoritätsvotum“ in Verbindung mit den „Chernov-Schranken“ eingesetzt. Bei PPTMs mit einseitigem Fehler wird sich später ein einfacheres Boosting-Argument ergeben.

[6] s. auch Begleitmaterial zur Vorlesung

Wir versuchen im Folgenden, *BPP* in die polynomielle Hierarchie einzuordnen. Zu diesem Zweck definieren wir eine neue Komplexitätsklasse:

Definition 17.9 *Wir sagen eine Sprache $L \subseteq \Sigma^*$ gehört zur Komplexitätsklasse Φ_2 gdw eine Sprache $L_0 \in P$ existiert, so dass für alle $n \geq 0$ die folgenden Bedingungen gelten:*

$$x \in L \quad \Leftrightarrow \quad (\exists y)_{pol}(\forall z)_{pol} : \langle x, y, z\rangle \in L_0 \tag{4}$$

$$x \notin L \quad \Leftrightarrow \quad (\exists z)_{pol}(\forall y)_{pol} : \langle x, y, z\rangle \notin L_0 \tag{5}$$

Die Klasse Φ_2 wird von Δ_2 und $\Sigma_2 \cap \Pi_2$ „gesandwiched“:

Lemma 17.10 $\Delta_2 \subseteq \Phi_2 \subseteq \Sigma_2 \cap \Pi_2$.

Beweis Wir beweisen zunächst $\Phi_2 \subseteq \Sigma_2 \cap \Pi_2$. Wähle eine beliebige Sprache L aus Φ_2 fest aus. Aus Bedingung (4) lässt sich unmittelbar $L \in \Sigma_2$ ablesen. Aus Bedingung (5) ergibt sich nun $L \in \Pi_2$ wie folgt:

$$L = \{x|\, \neg(x \notin L)\} = \{x|\, \neg((\exists z)_{pol}(\forall y)_{pol} : \langle x, y, z\rangle \notin L_0)\} = \{x|\, (\forall z)_{pol}(\exists y)_{pol} : \langle x, y, z\rangle \in L_0\}$$

Somit hat sich insgesamt $L \in \Sigma_2 \cap \Pi_2$ und daher $\Phi_2 \subseteq \Sigma_2 \cap \Pi_2$ ergeben.
Der Rest des Beweises ist der Inklusion $\Delta_2 \subseteq \Phi_2$ gewidmet. Wir wählen eine beliebige Sprache $L \in \Delta_2 = P[NP]$ fest aus. Dann gibt es eine Sprache $L' \in NP$ und eine polynomiell zeitbeschränkte DOTM $M[L']$, welche Eingaben aus L erkennen kann. Wir wählen eine beliebige Eingabe $x \in \Sigma^n$ fest aus. Es seien

$$(g_1, b_1), \ldots, (g_s, b_s)$$

mit

$$b_i = \begin{cases} 0 & \text{falls } g_i \notin L' \\ 1 & \text{falls } g_i \in L' \end{cases}$$

die Fragen und Antworten in der Kommunikation zwischen $M[L']$ und ihrem L'-Orakel. Für alle i mit $g_i \in L'$ bezeichne h_i ein (in Polynomialzeit prüfbares) Zertifikat, das die Mitgliedschaft von g_i in L' bezeugt. Um $L \in \Phi_2$ nachzuweisen, suchen wir nach einer Darstellung von L, die den Bedingungen (4) und (5) genügt. Zu diesem Zweck definieren wir ein *Kommunikationscodewort* W gemäß

$$W := \langle w_1, \ldots, w_s\rangle \ ,$$

wobei

$$w_i := \begin{cases} \langle g_i, 0\rangle & \text{falls } g_i \notin L' \\ \langle g_i, 1, h_i\rangle & \text{falls } g_i \in L' \end{cases} \ .$$

