MIP und nichtdeterministische Exponentialzeit

Valentin Ziegler

19. Juli 2002

1 Einleitung

Wir haben bereits gesehen, dass $IP = PSPACE$. Nun stellt sich die Frage, wie mächtig ein interaktives Beweissystem ist, in dem der polynomiell zeitbeschränkte Verifier mit mehreren, unabhängigen Provern kommunizieren darf. Wir werden sehen, dass die Klasse MIP der von solchen Beweissystemen entschiedenen Sprachen genau $NEXP$ ist.

2 Zum Begriff des Orakels

Nach klassicher Definition entscheidet ein *Orakel* nur eine Sprache, hier werden wir jedoch von einem verallgemeinertem Orakelbegriff ausgehen. Ein Orakel sei eine beliebige, aber nichtadaptive Funktion der Orakelbandbeschrifftung. Der Funktionswert wird vom Orakel in einem Schritt auf das Orakelband zurückgeschrieben.

Bei polynomiell platzbeschränkter Orakelfunktion ist ein Verifier mit einem einem solchen Orakel jedoch nicht mächtiger als mit einem klassischem Orakel, der neue Formalismus dient lediglich der Vereinfachung der Beweise.

3 Propablistisches Orakel-Protokoll

Definition 1 (Probablistisches Orakel-Protokoll). *Eine Sprache L besitzt ein probablistisches Orakel-Protokoll, wenn eine polynomiel zeitbeschränkte probablistische Turingmaschine M existiert, so dass*

- Completeness: *Wenn $x \in L$, dann existert ein Orakel O so dass M^O mit mehr als $\frac{3}{4}$ Wahrscheinlichkeit akzeptiert.*
- Soundness: *Wenn $x \notin L$, dann akzeptiert M^O für jedes Orakel O mit weniger als $\frac{1}{4}$ Wahrscheinlichkeit.*

Es wurde bereits gezeigt, dass alle Sprachen in MIP ein probablistisches Orakel-Protokoll besitzen und umgekehrt.

4 $MIP \subseteq NEXP$

Sei $L \in MIP$. Somit existiert eine polynomiell zeitbeschränkte probablistische Orakelmaschine M mit den Eigenschaften gemäss Definition 1. OBdA können wir annehmen, dass M in jedem Rechenschritt nur ein Zufallsbit ziehen kann. Sei nun k so gewählt, dass n^k eine obere Schranke der Laufzeit von M auf einer Eingabe der Länge n ist.

Wir konstruieren anhand von M nun folgende nichtdeterministische Turingmaschine M': Bei Eingabe des Wortes $x, |x| = n$ wird eine Tabelle von Orakelantworten auf alle möglichen Fragen, welche nicht länger als n^k sind, erraten. Es gibt genau $2^{n^k+1} - 1$ solche Fragen. Danach simuliert M' M mit Eingabe x sequenziell auf allen S möglichen Berechnungspfaden, die sich aufgrund aller Auswahlen von Zufallsbits ergeben, und zählt wie oft M akzeptiert. Wenn die Anzahl akzeptierender Berechnungen von M grösser als $\frac{2}{3}S$ ist, so akzeptiert M', anderenfalls verwirft M.

Da M maximal n^k Zufallsbits ziehen kann, existert eine Konstante c, so dass die Laufzeit von M' durch 2^{n^c} beschränkt ist, somit $L \in NEXP$

5 $NEXP \subseteq MIP$

Sei $L \in NEXP$ und N_L eine nichtdeterministische 1-Band Turingmaschine mit dem Arbeitsalphabet $\Sigma = \{0, 1\}$ welche L in Exponentialzeit entscheidet. Wir brauchen nur zu zeigen, dass für L eine polynomiell zeitbeschränkte probablistische Orakel-TM M gemäss Definition 1 existiert.

5.1 Logische Codierung der Entscheidungsfrage

Nach der Tableau-Methode lässt sich für jedes $x \in \Sigma^*$ ein 3CNF-Ausdruck ϕ_x in maximal $2^{p(|X|)}$ Variablen konstruieren, so dass $x \in L$ gdw $\phi_x \in$ ef, wobei $p(|x|)$ ein Polynom ist, welches *nur* von N abhängt.

