Orakel-Quanten-Turing-Maschinen, Quanten-NP und die polynomielle Hierarchie

Markus Latte

30. April 2003
und
7. Mai 2003

Zusammenfassung

Orakel-Turingmaschinen spielen in der Komplexitätstheorie eine entscheidende Rolle (z.B. bei der polynominellen Hierarchie), da Orakel die Berechnungsstärke von Komplexitätsklassen in bestimmter Weise erhöhen können. Es stellt sich die Frage, wie sich Klassen für klassische TMs und QTMs unter Orakel verhalten. Exponentielle Trennungen gewisser Klassen werden vorgestellt. Im zweiten Teil brachten wir das Analogon von **NP** in der Quantenwelt und weisen nach, dass dieses außerhalb der polynominellen Hierarchie liegen muss, falls letzte nicht kollabiert.

1 Orakel – oder: woher kommt das Wissen?

Ein (klassisches) Orakel ist eine Sprache, deren Wortproblem-Lösung einer TM zu Verfügung gestellt wird: Mittels eines Übergang in einen dedizierten Zustand will die eine TM die Zugehörigkeit eines Worts auf einem ausgezeichnetem Band durch das Orakel gelöst wissen. Die Antwort erfolgt über zwei ebenfalls gesonderte Zustände. Das Orakel wird bei Maschinen im Exponenten angegeben. Diese Erweiterung verändert die Abzählbarkeit der TMs (oder von Teilklassen) offenbar nicht. Für den Zeit- und Platzverbrauch wird der einer möglichen TM, die das Wortproblem lösen kann, NICHT berücksichtigt. Durch die Einbeziehung von Orakeln kann die Berechnungsstärke von Klassen erhöht werden. Bekanntes Beispiel ist das offene Probelm $\mathbf{P} \neq \mathbf{NP}$. Für die entsprechenden Klassen mit Orakel lässt sich einerseits ein Orakel X angeben mit $\mathbf{P}^X = \mathbf{NP}^X$, anderseits via Diagonalisierung auch $\mathbf{P}^Y \neq \mathbf{NP}^Y$ für ein gewisses Orakel Y (siehe [6]). Letztes

Beweisverfahren werden wir u.a. zur Trennung von $\mathbf{QP}^Z$ und $\mathbf{NP}^Z$ bzw. $\mathbf{co-NP}^Z$ verwenden.
Neben der Definition von Orakeln als Sprachen, können auch Funktionen verwendet werden. Hierzu sei eine Funktion $f : \{0,1\}^n \to \{0,1\}^m$ gegeben. Daraus lässt sich eine Sprache $L_f = \{x\#y \mid \pi_{\sigma(y)}(f(x)) = 1\}$, wobei σ eine triviale Bijektion von $\{0,1\}^*$ in die natürlichen Zahlen darstellt (siehe unten), ableiten, aus der bei gegebenem m die TM die eigentliche Funktion mit linearem Overhead bestimmen kann.
Die Erweiterung auf QTMs erfolgt kanonisch unter Beachtung der Reversibilität. Um einen Vergleich zwischen den Komplexitätsklassen über TMs mit denen über QTMs zu gewährleiten, beschränken wir uns auf klassische Orakel, d.h. Orakel für QTMs sollen auf eine reine Anfrage mit einem reinen Zustand antworten. (Denkbar wäre auch eine Erweiterung, in der das Orakel auf einen reinen Zustand auch mit einer beliebigen Superpostion antworten darf.)

2 Klassentrennung

2.1 Definitionen und Notationen

Neben den Klassen $\mathbf{P}$, $\mathbf{NP}$, $\mathbf{co-NP}$ betrachten wir noch $\mathbf{QP}$, als die Menge derjenigen Sprachen, die jeweils von einer QTM in polynomineller Zeit entschieden werden können. Entsprechende notationelle Erweiterung für Orakel werden verwendet.
Im Folgenden sei das Alphabet $\Sigma = \{0,1\}$. Von Σ^* nach $\mathbb{N}$ lässt sich eine Bijektion σ definieren zu $\sigma(x) = bin(1x) - 1$, wobei "$1x$" den um das Zeichen "1" am Anfang verlängerten String x und bin eine Funktion, die ein Wort als Binärzeichen verstanden in ihren Wert übersetzt, darstellen. Die Umkehrfunktion bezeichnen wir mit ρ.
Weiter sei $(M_i)_{i\in\mathbb{N}}$ eine Aufzählung aller deterministischen TM (DTM) und $(N_i)_{i\in\mathbb{N}}$ eine Aufzählung aller existentiellen TM (ETM [1]). Bei beiden Aufzählungen sind jeweils die TMs gemeint, bei denen ein Orakel eingesetzt werden kann.
Sei $\alpha : \mathbb{N} \to \mathbb{N}$ eine beliebige Funktion, so dass jedes Urbild eine unendliche Kardinalität hat. $\wp : \mathbb{N} \to \mathbb{N}$ stellt eine hyperexponentiell wachsende Funktion dar, die rekursiv definiert ist zu:

[1] Die Bezeichnung nichtdeterministischen TM (NTM) soll hier nicht verwendet werden, da diese nur anzeigt, dass die Übergangsrelation nicht rechtseindeutig sein muss. Die in der Literatur sogenannte NTM, wird hier mit ETM bezeichnet, um den Unterschied zur generellen und alternierenden TM, die ebenfalls über NTMs definiert sind, aufzuzeigen.

$\wp(1) := 1$ und $\wp(n+1) = 2^{\wp(n)}$ für alle $n \in \mathbb{N}$.