W kodiert die (ggf. um Zertifikate erweiterte) Kommunikation zwischen $M[L']$ und ihrem Orakel. Wir nutzen im Folgenden aus, dass W eine effiziente Simulation von $M[L']$ auf Eingabe x erlaubt. Genauer: wir definieren eine polynomiell zeitbeschränkte DTM M_0 (die im Wesentlichen $M[L']$ simuliert) und zeigen, dass die von ihr induzierte Sprache L_0 den an eine Sprache $L \in \Phi_2$ gerichteten Bedingungen (4) und (5) genügt:

- Eine Eingabe von M_0 heiße *zulässig*, falls sie die Form $\langle x, y, z\rangle$ hat, wobei $y = W$ oder $z = W$ gelten soll.
- M_0 soll eine zulässige Eingabe $\langle x, y, z\rangle$ mit $y = W$ oder $z = W$ akzeptieren gdw $x \in L$.[7]

Man überlegt sich leicht, dass „syntaktisch inkorrekte“ Eingaben, die nicht von der Form $\langle x, y, z\rangle$ sind (wobei mindestens einer der Strings y, z von der Syntax her ein „potenzielles“ Kommunikationscodewort sein muss), in Polynomialzeit entlarvt werden. Wir können daher annehmen, dass eine syntaktisch korrekte Eingabe der Form $\langle x, y, z\rangle$ (mit zumindest einem potenziellen Kommunikationscodewort) vorliegt. Ein offensichtliches Dilemma für M_0 besteht darin, dass sie nicht weiß, ob y oder z (oder evtl. keiner von beiden) das korrekte Kommunikationscodewort ist. Vergleichsweise harmlos ist dabei der Fall, dass einer der Strings y, z (oder gar beide) eines der folgenden (in Polynomialzeit erkennbaren) „Fouls“ begeht:

Foul 1 Er weicht bereits syntaktisch von einem Kommunikationscodewort ab.

Foul 2 Er hat die syntaktische Form eines Kommunikationscodewortes, enthält aber nicht exakt die Fragen, die $M[L']$ an ihr Orakel richtet.

Falls weder $y = W$ noch $z = W$ (insbesondere also, falls beide Strings ein Foul begehen), dann darf M_0 sowieso machen, was sie will. Nehmen wir also an, dass eine zulässige Eingabe mit $y = W$ oder $z = W$ vorliegt. Falls einer der Strings ein Foul begeht, dann ist (bei einer zulässigen Eingabe) der andere String das Kommunikationscodewort und M kann die Simulation von $M[L']$ problemlos durchführen.

Böse, böse Was aber, wenn weder y noch z ein Foul begeht?

Man überlegt sich leicht, dass M_0 auch unter diesen (maximal widrigen) Umständen korrekt arbeiten kann, indem sie ein paar Regeln beherzigt:

- Falls y, z zur Anfrage g_i das gleiche Antwortbit enthalten, dann setze die Simulation mit diesem Antwortbit fort.
- Im Konfliktfall benutze das Zertifikat h_i, um

  $$g_i \in L' \ ?$$

  zu testen. Falls die Verfikation gelingt, dann setze die Simulation mit Antwortbit 1 fort. Falls nicht, dann mit Antwortbit 0.

Es hat sich also gezeigt, dass eine DTM M_0 mit den gewünschten Eigenschaften existiert. Es bezeichne L_0 die zugehörige Sprache. Wir wollen nun argumentieren, dass L, L_0 in der Beziehung (4) zueinander stehen:

- Die Richtung „$\Rightarrow$“ ergibt sich mit $y = W$.

[7] Bei unzulässigen Eingaben machen wir M_0 keine Vorschriften.

- Die umgekehrte Richtung kann indirekt bewiesen werden. Für $x \notin L$ kann kein y existieren, so dass für alle z die Bedingung $\langle x, y, z\rangle \in L_0$ erfüllt ist. Zumindest für $z = W$ würde nämlich M_0 die Eingabe $\langle x, y, z\rangle$ verwerfen.