Man kann nun den Ansatz verfolgen, ein Interaktives Orakelprotokoll zu entwickeln, welches ϕ_x (bzw einen gleichwertigen arithmetischen Ausdruck) verifiziert, ähnlich wie wir es bereits für $PSPACE$-Sprachen und IP gesehen haben.

Bei $NEXP$-schweren Sprachen haben wir jedoch das Problem, dass ϕ_x eine Klauselmenge von exponentieller Kardinalität besitzt, deshalb kann keine polynomiell zeitbeschränkte Maschine eine vollständige Beschreibung von ϕ_x generieren und bearbeiten.

Allerdings ist es in Polynomialzeit möglich, zu gegebenen $x \in \Sigma^*$ für eine beliebige Klausel y zu testen, ob diese in ϕ_x enthalten ist.

Sei $p'(n) = 3p(n) + 3$. Dann gibt es einen Isomorphismus zwischen den binären Strings der Länge $p'(n)$ und der Menge der möglichen 3CNF-Klauseln über $p(n)$ Variablen, denn es gibt genau $2p(n)$ Literale, Eineindeutig codierbar in $p(n) + 1$ Bits, von denen jeweils 3 in einer Klausel auftauchen.

Seien nun $v_1, \ldots, v_{2^{p(|x|)}}$ die Variablen in ϕ_x. Wir einigen uns auf folgende Kodierung: Seien $n_1, n_2, n_3 \in \{0, \ldots, 2^{p(|x|)} - 1\}, b_1, b_2, b_3 \in \{0, 1\}$. Dann stehe $y = n_1 n_2 n_3 b_1 b_2 b_3$ für die Klausel

$$(l_1 \vee l_2 \vee l_3) \quad \text{wobei} \quad l_i = \begin{cases} v_{n_i} & , b_i = 1 \\ \overline{v_{n_i}} & , b_i = 0 \end{cases}$$

Sei A eine Belegung der Variablen in ϕ_x. Betrachten wir A als eine Abbildung von $\{0,1\}^{p(|x|)} \longrightarrow \{0,1\}$ so ist ersichtlich, dass

$*$ A erfüllt die Klausel $n_1 n_2 n_3 b_1 b_2 b_3$, falls $(A(n_1) = b_1) \vee (A(n_2) = b_2) \vee (A(n_3) = b_3)$

Für jedes $x \in \Sigma^*$ gilt zudem nach Voraussetzung

$**$ $x \in L$ genau dann, wenn es eine Belegung A der Variablen in ϕ_x gibt, so dass A *jede* Klausel aus ϕ_x erfüllt.

Wir definieren die Sprache

$$B = \{x\#y | x \in \Sigma^* \wedge y \in \Sigma^{p'(|x|)} \wedge \text{ Klausel } y \text{ enthalten in } \phi_x\} \tag{1}$$

Diese Sprache liegt bekannterweise in P. Kombinieren wir dies nun $(*)$ und $(**)$, so erhalten wir

$$\begin{aligned} &\exists A : \{0,1\}^{p(|x|)} \longrightarrow \{0,1\} \\ &\forall y = n_1 n_2 n_3 b_1 b_2 b_3 \in \{0,1\}^{p'(|x|)} \\ &(x\#y \in B \Rightarrow (A(n_1) = b_1) \vee (A(n_2) = b_2) \vee (A(n_3) = b_3)) \end{aligned} \tag{2}$$

Wir betrachten eine nichtdeterministische polynomiell zeitbeschränkte Maschine N_B, welche bei Eingabe x eine Klausel y rät und genau dann akzeptiert, wenn $x\#y \in B$. Wir können anhand von N_B mit der Tabelau-Methode zu einer Eingabe x eine entsprechende 3-CNF Ψ_x ermitteln. Da N_B in nichtdeterministischer Polynomialzeit stoppt, existieren Polynome q, l, so dass Ψ_x ein Ausdruck über max. $q(|x|)$ vielen Variablen und $l(|x|)$ vielen Klauseln ist.