2.2 QTM schlägt DTM und ETM via Orakel

Für jede Menge $X \subseteq \Sigma^*$ sagen wir, sie habe die Eigenschaft $B(X)$ falls gilt:

$$\forall n \in \mathbb{N}. X \cap \Sigma^n = \emptyset \not\leftrightarrow |X \cap \Sigma^n| = 2^{n-1}.$$

Weiter sei

$$S_X := \{1^n \mid X \cap \Sigma^n = \emptyset\}.$$

Aus [3] übernehmen wir die Existenz einer QTM mit einem Orakel für X, welche die Sprache S_X in linearer Zeit entscheidet, falls $B(X)$ gilt.

Satz 1 *Es gibt ein Orakel Y, so dass es eine entsprechende Orakel-QTM gibt, die das Wortproblem der Sprache S_Y in linearer Zeit entscheidet, wogegen jede ETM bereits für unendlich viele Wörter exponentielle Zeit benötigt.*

Beweis. Mittels Diagonalisierung.
Im ersten Schritt konstruieren wir eine Familie von Sprachen $(Y_n)_{n \geq 1}$ mit $Y_1 := \emptyset$.
Sei $n \geq 1$. Lasse die existentielle TM mit der Nummer $N_{\alpha(n)}$ mit Orakel Y_n auf der Eingabe $1^{\wp(n)-1}$ maximal $2^{\wp(n)-1}$ Schritte laufen. Unterscheide nach der Menge der dabei erzeugten Traces.

- Falls es einen Berechungspfad mit mehr als $2^{\wp(n)-1}$ Schritten gibt oder die TM die Eingabe verwirft, setze $Y_{n+1} := Y_n$.
- Anderfalls haben alle Pfade höchstens $2^{\wp(n)-1}$ Schritte benötigt und ein Pfad hat akzeptiert. Sei Q die Menge aller Fragen, die ein (beliebiger) akzeptierender Pfad an das Orakel gestellt hat. Dann gibt es eine davon disjunkte Menge $R \subset \Sigma^{\wp(n)}$ der Kardinalität $2^{\wp(n)-1}$, da alle Wörter in Q nicht länger als $2^{\wp(n)-1}$ sind und $|Q| \leq 2^{\wp(n)-1}$. Setze $Y_{n+1} := Y_n \cup R$.

Offenbar gilt $Y_n \subseteq Y_{n+1}$ für alle $n \geq 1$. Die gesuchte Orakelsprache Y legen wir zu $\bigcup_{n \geq 1} Y_n$ fest.
Es ist ersichtlich, dass die Eigenschaft $S(Y)$ gilt. Somit gibt es eine QTM, die das Wortproblem in linearer Zeit löst.
Es bleibt so zeigen, dass jede ETM exponentielle Zeit für das Wortproblem benötigt. Wähle eine (beliebige) ETM, von der behauptet wird, sie löse das Wortproblem. Dann gibt es ein i, so dass N_i obige TM darstellt. Wir zeigen nun, dass für jedes der unendlich vielen $n \in \alpha^{-1}(i)$ N_i^Y für die Eingabe $1^{\wp(n)}$ mehr als $2^{\wp(n)-1}$ Schritte braucht. Dazu unterscheide folgende Fälle:

Ein Pfad von $N_i^{Y_n}$ braucht mehr als $2^{\wp(n)-1}$ Schritte. Nach Konstruktion sind Y und Y_n gleich, bzgl. der Wörter, die kürzer als $\wp(n+1) = 2^{\wp(n)}$ sind, da $Y_{n+1} = Y_n$. Somit ist das Verhalten von $N_i^{Y_n}$ und N_i^Y auf den ersten $2^{\wp(n)}$ Schritten gleich. Daher hat auch N_i^Y einen Pfad mit mehr als $2^{\wp(n)-1}$ Schritten.

$N_i^{Y_n}$ verwirft die Eingabe innerhalb von $2^{\wp(n)-1}$ Schritten. Analog zu oben verwirft auch N_i^Y die Eingabe. Da N_i^Y die Sprache S_Y entscheidet, gilt $1^{\wp(n)} \notin S_Y$, also $Y \cap \Sigma^{\wp(n)} \neq \emptyset$. Nach Konstruktion von Y hat $N_{\alpha(n)}$ die Eingabe akzeptiert, wobei $\alpha(n) = i$. Widerspruch.

$N_i^{Y_n}$ akzeptiert die Eingabe innerhalb von $2^{\wp(n)-1}$ Schritten. Wieder akzeptiert auch N_i^Y, da zum einen nur Wörter, die nicht länger als $2^{\wp(n)-1}$ sind, betrachtet werden und $Y_{n+1} \setminus Y_n$ nur Wörter enthält, die von einem gewissen akzeptierenden Pfad nicht gefragt wurden. Da N_i^Y die Sprache S_Y entscheidet, gilt $1^{\wp(n)} \in S_Y$, also $Y \cap \Sigma^{\wp(n)} = \emptyset$. Nach Konstruktion von Y hat $N_{\alpha(n)}$ die Eingabe verworfen, wobei $\alpha(n) = i$. Widerspruch.