Schließlich argumentieren wir, dass L, L_0 auch in der Beziehung (5) zueinander stehen:

- Die Richtung „$\Rightarrow$“ ergibt sich mit $z = W$.
- Die umgekehrte Richtung kann indirekt bewiesen werden. Für $x \in L$ kann kein z existieren, so dass für alle y die Bedingung $\langle x, y, z\rangle \notin L_0$ erfüllt ist. Zumindest für $y = W$ würde nämlich M_0 die Eingabe $\langle x, y, z\rangle$ akzeptieren.

Somit hat sich $L \in \Phi_2$ und daher auch $\Delta_2 \subseteq \Phi_2$ ergeben. **qed.**

Zum Beweis des Hauptresultates benötigen wir noch das folgende

Lemma 17.11 *Es sei $A \subseteq \{0,1\}^N$ mit $|A| \geq \frac{2}{3}2^N$ und $k = 18N$. Dann existieren $y_1, \ldots, y_k \in \{0,1\}^N$, so dass für alle $z \in \{0,1\}^N$ die Bedingung*

$$|\{i \in \{1, \ldots, k\} : \ y_i \oplus z \in A\}| > \frac{k}{2} \tag{6}$$

erfüllt ist (wobei „$\oplus$“ die komponentenweise Addition modulo 2 bezeichnet).

Beweis Wir verwenden die probabilistische Methode, d.h., wir zeigen, dass es eine echt positive Wahrscheinlichkeit gibt, eine „passende“ Sequenz $\bar{y} := (y_1, \ldots, y_k) \in \{0,1\}^{kN}$ zufällig zu generieren. Wir nennen die Sequenz $\bar{y}$ „gut“ für $z \in \{0,1\}^N$, falls $\bar{y}$ und z die Bedingung (6) erfüllen. Ansonsten heiße $\bar{y}$ „schlecht“ für $z \in \{0,1\}^N$. In dieser Sprechweise ist zu zeigen, dass eine Sequenz $\bar{y} \in \{0,1\}^{kN}$ existiert, die simultan für alle $z \in \{0,1\}^N$ gut ist. Betrachte eine Sequenz $\bar{y}$ die (bezüglich der uniformen Verteilung) zufällig aus $\{0,1\}^{kN}$ gewählt ist. Für ein beliebig aber fest ausgewähltes $z \in \{0,1\}^N$ sei Z_i die Bernoulli-Variable

$$Z_i := \begin{cases} 1 & \text{falls } y_i \oplus z \in A \\ 0 & \text{falls } y_i \oplus z \notin A \end{cases} .$$

Dann sind $Z_1, \ldots, Z_k$ identisch verteilte unabhängige Bernoulli-Variable mit Erfolgswahrscheinlichkeit $p \geq 2/3$. Damit $\bar{y}$ schlecht für z ist, müsste $Z_1 + \cdots + Z_k$ um mindestens $k/6$ vom Erwartungswert $pk \geq 2k/3$ abweichen. Mit den (uns inzwischen bekannten) Chernov-Schranken (und mit $k = 18N$) ergibt sich, dass die Wahrscheinlichkeit hierfür durch $e^{-2(1/6)^2 k} = e^{-N}$ nach oben beschränkt ist. Mit der Subadditivität von Wahrscheinlichkeitsmaßen folgern wir weiter: die Wahrscheinlichkeit, eine zufällige Sequenz $\bar{y}$ zu generieren, so dass ein $z \in \{0,1\}^N$ existiert, für welche $\bar{y}$ schlecht ist, ist nach oben durch $2^N e^{-N} < 1$ beschränkt. Somit gibt es eine positive Wahrscheinlichkeit, eine Sequenz $\bar{y}$ zu realisieren, die für alle $z \in \{0,1\}^N$ gut ist. **qed.**

Nun zum Finale des laufenden Kapitels:

Satz 17.12 $BPP \subseteq \Phi_2$.