Offensichtlich akzeptiert N_B immer, so dass es eine erfüllende Belegung für jedes Ψ_x gibt. Da wir ohne Einschränkung annehmen können, dass N_B die Klausel y immer zu Beginn der Berechnung errät und auf eine feste Stelle des Bandes schreibt, sind wir in der Lage, jene Klauseln in Ψ_x zu finden, welche die Bandinschrifft von y enthalten.

Wir nutzen dies, um Ψ_x zu einer generischen 3-CNF ζ_x in $p''(|x|) := p'(|x|) + q(|x|)$ Variablen und $m(|x|) := \mathcal{O}(p'(|x|)) + l(|x|)$ Klauseln zu erweitern, so dass $\zeta_x(v_1 \ldots v_{p''(|x|)})$ genau dann erfüllt ist, wenn $v_1 \ldots v_{q(|x|)}$ eine akzeptierende Berechnung von N_B codiert und $v_{q(|x|)+1} \ldots v_{p''(|x|)}$ das bei jener Berechnung "erratene" y ist.

Somit können wir (2) auch wie folgt schreiben:

$$\begin{aligned} &\exists A : \{0,1\}^{p(|x|)} \longrightarrow \{0,1\} \\ &\forall y = n_1 n_2 n_3 b_1 b_2 b_3 \in \{0,1\}^{p'(|x|)} \\ &\forall z \in \{0,1\}^{q(|x|)} \\ &(\zeta_x(yz) \Rightarrow (A(n_1) = b_1) \vee (A(n_2) = b_2) \vee (A(n_3) = b_3)) \end{aligned} \tag{3}$$

Nun hat ζ_x die Form $K_1 \wedge K_2 \wedge \cdots \wedge K_{m(|x|)}$. Nach einfachen Regeln der Aussagenlogik gilt:

$$\begin{aligned}
&\zeta_x(yz) \Rightarrow (A(n_1) = b_1) \vee (A(n_2) = b_2) \vee (A(n_3) = b_3) \\
&\equiv \neg K_1(yz) \vee \neg K_2(yz) \vee \cdots \vee \neg K_{m(|x|)}(yz) \vee (A(n_1) = b_1) \vee \cdots \vee (A(n_3) = b_3)) \qquad (4) \\
&\equiv_{def} Z_{[A]}(\omega) \quad , \ \omega = n_1 n_2 n_3 b_1 b_2 b_3 z \in \{0,1\}^{p''(|x|)}
\end{aligned}$$

5.2 Arithmetisierung des Ausdruckes $Z_{[A]}$

Wir werden nun einen arithmetischen Ausdruck für den Ausdruck $Z_{[A]}$ aus (4) konstruieren. Wir setzen im folgenden x fest, dementsprechend schreiben wir abkürzend
$p = p(|x|),\ m = m(|x|),\ p'' = p''(|x|)$.

Zunächst definieren wir ein Polynom $f_{[A]}(\xi_1, \ldots \xi_{p''})$ über $\{0,1\}^{p''}$. Sei $a_i = n_1 n_2 n_3 b_1 b_2 b_3$ die Kodierung der Klausel K_i in $Z_{[A]}$. Wir setzen

$$\alpha_i = (\xi_{n_1} - (1 - b_1))^2 + (\xi_{n_2} - (1 - b_2))^2 + (\xi_{n_3} - (1 - b_3))^2$$

Nun definieren wir

$$f_{[A]}(\xi_1, \ldots \xi_{p''}) = \beta_1 \beta_2 \beta_3 \prod_{i=1}^{m} \alpha_i \qquad (5)$$

wobei

$$\beta_i = (A(\xi_{(i-1)p+1}, \ldots, \xi_{ip}) - \xi_{3p+i})^2$$

Man erkennt leicht, dass $f_{[A]} = 0$ gdw. eine äquvivalente Belegung der Argumente von $Z_{[A]}$ mit Wahrheitswerten ($\xi_i = 1 \Leftrightarrow \omega_i$ wahr) den Ausdruck erfüllt.