Folglich kann es nur sein, dass $N_i^{Y_n}$ mehr als $2^{\wp(n)-1}$ Schritte braucht (Sherlock Holmes' Prinzip oder via "ex falso quodlibet"). □

Satz 2 *Es gibt ein Orakel Y mit*

$$\mathbf{QP}^Y \nsubseteq \mathbf{NP}^Y \cup \mathbf{co}-\mathbf{NP}^Y.$$

Beweis. Als Orakel benutzen wir das aus obigem Beweis. Aus diesem folgt unmittelbar $S_Y \in \mathbf{QP}^Y$ und $S_Y \notin \mathbf{NP}^Y$. Also $\mathbf{QP}^Y \nsubseteq \mathbf{NP}^Y$.
Da die Klasse **QP** unter Komplement abgeschlossen ist, gilt ebenfalls $\overline{S_Y} \in \mathbf{QP}^Y$ und $\overline{S_Y} \notin \mathbf{co}-\mathbf{NP}^Y$. Wähle $Z := S_Y \cup \{0^n \mid 1^n \notin S_Y\}$. Dann gilt:

$Z \in \mathbf{QP}$: Konstruiere folgende QTM: falls Eingabe aus 1^*, dann arbeite wie QTM aus obigem Beweis. Falls Eingabe aus 0^*, wandle 0 in 1 um und simuliere die zu oben komplementäre QTM. Andernfalls verwerfe Eingabe.

$Z \notin \mathbf{NP}^Y \cup \mathbf{co}-\mathbf{NP}^Y$: Ansonsten wäre oBdA. (bzw. konvertiere Eingabe vorher) $Z \in \mathbf{NP}^Y$. Dies würde $S_Y \in \mathbf{NP}^Y$ implizieren, denn die Bestimmung, ob die Eingabe in 1^* liegt, ist trivial. Widerspruch.

□

Anmerkung. Die Sprache S_Y ist im folgendem Sinn dünn besetzt: für alle n enthält S_Y höchstens ein Wort einer Länge zwischen n und $2^n - 1$.

Genauer: die Menge der Längen ist eine Teilmenge von $2^{\wp(\mathbb{N})}$, also gilt: $\forall n \in \mathbb{N}.\exists! i \in \mathbb{N}.n \leq 2^{\wp(i)} < 2^n$. Daher kann die Sprache von einer TM in fast allen Fällen in linearer Zeit entschieden werden: teste Eingabe, ob diese in 1^*.
Unter Verwendung der Bijektion σ kann die Sprache S_X exponentiell gepackt werden:

$$V_X := \{x \in \Sigma^* \mid X \cap \Sigma^{\sigma(x)} = \emptyset\}.$$

Also kann V_X von einer QTM in exponentieller Zeit durch Umwandlung der quasi-binären Eingabe in ihre unäre Darstellung entschieden werden.

Satz 3 *Es gibt ein Orakel Z und eine Sprache, so dass das Wortproblem auf einer entsprechenden Orakel-QTM in exponentieller Zeit gelöst werden kann, während jede deterministische TM mit Orakel Z doppelt-exponentielle Zeit dafür braucht.*

Beweis. Das Orakel wird — wie oben — wieder stufenweise aus einer Folge $(Z_n)_{n\geq 3}$ mit $Z_3 = \emptyset$ konstruiert. Zusätzlich wird parallel eine Liste von "fehlerhaften" TM mitgeführt. Die komplementäre Menge wird "funktionierend" genannt. Anfangs sind alle TM funktionierend ("in dublio pro reo").
Sei $n \geq 3$. Lasse die DTMs $M_1, \ldots, M_n$ auf den Eingaben $\rho(\{1, \ldots, n\})$ unter Verwendung des Orakels Z_n für maximal $2^{n-1}/n^2$ Schritt laufen. Q sei die Menge der Fragen, die dabei an das Orakel gestellt wurden. Offenbar ist die Kardinalität von Q durch 2^{n-1} nach oben beschränkt.
Z_{n+1} wird wie folgt definiert:

- Wenn keine der funktionierenden TMs innerhalb von $2^{n-1}/n^2$ Schritten bei Eingabe $\rho(n)$ hält, setzte $Z_{n+1} := Z_n$.
- Sonst gibt es ein kleinstes i, so dass M_i funktioniert und innerhalb von $2^{n-1}/n^2$ Schritten bei Eingabe $\rho(n)$ hält.
  - Falls $M_i^{Z_n}$ die Eingabe verwirft, setze $Z_{n+1} := Z_n$.
  - Setze sonst $Z_{n+1} := Z_n \cup R$, wobei $R \subset \Sigma^n$ mit $|R| = 2^{n-1}$ und $R \cap Q = \emptyset$.

  Füge M_i in die Liste der "fehlerhaften" TM ein.

Als Orakel wählen wir $Z = \bigcup_{n\geq 3} Z_n$. Somit kann die Sprache V_Z von einer QTM unter Verwendung des Orakels Z in exponentieller Zeit entschieden werden.