Beweis Wähle eine beliebige Sprache $L \in BPP$ fest aus. Es sei M eine PPTM für L mit einer polynomiellen Zeitschranke $T(n)$ und einer durch 1/3 beschränkten Fehlerrate (für die Fehler jeweils beider Arten).

Ziel Darstellung von L gemäß der Definition von Φ_2.

Wähle ein $n \geq 0$ und eine Eingabe $x \in \Sigma^n$ beliebig aber fest aus. Setze $N := T(n)$ und $k := 18N$. Es bezechne χ_L die charakteristische Funktion für die Sprache L. Die Menge

$$A := \{y \in \{0,1\}^N |\ M(y,x) = \chi_L(x)\}$$

hat eine Mächtigkeit $|A| \geq \frac{2}{3}2^N$, da $|A|$ die Anzahl der korrekten Rechnungen der PPTM M (mit einer durch 1/3 beschränkten Fehlerwahrscheinlichkeit) auf Eingabe x ist. Es bezeichne L_0 die Sprache bestehend aus allen $\langle x, y, z\rangle$ mit folgenden Eigenschaften:

1. y hat die Form $\langle y_1, \ldots, y_k\rangle$ mit $y_i \in \{0,1\}^N$ für $i = 1, \ldots, k$.
2. z hat die Form $\langle z_1, \ldots, z_k\rangle$ mit $z_i \in \{0,1\}^N$ für $i = 1, \ldots, k$.
3. Eine Mehrheit der k^2 Rechnungen $M(y_i \oplus z_j, x)$ ist akzeptierend.

Da A die Bedingungen von Lemma 17.11 erfüllt, folgt leicht, dass L und L_0 in den Beziehungen (4) und (5) zueinander stehen. Somit gilt $L \in \Phi_2$ und daher auch $BPP \subseteq \Phi_2$.

qed.

17.3 Die Klasse RP

Die Komplexitätsklassen PP und BPP lassen prinzipiell einen beidseitigen Fehler zu. In diesem Abschnitt lernen wir eine Klasse kennen, bei welcher lediglich ein einseitiger Fehler zugelassen ist. Die Klasse RP (Random Polynomial Time) ist gegeben durch

$$RP = R_{\leq 0, \geq 1/2}\ .$$

Eingaben $x \notin L$ werden also stets verworfen und Eingaben $x \in L$ werden mit einer Wahrscheinlichkeit von mindestens 1/2 akzeptiert. Eine PPTM, die diesem Kriterium genügt, ist also auf Eingaben $x \notin L$ fehlerfrei. Wir merken kurz an, dass RP wegen

$$RP = R_{\leq 0, \geq 1/2} \subseteq R_{\leq 0, >0} = NP$$

in NP enthalten ist.

Ähnlich wie im Falle BPP werden wir zeigen, dass die Wahl der Konstante 1/2 willkürlich ist:

Lemma 17.13 *Für die in (2) definierten Nullfolgen $\epsilon = (\epsilon_n)_{n\geq 0}$ und $\delta = (\delta_n)_{n\geq 0}$ gilt:*

$$R_{\leq 0,\geq 1-\epsilon} = \overbrace{R_{\leq 0,\geq 1/2}}^{=:RP} = R_{\leq 0,\geq \delta} \ .$$

Beweis Wegen der offensichtlichen Inklusion

$$R_{\leq 0,\geq 1-\epsilon} \subseteq R_{\leq 0,\geq 1/2} \subseteq R_{\leq 0,\geq \delta}$$

genügt es,

$$R_{\leq 0,\geq \delta} \subseteq R_{\leq 0,\geq 1-\epsilon}$$

zu beweisen. In Worten: jede PPTM mit einseitigem Fehler, aber einer Fehlerrate von $1-\delta$ auf Eingaben aus L, kann in eine PPTM mit einseitigem Fehler transformiert werden, die auch auf Eingaben in L eine exponentiell kleine Fehlerrate aufweist. Sei nun eine Sprache $L \in R_{\leq 0,\geq \delta}$ mit einer hierzu passenden PPTM M vorgegeben, so dass für alle $n \geq 0$ und alle $x \in \Sigma^n$ gilt:

$$\begin{aligned} x \notin L &\Rightarrow \Pr_y[M(y,x) = 1] = 0 \\ x \in L &\Rightarrow \Pr_y[M(y,x)] = 1] \geq \delta_n \end{aligned}$$

