

Übersicht über die probabilistischen Komplexitätsklassen

In der Vorlesung wurden die vier Komplexitätsklassen PP, BPP, RP und ZPP definiert. Die „Fehlerarten“ können auf die folgende Weise klassifiziert werden: Es gibt *zweiseitigen* Fehler (d.h., Wörter aus der Sprache L dürfen auch verworfen werden, Wörter, die nicht in L sind, dürfen auch akzeptiert werden), *einseitigen* Fehler (Wörter aus L dürfen verworfen werden, Wörter, die nicht L sind, dürfen aber nicht akzeptiert werden), sowie den *fehlerfreien* Fall (der Algorithmus darf die Antwort verweigern, wenn er aber antwortet, muss das Ergebnis stimmen). Es sollte klar sein, dass die Fehlerwahrscheinlichkeit bei zweiseitigem Fehler kleiner als $1/2$ sein muss, anderenfalls kann man auch einfach würfeln, und bei einseitigem Fehler kleiner als 1 sein muss, anderenfalls könnte man immer verwerfen. Man unterscheidet daher auch die Fälle, dass die Fehlerwahrscheinlichkeit beliebig nahe an der trivialen Schranke sein darf (hier spricht man auch von *unbeschränktem Fehler*) oder dass die Fehlerwahrscheinlichkeit einen konstanten Abstand von der trivialen Fehlerschranke haben muss, in diesem Fall spricht man von *beschränktem Fehler*. Bei dieser Klassifizierung der Fehlerarten erhält man sechs mögliche Definitionen von probabilistischen Komplexitätsklassen, die in der folgenden Tabelle dargestellt sind.

	zweiseitiger Fehler		einseitiger Fehler		fehlerfrei
unbeschränkter Fehler	PP	\supseteq	NP	\supseteq	$\text{NP} \cap \text{co-NP}$
	UI		UI		UI
beschränkter Fehler	BPP	\supseteq	RP	\supseteq	ZPP = $\text{RP} \cap \text{co-RP}$

Bei einseitigem unbeschränktem Fehler erhält man die Komplexitätsklasse NP, denn wir haben in der Vorlesung gezeigt:

Satz: Eine Sprache L ist genau dann in NP, wenn es eine polynomiell zeitbeschränkte probabilistische Turingmaschine M gibt, so dass für alle Wörter $w \in \Sigma^*$ gilt:

$$\begin{aligned} w \in L &\Rightarrow \text{Prob}(M(w) = 1) > 0, \\ w \notin L &\Rightarrow \text{Prob}(M(w) = 0) = 1. \end{aligned}$$

Analog kann man zeigen (vgl. Übungsaufgabe 7.1):

Satz: Eine Sprache L ist genau dann in $\text{NP} \cap \text{co-NP}$, wenn es eine polynomiell zeitbeschränkte probabilistische Turingmaschine M gibt, so dass für alle Wörter $w \in \Sigma^*$ gilt:

$$\begin{aligned} w \in L &\Rightarrow \text{Prob}(M(w) = 0) = 0 \wedge \text{Prob}(M(w) = 1) > 0, \\ w \notin L &\Rightarrow \text{Prob}(M(w) = 1) = 0 \wedge \text{Prob}(M(w) = 0) > 0. \end{aligned}$$

D.h., $\text{NP} \cap \text{co-NP}$ ist die Komplexitätsklasse, die die Sprachen enthält, die in polynomieller Zeit fehlerfrei mit unbeschränkter Misserfolgswahrscheinlichkeit akzeptiert werden können,

wobei ein Misserfolg die Situation ist, dass der zugehörige Algorithmus die Antwort verweigert.

Es ist naheliegend, dass die in der Tabelle angegebenen Inklusionen gelten (wobei natürlich für alle Inklusionen offen ist, ob sie echt sind oder nicht). Allerdings folgen nur wenige dieser Inklusionen direkt aus den Definitionen der Komplexitätsklassen, die übrigen Inklusionen, sowie die Aussage $ZPP = RP \cap \text{co-RP}$ mussten extra bewiesen werden. Eine wichtige Technik hierbei ist die Probability Amplification, d.h. die Wiederholung von Algorithmen, um die Fehler- bzw. Misserfolgswahrscheinlichkeit zu verringern. Wir haben uns auch davon überzeugt, dass bei Anwendung der vorgestellten Variante von Probability Amplification auf Algorithmen mit unbeschränktem Fehler exponentiell viele Iterationen nötig sind, um Algorithmen mit beschränktem Fehler zu erhalten. Dies deutet nochmal den Unterschied zwischen den Komplexitätsklassen mit beschränktem und unbeschränktem Fehler an.