Wir zeigen nun, dass jede deterministische TM M_i^Z, die das Wortproblem V_Z entscheidet, nicht als "fehlerhaft" eingestuft wurde. Angenommen, dies wäre nicht der Fall, dann wurde für ein gewisses n die TM bei der Konstruktion von Z_{n+1} als fehlerhaft festgelegt. Die Konstruktion von Z stellt sicher, dass $M_i^{Z_n}$ die Eingabe $\rho(n)$ verwirft gdw. es kein Wort der Länge n in Z gibt, also $\rho(n) \in V_Z$. Somit entscheidet $M_i^{Z_n}$ V_Z nicht. Weiter verhalten sich M_i^Z und $M_i^{Z_n}$ auf der Eingabe $\rho(n)$ identisch, da sich Z und Z_n höchstens in Wörter unterscheiden, die $M_i^{Z_n}$ nicht an ihr Orakel gestellt hat. (Beachte: $2^{m-1}/m^2 \geq 2^{n-1}/n^2$ für alle $m \geq n \geq 3$.) Widerspruch.
Sei M_i^Z nun eine TM, die V_Z entscheidet. Wähle n_0 so, dass alle M_j mit $j < i$, die jemals als "fehlerhaft" deklariert werden, bis zum Schritt n_0 auch als solche deklariert werden. Sei $n > n_0$. Dann kann M_i beim Schritt n nicht als "fehlerhaft" festgestellt werden. Da auch keine Maschine mit niedrigerem Index in diesem Schritt als "fehlerhaft" bezeichnet werden kann, braucht $M_i^{Z_n}$ auf der Eingabe $\rho(n)$ mehr als $2^{n-1}/n^2$ Schritte. Mit gleicher Argumentation wie oben braucht M_i^Z auch mehr als $2^{n-1}/n^2$ Schritte. Letztlich ist n exponentiell in der Länge von $\rho(n) = x$. Also benötigt die TM M_i^Z bei Eingabe x mehr als

$$\frac{2^{c^{|x|}-1}}{c^{2|x|}} \in \mathcal{O}\left(b^{\left(c^{|x|}\right)}\right)$$

Schritte (für gewisse $c > 1$ und $1 < b < 2$). □

Anmerkung. Interessant wäre zu wissen, ob die Inklusion $\mathbf{P} \subseteq \mathbf{QP}$ echt ist oder nicht. Doch diese Frage hat die gleiche Härte wie $\mathbf{P} \neq \mathbf{PSPACE}$, da $\mathbf{QP} \subseteq \mathbf{PSPACE}$ nachgewiesen werden konnte (Bernstein und Vazirani, Verweis in [1]).

3 QTM schlägt PTM via Fehlerwahrscheinlichkeit

Im letzten Abschnitt wurde eine Trennung von Orakel-Komplexitätsklassen mit einer Fehlerwahrscheinlichkeit von 0 vorgestellt. Wir setzen uns nun mit der Frage auseinander, wie groß der Aufwand für eine QTM und PTM[2] jeweils ist, eine zufällige Funktion einer von zwei Klassen bei gegebener maximalen Fehlerwahrscheinlichkeit zuzuordnen.

[2] probabilistische TM

3.1 FOURIER-Transformation

Lemma 1 *Gegeben seien n QuBits im Zustand* $|\psi\rangle \in \mathbb{C}^{2^n}$*, auf denen jeweils die Hadamard-Matrix*

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

angewendet wird. Als Ergebnis stellt sich

$$F(|\psi\rangle) = 2^{-n/2} \sum_{x \in \{0,1\}^n} (-1)^{\psi \cdot x} |x\rangle$$

ein.

Beweis. Für $n = 1$: Fallunterscheidung. Sonst:

$$\begin{aligned} \bigotimes_{i=1}^{n} H &= 2^{-n/2} \bigotimes_{i} \sum_{x_i \in \{0,1\}} (-1)^{\psi_i x_i} |x_i\rangle \\ &= 2^{-n/2} \sum_{x \in \{0,1\}^n} \prod_{i=1}^{n} (-1)^{\psi_i x_i} |x\rangle \\ &= 2^{-n/2} \sum_{x \in \{0,1\}^n} (-1)^{\psi \cdot x} |x\rangle \end{aligned}$$

□

3.2 Ist eine Funktion invariant unter einer gewissen exor-Maske?

Definition 1 (exor-Invarianz-Problem) *Sei eine Funktion* $f : \{0,1\}^n \to \{0,1\}^m$ *mit* $m \geq n$ *gegeben, von der behauptet wird, es gebe ein* $s \in \{0,1\}^n$*, so dass* $\forall x \neq x'. f(x) = f(x') \leftrightarrow x' = x \oplus s$ *hält*[3]*. Falls* $s = 0$*, ist f offenbar injektiv, ansonsten bildet sie je genau zwei Werte auf einen Funktionswert ab. Ziel ist es nun, die Fälle zu separieren und im zweiten Fall die exor-Maske s zu bestimmen.*

Anmerkung. Die Intention des folgenden Beweises in [8] ist der Nachweis, dass das exor-Problem auf einer QTM mit einer *Null*-Fehlerwahrscheinlichkeit lösbar ist. Jedoch ist der Beweis nicht nachvollziehbar. Daher wird hier eine Modifikation vorgestellt.

[3] $\oplus$: bitweises exklusiv-Oder

Satz 4 *Das exor-Invarianz-Problem ist auf einer QTM in Zeit*

$$\mathcal{O}\left(nT_f(n) + G(n)\right)$$

lösbar, wobei $T_f(n)$ die Zeit für eine Bestimmung von f bei einer Eingabe der Länge n ist und $G(n)$ die Zeit zum Lösen eines $n \times n$-Gleichungssystems über dem Körper $\mathbb{F}_2$ ist. Das Ergebnis ist im Wesentlichen von der Form $|\psi\rangle = \alpha|0^n\rangle + \beta|s\rangle$, wobei $\beta \neq 0$ und s die exor-Maske darstellt, wenn f nicht injektiv ist. Sonst ist $|\psi\rangle = |0^n\rangle$.