Dann sei M' die PPTM, die auf Eingabe $x \in \Sigma^n$ arbeitet wie folgt:

1. Wende M R-mal (jeweils mit neuen unabhängigen Zufallsbits) auf Eingabe x an (wobei wir R aus didaktischen Gründen erst später festlegen).
2. Falls x in mindestens einer der R Rechnungen akzeptiert wurde, so stoppe akzeptierend. Andernfalls stoppe verwerfend.

Falls $x \notin L$, dann wird x von M R-mal verworfen. Somit stoppt auch M' schließlich verwerfend. Falls $x \in L$, dann macht M' einen Fehler (verwerfendes Stoppen) gdw alle R Rechnungen von M auf Eingabe x verwerfend sind. Da eine einzelne Rechnung von M auf x mit einer Wahrscheinlichkeit von höchstens $1-\delta_n$ verwerfend ist, ergibt sich für die Wahrscheinlichkeit, dass alle R (unabhängig voneinander ausgeführten) Rechnungen von M auf Eingabe x verwerfend sind (gemäß der Produktformel für die Konjunktion unabhängiger Ereignisse) die obere Schranke

$$(1-\delta_n)^R < e^{-\delta_n R} \ .$$

(Hierbei wurde die Formel $1+a \leq e^a$ mit Gleichheit nur für $a = 0$ benutzt, die für alle $a \in \mathbb{R}$ gültig ist.) Eine elementare Rechnung ergibt

$$e^{-\delta_n R} \leq \epsilon_n \Leftrightarrow R \geq \frac{\ln(1/\epsilon_n)}{\delta_n} \ .$$

Wegen $\epsilon_n = 2^{-n^l}$ und $\delta_n = n^{-k}$ ist $R := n^{k+l}$ eine hinreichend große Wahl von Parameter R. Unsere Diskussion hat ergeben, dass M' eine PPTM ist, welche $L \in R_{\leq 0,\geq 1-\epsilon}$ bezeugt (was zu beweisen war). **qed.**

Folgerung 17.14 *Für jede Konstante $0 < c < 1$ gilt: $RP = R_{\leq 0,\geq c}$.*

17.4 Die Klasse ZPP

Die bisher betrachteten PPTMs lassen für den menschlichen Benutzer einen Rest Unsicherheit übrig. Selbst bei PPTMs mit einseitigem Fehler können wir nicht sicher sein (obschon die Fehlerwahrscheinlichkeit via „Boosting“ vernachlässigbar klein gemacht werden kann), dass eine verworfene Eingabe nicht vielleicht doch zur Sprache gehört und hätte akzeptiert werden sollen. Eine Fehlerwahrscheinlichkeit von, sagen wir, 2^{-1000} ist sicher nicht größer als die Wahrscheinlichekeit, dass alle Kernkraftwerke dieser Erde gleichzeitig ihren Supergau erleben. Allein: eine auch noch so kleine Fehlerwahrscheinlichkeit bleibt ein „Stachel im Fleisch“ des wahren Perfektionisten! Voilá, hier ist nun die Definition der PPTM, die Balsam auf die Wunden aller Puristen und Perfektionisten träufelt:

Definition 17.15 *Eine* fehlerfreie *PPTM M, ist eine PPTM, deren Ausgaben* 0 *(für verwerfende Rechnungen) und* 1 *(für akzeptierende Rechnungen) stets absolut verlässlich sind. Sie verfügt über eine dritte Ausgabe „?“ (für „weiß nicht“), die aber auf jeder Eingabe mit einer Wahrscheinlichkeit von höchstens* $1/2$ *produziert wird.*

Definition 17.16 *Die Klasse ZPP (Zerro Error Probabilistic Polynomial Time) besteht aus allen Sprachen, die eine fehlerfreie PPTM zum Akzeptor haben.*

Wie im Falle von RP ist die Wahl der Konstanten $1/2$ in der Definition von fehlerfreien PPTMs willkürlich. Sie kann durch jede Konstante $0 < c < 1$ (oder auch durch $1 - \epsilon_n$ bzw. δ_n) ersetzt werden, ohne dass die davon induzierte Klasse ZPP sich ändert.

Wenn wir bei einer fehlerfreien PPTM für die Sprache L Ausgabe 0 und Ausgabe 1 vertauschen, erhalten wir eine fehlerfreie PPTM für die Komplementärsprache $\bar{L}$. Somit gilt das

Lemma 17.17 *ZPP = co-ZPP.*

Folgendes Resultat klärt das Verhältnis von RP und ZPP:

Satz 17.18 *$ZPP = RP \cap$ co-RP.*

Beweis Wir weisen zunächst

$$ZPP \subseteq RP$$

nach. Sei $L \in ZPP$ und M eine dazu passende fehlerfreie PPTM. Die PPTM M' arbeite wie M, außer dass statt „?“ stets „0“ ausgegeben werde. Es folgt:

$$\begin{aligned} x \notin L &\Rightarrow \Pr_y[M'(y,x)=1] = \Pr_y[M(y,x)=1] = 0 \\ x \in L &\Rightarrow \Pr_y[M'(y,x)=1] = \Pr_y[M(y,x)=1] \geq \frac{1}{2} \end{aligned}$$

M' bezeugt, dass $L \in RP$. Somit gilt $ZPP \subseteq RP$.
Durch Dualisierung erhalten wir

$$ZPP = \text{co-}ZPP \subseteq \text{co-}RP,$$

womit dann auch

$$ZPP \subseteq RP \cap \text{co-}RP$$

nachgewiesen wäre.
Bleibt also zu zeigen, dass

$$RP \cap \text{co-}RP \subseteq ZPP \ .$$

Sei $L \in RP \cap \text{co-}RP$. Es sei weiter M die PPTM, welche $L \in RP$ bezeugt und entsprechend $\bar{M}$ die PPTM, welche $L \in \text{co-}RP$ und somit $\bar{L} \in RP$ bezeugt. Wir betrachten die PPTM M', die arbeitet wie folgt:

1. Wende M und $\bar{M}$ auf Eingabe x an.
2. Falls M akzeptiert, gib 1 aus, falls $\bar{M}$ akzeptiert gib 0 aus, und falls weder M noch $\bar{M}$ akzeptiert, gib „?“ aus.

Wegen

$$\begin{aligned} x \notin L \quad &\Longrightarrow \quad \left(\Pr_y[M(y,x)=1] = 0 \text{ und } \Pr_y[\bar{M}(y,x)=1] \geq \frac{1}{2} \right) \\ x \in L \quad &\Longrightarrow \quad \left(\Pr_y[\bar{M}(y,x)=1] = 0 \text{ und } \Pr_y[M(y,x)=1] \geq \frac{1}{2} \right) \end{aligned}$$

entscheidet sich M' immer korrekt und gibt mit einer Wahrscheinlichkeit von maximal $1/2$ ein Fragezeichen aus. M' bezeugt, dass $L \in ZPP$, womit auch $RP \cap \text{co-}RP \subseteq ZPP$ bewiesen wäre. **qed.**

17.5 Abschluss unter Komplement

Wir hatten bereits angemerkt, dass $ZPP = \text{co-}ZPP$. In diesem Abschnitt gehen wir der Frage des Abschlusses unter Komplement etwas systematischer nach.