Beweis. Als Erstes führen wir die doppelte FOURIER-Transformation (`FfF`) aus:

1. Bestimme $F(|0^n\rangle) = 2^{-n/2} \sum_{x \in \{0,1\}^n} |x\rangle$.
2. Wende darauf die Funktion $\lambda|x\rangle.|x\rangle|f(x)\rangle$ an. Als Ergebnis stellt sich $2^{-n/2} \sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle$ ein.
3. F auf x angewandt liefert $2^{-n} \sum_{x,y \in \{0,1\}^n} (-1)^{x \cdot y} \; |y\rangle|f(x)\rangle$.

Im Folgenden wollen wir die Ausprägung der $|y\rangle|f(x)\rangle$ untersuchen.

f **injektiv:** Alle Konfigurationen der Form $|y\rangle|f(x)\rangle$ sind mit einer Phase von $\pm 2^{-n}$ in der Superpostion vertreten.

f **ist nicht injektiv:** Dann gibt es ein $s \neq 0$ mit $\forall x \neq x'.f(x) = f(x') \leftrightarrow x' = x \oplus s$. Für alle x und y sind die Konfigurationen $|y\rangle|f(x)\rangle$ und $|y\rangle \otimes |f(x \oplus s)\rangle$ identisch. Ihre Amplitude ist somit

$$\begin{aligned} \alpha(x,y) &= 2^{-n}((-1)^{xy} + (-1)^{(x \oplus s)y}) \\ &= 2^{-n} \begin{cases} (-1)^{xy} + (-1)^{xy}(-1)^0 & \text{falls } y \cdot s \equiv 0 \pmod 2 \\ (-1)^{xy} + (-1)^{xy}(-1)^1 & \text{sonst} \end{cases} \\ &= \begin{cases} 2^{-n+1} \cdot (-1)^{xy} & \text{falls } y \cdot s \equiv 0 \pmod 2 \\ 0 & \text{sonst} \end{cases} . \end{aligned}$$

Dabei ist zu beachten, dass $\oplus$ der Addition im $\mathbb{F}_2^n$ entspricht.

Angenommen, man wiederhole den Algorithmus `FfF` n mal und messe jeweils die Endkonfiguration. Dabei bezeichne $y_1, \ldots, y_n$ die Folge der Messwerte für den "y"-Part.
Auf diesen Daten lassen wir folgenden (deterministischen) Algorithmus (`solve`) laufen:

```
Sei M die Matrix mit den Zeilen y_1,...,y_n.
if (rang(M) == n-1)
{
  Bestimme den eindeutigen Lösungsvektor s' ≠ 0 zu M · s' = 0
  if (f(0^n) == f(s'))
    return s';
  else
    return 0^n;
}
else
  return 0^n;
```

Es ist offensichtlich, dass das Ergebnis z nur ungleich 0^n sein kann, wenn die Funktion nicht injektiv ist und $z = s$ gilt.
Den gesuchten Algorithmus erhalten wir, durch n-fache Wiederholung von `FfF` bei gleicher Eingabe. Die "y"-QuBits werden auf `solve` in der Formulierung für eine QTM angewendet. Als Endzustand ergibt sich im Wesentlichen $|\psi\rangle = \alpha|0^n\rangle + \beta|s\rangle$ bis auf weitere Informationen, die aufgrund der Resersibilität bestehen.
Es bleibt die Behauptung über die Ausgangskonfiguration zu zeigen.

f **injektiv:** $\beta = 0$. Würde `solve` einen anderen Wert als $|0^n\rangle$ liefern, wäre f nicht injektiv.

f **ist nicht injektiv:** Es ist nachzuweisen, dass es eine Messfolge gibt, so dass der Rang von M $n-1$ ist. Nach obiger Fallunterscheidung stehen *alle* Messwerte senkrecht auf s. Der zu $span(\{s\})$ orthogonale Raum der Diminsion $n-1$ liefert aber nun die gesuchten $n-1$ Vektoren. Also $\beta \neq 0$.

$\square$

Satz 5 *Sei O ein Orakel, das wie folgt konstruiert ist: für jedes $n \in \mathbb{N}$ wähle man (gleichverteilt) ein $b(n) \in \{0,1\}$ und $s(n) \in \{0,1\}^n$. Das Orakel O "berechne" eine Funktion $f_n : \{0,1\}^n \to \{0,1\}^n$, die eine zufällige Bijektion ist, falls $b(n) = 0$, und andernfalls eine zufällige Funktion mit $f_n(x) = f_n(x')$ gdw. $x = x' \oplus s(n)$. Dann ist die Wahrscheinlichkeit, dass eine beliebige PTM, die maximal $2^{n/4}$ Fragen an das Orakel stellt, $b(n)$ richtig bestimmt nach oben durch $0.5 + 2^{-n/2}$ begrenzt.*

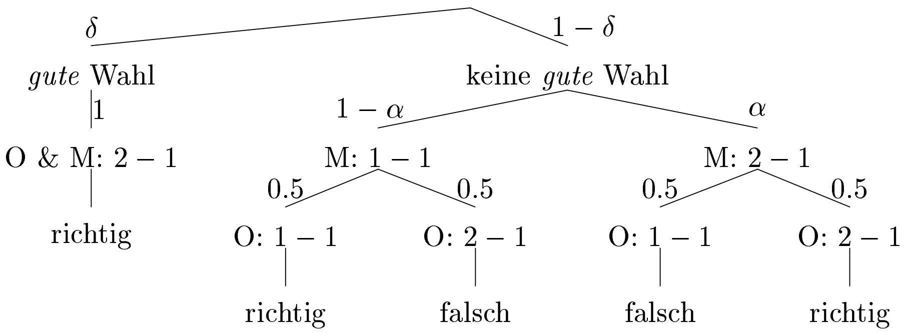
Beweis. Sei M nun eine solche PTM. Folgende Aussagen lassen sich treffen:

1. Je mehr Fragen gestellt werden, desto besser ist die Wahrscheinlichkeit für eine korrekte Antwort. Also stelle M genau $2^{n/4}$ Fragen.

2. Aus der Antwort des Orakels kann nur die Gleichkeit oder Verschiedenheit zu vorhergehenden Antworten festgestellt werden. Die konkreten Werte lassen keine Schlüsse auf das Orakel zu, da die durch das Orakel dargestellte Funktion zufällig ist.

3. OBdA. können wir annehmen, dass M zuerst eine Menge von Fragen bestimmt, diese dann an das Orakel stellt, die Antworten auf Kollision überprüft und in diesem Fall mit "$b(n) = 1$" antwortet. Ansonsten rät M mit Wahrscheinlichkeit α obige Antwort.

Eine Menge von Fragen, die eine Kollision der Funktionswerte zur Folge hat, nennen wir *gut*. Sei δ nun die Wahrscheinlichkeit, dass eine fixierte Wahl von $2^{n/4}$ Fragen *gut* ist.



Dann ist die Wahrscheinlichkeit für eine korrekte Antwort von M

$$\underbrace{\delta}_{\textit{gute}\text{ Wahl}} + \underbrace{(1-\delta)}_{\text{keine }\textit{gute}\text{ Wahl}} \left(\underbrace{(1-\alpha)\cdot 0.5 + \alpha \cdot 0.5}_{M \text{ und } O \text{ "synchron"}} \right) \leq 0.5 + \delta$$

Es bleibt δ zu bestimmen. Die Wahrscheinlichkeit für eine Folge *guter* Fragen der Größe k ist gleich der Summe über alle $j \leq k$ der Wahrscheinlichkeit, dass die j-te Frage die erste *gute* Frage ist. Anstelle diese Wahrscheinlichkeit zu bestimmen, konzentrieren wir uns auf die der Kollision. Bei j Fragen gibt es

$$2^n - \sum_{i=1}^{j}(i-1) = 2^n - \frac{j(j-1)}{2}$$

Möglichkeiten für kollisionfreie Antworten.

Die Wahrscheinlichkeit für eine Folge von k Fragen, wobei die letzte die erste *gute* sei, ist

$$\frac{k}{2^n - (k-1)(k-2)/2}.$$

Nun zur Fragenfolge von M (mit $j = 2^{n/4}$). Die Wahrscheinlichkeit, eine *gute* Menge von Fragen zu wählen, lässt sich mit Obigem abschätzen zu:

$$\begin{aligned}
\delta &= \sum_{k=1}^{j} \frac{k}{2^n - (k-2)(k-1)/2} \\
&\leq \sum_{k=1}^{j} \frac{k}{j^4 - j^2} \\
&\leq \frac{1}{j^4 - j^2} \frac{j(j+1)}{2} \\
&\leq \frac{2 \cdot 2^{n/2}}{2 \cdot 2^n} \\
&= 2^{-n/2}
\end{aligned}$$

□

4 Quanten-NP ist hart für die Polynominelle Hierarchie

Zunächst wollen wir die polynominelle Hierarchie (**PH**) betrachten. Sie wird definiert durch ($k \in \mathbb{N}$):

$$\begin{array}{ccc}
\Sigma_0 = \mathbf{P} & \Pi_0 = \mathbf{P} & \Delta_0 = \mathbf{P} \\
\Sigma_{k+1} = \mathbf{NP}^{\Sigma_k} & \Pi_{k+1} = \mathbf{co}\text{-}\Sigma_{k+1} & \Delta_{k+1} = \mathbf{P}^{\Sigma_k}
\end{array}$$

$$\mathbf{PH} = \bigcup_{k \geq 0} \Sigma_k = \bigcup_{k \geq 0} \Pi_k = \bigcup_{k \geq 0} \Delta_k$$

Satz 6 (Eigenschaften der PH) *Es gelten folgende Aussagen:*

- $\mathbf{NP} = \Sigma_1$
- $\mathbf{co} - \mathbf{NP} = \Pi_1$
- $\Delta_k = co - \Delta_k$
- $\Sigma_k \cup \Pi_k \subseteq \Delta_{k+1}$
- $\Delta_k \subseteq \Sigma_k \cap \Pi_k$

- $P^{\Delta_k} = \Delta_k$

Beweis. Beweis in [2], ab Seite 180. $\square$

Eine offene Frage ist, ob die polynominelle Hierarchie kollabiert, d.h. $\exists i.\Sigma_i = \Sigma_{i+1}$. Davon wird aber nicht ausgegangen.

Satz 7 (Eigenschaften bei einem Kollaps der PH) *Es gelten folgende Aussagen:*

- $\Sigma_k = \mathbf{PH} \Rightarrow \Sigma_k = \Pi_k$
- $\Sigma_{k+1} = \Pi_{k+1} \Rightarrow \Sigma_{k+1} = \Sigma_{k+2}$
- $\Sigma_k = \Sigma_{k+1} \Rightarrow \Sigma_k = \mathbf{PH}$