Es sei M eine PPTM, die $L \in R_{\leq\alpha,\geq\beta}$ bezeugt und $\bar{M}$ die PPTM, die aus M durch Vertauschen der Ausgaben 0 und 1 hervorgeht. Hieraus ergibt sich

$$x \in \bar{L} \Rightarrow \Pr_y[M(y,x)=1] \leq \alpha \Rightarrow \Pr_y[M(y,x)=0] \geq 1-\alpha \Rightarrow \Pr_y[\bar{M}(y,x)=1] \geq 1-\alpha$$

und

$$x \notin \bar{L} \Rightarrow x \in L \Rightarrow \Pr_y[M(y,x)=1] \geq \beta \Rightarrow \Pr_y[M(y,x)=0] \leq 1-\beta \Rightarrow \Pr_y[\bar{M}(y,x)=1] \leq 1-\beta \ .$$

Offensichtlich bezeugt $\bar{M}$, dass $L \in R_{\leq 1-\beta,\geq 1-\alpha}$. Aus diesen Überlegungen lassen sich folgende Schlüsse ziehen:

Folgerung 17.19 *Es gilt:*

$$\begin{aligned} L \in R_{\leq\alpha,\geq\beta} &\Rightarrow \bar{L} \in R_{\leq 1-\beta,\geq 1-\alpha} \\ L \in R_{\leq\alpha,\geq 1-\alpha} &\Rightarrow \bar{L} \in R_{\leq\alpha,\geq 1-\alpha} \end{aligned}$$

Analoge Aussagen gelten für die Klassen $R_{<\alpha,\geq\beta}$, $R_{\leq\alpha,>\beta}$ und $R_{<\alpha,>\beta}$. Insbesondere gilt

$$PP = \textit{co-PP} \text{ und } BPP = \textit{co-BPP} \ .$$

Unter den von uns näher diskutierten Klassen ist RP die einzige „asymmetrisch definierte“ Klasse, die vermutlich nicht unter Komplement abgeschlossen ist.

17.6 Die Landschaft der Komplexitätsklassen

Wir wollen ein Bild entwerfen, das die probabilistischen und deterministischen Komplexitätsklassen (sagen wir zwischen $\mathcal{L}$ und *PSpace*) und ihre Querbeziehungen darstellt. Aus der trivialen (und auch schon mehrfach ausgenutzten) Beziehung

$$0 \leq \alpha \leq \alpha' < \beta' \leq \beta \leq 1 \Rightarrow R_{\leq\alpha,\geq\beta} \subseteq R_{\leq\alpha',\geq\beta'}$$

(und den analogen Beziehungen für die Klassen $R_{<\alpha,\geq\beta}$, $R_{\leq\alpha,>\beta}$, $R_{<\alpha,>\beta}$) und aus

$$P = R_{\leq 0,\geq 1},\ RP = R_{\leq 0,\geq 1/2},\ BPP = R_{\leq 1/3,\geq 2/3},\ PP = R_{<1/2,>1/2},\ NP = R_{\leq 0,>0}$$

lässt sich sofort

$$P \subseteq RP \subseteq BPP \subseteq PP \text{ und } RP \subseteq NP$$

ableiten. Wegen

$$ZPP = RP \cap \text{co-}RP,\ P = \text{co-}P,\ BPP = \text{co-}BPP$$

folgt weiterhin

$$P \subseteq ZPP \subseteq RP \subseteq RP \cup \text{co-}RP \subseteq BPP \text{ und } ZPP \subseteq NP \cap \text{co-}NP \ .$$

Da alle probabilistischen Klassen, die unserer generischen Definition genügen, in *PSpace* liegen, gilt insbesondere

$$PP \subseteq PSpace \ .$$

Schließlich sei an die Inklusionen

$$\Delta_2 \subseteq \Phi_2 \subseteq \Sigma_2 \cap \Pi_2 \text{ und } BPP \subseteq \Phi_2$$

erinnert. Wenn wir diese Puzzlesteine zusammensetzen, erhalten wir die „Landschaft der Komplexitätsklassen“ wie sie in Abbildung 1 dargestellt ist.