Beweis. Beweis in [6], Übungsblatt 5. $\square$

Eine kleine Grafik verdeutlicht die letzten beiden Sätze:

$$
\begin{tikzcd}
& {} \arrow[d, dash, dotted] & \\
& \Sigma_2 \cup \Pi_2 \arrow[dl, dash] \arrow[dr, dash] & \\
\Sigma_2 \arrow[dr, dash] & & \Pi_2 \arrow[dl, dash] \\
& \Sigma_2 \cap \Pi_2 \arrow[d, dash] & \\
& \Delta_2 \arrow[d, dash] & \\
& \Sigma_1 \cup \Pi_1 \arrow[dl, dash] \arrow[dr, dash] & \\
\Sigma_1 \arrow[dr, dash] & & \Pi_1 \arrow[dl, dash] \\
& \mathbf{P} &
\end{tikzcd}
$$

Die Definition der Klasse **NP** lässt sich für die Quantenwelt erweitern zu:

Definition 2 (NQP) *Eine Sprache L ist in* **NQP** *gdw. es eine QTM Q und ein Polynom p gibt, so dass*

$$x \in L \Leftrightarrow Pr[Q \text{ akzeptiert } x \text{ in max. } p(|x|) \text{ Schritten}] \neq 0.$$

Ein offenbar für diese Klasse vollständiges Problem ist die Sprache

$$QAP := \{\langle M, x, 0^t \rangle \mid M \text{ Kodierung einer QTM, die } x \text{ mit einer Wahrscheinlichkeit } \neq 0 \text{ in max. } t \text{ Schritte akzeptiert}\},$$

da nach [7] eine QTM mit polynominellen Laufzeitverlust simuliert werden kann.

Definition 3 *Für jede Sprache $L \subseteq \Sigma^*$ und jedes $x \in \Sigma^*$ sei $L_x = \{y \in \Sigma^* \mid \langle x, y \rangle \in L\}$. Ein Funktion $f : \Sigma^* \to \mathbb{Z}$ ist in* **GapP**[4] *gdw. es eine*

[4] gap, engl.: Abstand

Sprache $L \in \mathbf{P}$ und eine $k \in \mathbb{N}$ gibt, so dass

$$f(x) = \frac{|\Sigma^{n^k} \cap L_x| - |\Sigma^{n^k} \setminus L_x|}{2}$$

mit $n = |x|$ gilt.

Die Menge L_x kann als die Menge der Zeugen für die akzeptierenden Berechnungspfade einer NTM, die eine Sprache in **NP** entscheidet, bei einer Eingabe x angesehen werden. Eine Funktion $f(x)$ stellt nun die halbe Bilanz zwischen akzeptierenden und ablehnenden Berechungspfade dar.

Satz 8 *Für jede Funktion $f \in \mathbf{GapP}$ gibt es eine QTM Q mit polynominellen Zeitverbrauch und ein Polynom p, so dass für alle Wörter x der Länge n*

$$Pr[Q \text{ akzeptiert } x] = \frac{f(x)^2}{2^{p(n)}}$$

gilt. Genauer gesagt, hat Q für jedes Wort eine eindeutige akzeptierende Konfiguration mit der Wahrscheinlichkeitsamplitude[5] $-f(x)/2^{p(n)/2}$.

Beweis. Sei $f \in \mathbf{GapP}$, $k \in \mathbb{N}$ und $L \in \mathbf{P}$, so dass für alle x der Länge n

$$f(x) = \frac{|\Sigma^{n^k} \cap L_x| - |\Sigma^{n^k} \setminus L_x|}{2}$$

gilt. Sei weiter M eine DTM, die L in polynomineller Zeit entscheidet. Für ein beliebiges, aber festes, Wort x sei $n = |x|$ und $m = n^k$. Die zu konstruierende QTM Q arbeitet auf der Eingabe x und wird ein zusätzliches Band mit $m+1$ QuBits benötigen. Die Konfiguration sieht im Wesentlichen wie folgt aus:

$$|x, y, b\rangle,$$

wobei y ein Vektor aus QuBits der Länge n und b ein einzelnes QuBit ist. Zur vollständigen Beschreibung der Konfiguration sind noch weitere Informationen notwendig, die hier nicht aufgezeigt werden sollen, da sie aus obiger "Teil"-Konfiguration ableitbar oder nicht von Interesse sind, da sie reproduzierbare Zwischenberechnungen der QTM darstellen.
Anfangs sind $y = 0$ und $b = 0$. Auf dem "y"-Part wird als erstes die Fourier-Transformation angewendet mit dem Ergebnis:

$$2^{-m/2} \sum_{y \in \{0,1\}^m} |x, y, 0\rangle.$$

[5] Die Wahrscheinlichkeit ist identisch mit dem Betragsquadrat der Wahrscheinlichkeitsamplitude (hier: der Koeffizient eines Basiselements in einer Superposition).

Auf der Eingabe $\langle x, y\rangle$ wird M mittels eine zusätzlichen Arbeitsbandes simuliert. Das Ergebnis sei $b_y \in \{0,1\}$, wobei der Wert 1 die Akzeptanz von M bedeute. Die Konfiguration der QTM befindet sich dann in der Superpostion

$$2^{-m/2} \sum_{y\in\{0,1\}^m} |x, y, b_y\rangle.$$

Auf den letzten $m+1$ QuBits, bestehend aus y und b_y, wird erneut die FOURIER-Transformation angewendet:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\frac{1}{2^m} \sum_{y\in\{0,1\}^m} \sum_{y'\in\{0,1\}^m} \sum_{b'\in\{0,1\}} (-1)^{y\cdot y' + b_y \cdot b'} |x, y', b'\rangle.$$