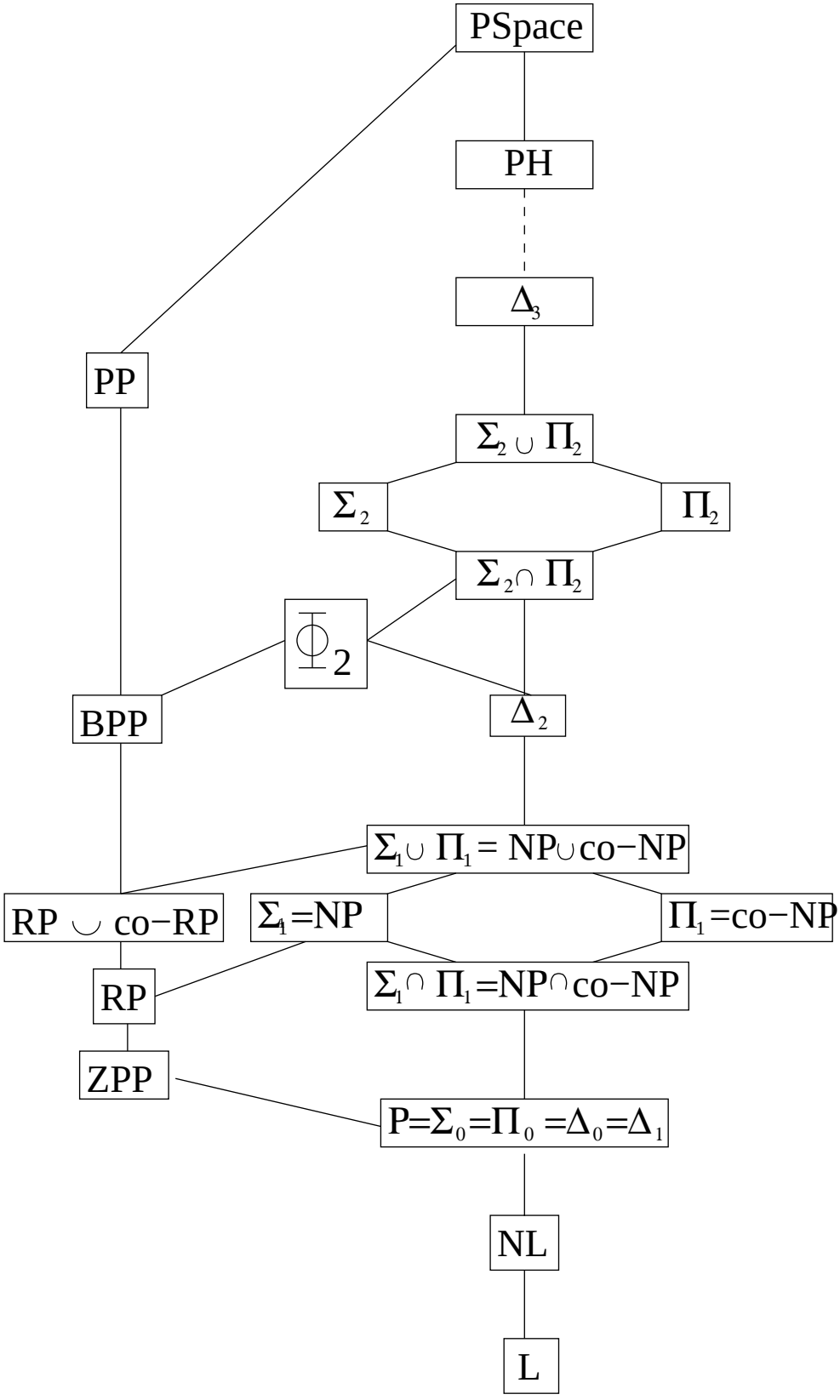

Abbildung 1: Die Landschaft der Komplexitätsklassen