Betrachte nun den Koeffizient von $|x, 0, 1\rangle$ in $|\psi\rangle$:

$$\begin{aligned} \langle x, 0, 1|\psi\rangle &= \frac{1}{\sqrt{2}}\frac{1}{2^{m/2}} \sum_{y\in\{0,1\}^m} (-1)^{y\cdot 0 + b_y \cdot 1} \\ &= \frac{1}{\sqrt{2}}\frac{1}{2^{m/2}} \sum_{y\in\{0,1\}^m} (-1)^{b_y} \\ &= -\frac{1}{\sqrt{2}}\frac{1}{2^{m/2}}\frac{f(x)}{2} \end{aligned}$$

Der letzte Schritt der QTM ergibt sich aus der Superpostion der deterministischen Entscheidung: Akzeptiere die Eingabe gdw. die letzten $m+1$ Bits (nicht QuBits!) $0^m, 1$ sind. Diese eindeutige Endkonfiguration hat die behauptete Wahrscheinlichkeitsamplitude

$$-\frac{1}{2^{p(n)/2}} f(x)$$

bei einer Wahl von $p(n) = 2m + 1 = 2n^k + 1$. □

Die Umkehrung des Satzes ist ebenfalls gültig:

Satz 9 (Fortnow, Rogers) *Für jede QTM M mit polynominellen Zeitverbrauch und Übergangskoeffizienten aus $\pm\sqrt{\mathbb{Q}_0^+}$, gibt es eine Funktion $f \in$ **GapP**, ein $d \in \mathbb{N}$ und eine Polynom p, so dass M jede Eingabe x mit Wahrscheinlichkeit $f(x)/d^{p(|x|)}$ akzeptiert.*

Beweis. Verweis in [5], Seite 8. □

Definition 4 *Eine Sprache L ist in $\mathbf{C_=P}$ gdw. es gibt eine Funktion $f \in$ **GapP**, so dass für alle Wörter x gilt: $x \in L \Leftrightarrow f(x) = 0$. Das Komplement wird mit $\mathbf{coC_=P}$ bezeichnet.*

Jede Sprache L in der Klasse $\mathbf{C_=P}$ lässt sich also durch eine NTM charakterisieren, bei der für jede Eingabe x gilt: $x \in L$ gdw. es gibt gleich viele akzeptierende wie ablehnende Berechnungspfade.

Korrolar 1 *Eine Sprache L ist in $\mathbf{C_=P}$ (bzw. $\mathbf{coC_=P}$) gdw. eine QTM Q mit polynominellen Zeitverbrauch gibt, so dass für alle x gilt:*

$$x \in L \Leftrightarrow Pr[Q \text{ akzeptiert } x] = 0 \text{ (bzw. } Pr[Q \text{ akzeptiert } x] \neq 0).$$

Damit folgt $\mathbf{NQP} = \mathbf{coC_=P}$.

Definition 5 *Eine Sprache $A \subseteq \Sigma^*$ reduziert in zufälliger polynomineller Zeit auf eine Sprache $B \subseteq \Sigma^*$, in Zeichen $A \leq_r B$, falls es eine in polynomieller Zeit berechenbare Funktion $f : \Sigma^* \rightarrow \Sigma^*$ und ein Polynom p gibt mit:*

$$\forall x \in A.\ Pr[f(\langle x, y\rangle) \in B \mid y \in \{0,1\}^{p(|x|)}] \geq 2/3$$

$$\forall x \notin A.\ Pr[f(\langle x, y\rangle) \in B \mid y \in \{0,1\}^{p(|x|)}] \leq 1/3.$$

Wir benutzen die gewöhnliche Erweiterung auf Klassen.

Es kann $\mathbf{PH} \leq_r \mathbf{C_=P}$ gezeigt werden ([5]).

Korrolar 2 *QAP ist hart für* $\mathbf{PH}$ *unter zufälliger Reduktion.*

Falls also die Klasse QAP irgendwo in $\mathbf{PH}$ zu finden ist, muss die polynomielle Hierarchie ab dieser Stelle kollabieren.

Literatur

[1] A. Berthiaume and G. Brassard. Oracle quantum computing. In *Proc. of the Workshop on Physics of Computation*, pages 195–199, Los Alamitos, CA, 1992. Institute of Electrical and Electronic Engineers Computer Society Press. `http://www.tcs.informatik.uni-muenchen.de/lehre/SS03/Quanten/papers/bb92.pdf.gz`.

[2] P. Clote. Komplexitätstheorie, WS97/98. `http://www.tcs.informatik.uni-muenchen.de/lehre/SS97/Komplex/index.html`.

[3] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. In *Proc. of the Royal Society* **A**, London, 1992.

[4] S. Fenner, F. Green, S. Homer, and R. Pruim. Determining acceptance possibility for a quantum computation is hard for PH. Technical report 1998-008, CS Department, Boston University, April 1998.

[5] S. Fenner, F. Green, S. Homer, and R. Pruim. Quantum NP is hard for PH. In *In Proc. 6th Italian Conf. on Theoretical Computer Science*, pages 241–252. World-Scientific, 1998.

[6] M. Hofmann and S. Jost. Komplexitätstheorie, WS02/03. `http://www.tcs.informatik.uni-muenchen.de/lehre/WS02-03/-TCS/`.

[7] M. Schmeißer. Die universelle Quanten-Turing-Maschine. `http://www.tcs.informatik.uni-muenchen.de/lehre/SS03/-Quanten/vortraege/Schmeisser.ps`.

[8] D. R. Simon. On the Power of Quantum Computation. *SIAM Journal on Computing, 26(5)*, pages 1474–1483, October 1997. `http://www.tcs.informatik.uni-muenchen.de/lehre/SS03/-Quanten/papers/sim97.pdf.gz`.