Zur Erinnerung: $x \in L$ genau dann, wenn es ein $A : \{0,1\}^{p(|x|)} \longrightarrow \{0,1\}$ gibt, so dass

$$\forall \omega \in \{0,1\}^{p''}(Z_{[A]}(\omega))$$

Bzw, wie wir gesehen haben

$$\forall \vec{\xi} \in \{0,1\}^{p''}(f_{[A]}(\vec{\xi}) = 0) \qquad (6)$$

Nun ist $f_{[A]}$ noch kein Polynom, da A keines ist. Wir wollen A nun durch ein Polynom von handhabbar niedrigem Grad ersetzen, und dabei moglichst auf einem Restklassenring arbeiten, damit die Funktionswerte beschränkt bleiben. Interessanterweise gibt es ein solches Polynom:

betrachte folgendes Polynom B:

$$B(\xi_1, \ldots, \xi_n) = \sum_{(c_1,\ldots,c_p) \in \{0,1\}^p} A(c_1, \ldots, c_p) \prod_{i=1}^{p} l(c_i, \xi_i)$$

wobei

$$l(c, \xi) = \begin{cases} \xi & , c = 1 \\ 1 - \xi & , c = 0 \end{cases}$$

Es ist offensichtlich, dass B bei Einschränkung der Funktionsargumente auf $0, 1$ die gleichen Werte (modulo einer hinreichend gross gewählten Primzahl Q) liefert, wie A. Man kann auch zeigen, dass die Existenz eines solchen B nicht nur notwendige, sondern auch hinreichende Bedingung für die Entscheidungsfrage ist:

Lemma 1 (Arithmetische Charakterisierung von L). *Sei Q eine beliebige Primzahl grösser als 3^m. Dann ist $x \in L$ genau dann wenn ein Polynom $B : \mathbb{Z}_Q^p \longrightarrow \mathbb{Z}_Q$ existert, so dass*

- $(a_1, \ldots a_p) \in \{0,1\}^p \Rightarrow [B(a_1, \ldots a_p) \equiv 0 \mod Q] \vee [B(a_1, \ldots a_p) \equiv 1 \mod Q]$
- $\forall (a_1, \ldots a_p) \in \{0,1\}^{p''} (f_{[B]} \equiv 0 \mod Q)$
- *Grad von B ist nicht grösser als p*

Zum Beweis der Rückrichtung siehe Literatur.

5.3 Interaktives Orakel Protokol

Auf Basis von Lemma 1 werden wir nun ein Interaktives Orakel Protokol für L entwickeln.

Wir werden die probablistische Maschine M, welche dieses Protokoll ausführt, im folgenden auch als Verifier bezeichnen.

Wir erwarten von unserem Orakel, zur Eingabe x ein Polynom $\hat{B}$ zu berechnen und eine Primzahl $Q \in [2^{2m}, 2^{4m}]$ bereitzustellen, von denen wir fordern, dass die Bedingungen aus Lemma 1 erfüllt sind.

Jedoch müssen wir damit rechnen, dass $x \notin L$, und das Orakel versucht uns zu belügen. In dem Intervall $[2^{2m}, 2^{4m}]$ gibt es eine Primzahl, für die sich ein Zertifikat in polynomieller Zeit berechnen lässt, und da sowohl das freundliche als auch alle bösen Orakel versuchen werden, den Verifier zu überzeugen, können wie OBdA annahmen, dass der Verifier eine solche Primzahl erhält.

Anders sieht es für $\hat{B}$ aus. Da die Koeffizienten von $\hat{B}$ beliebig gross seien können, können wir die Funktion nicht einfach vom Orakel erfragen, sondern müssen damit auskommen, uns zu Argumententupel die Funktionswerte (modulo Q) berechnen zu lassen.

Wir haben nicht die Möglichkeit, in Polynomialzeit für alle $\vec{a} \in \{0,1\}^p$ zu testen, ob $\hat{B}(\vec{a})$ kongruent zu 0 oder 1 modulo Q ist, zumal wir testen müssen, dass es sich bei $\hat{B}$ tatsächlich um ein Polynom vom Grad p handelt.

Wir fordern nun vom Orakel, dass es uns auch die folgenden Polynome berrechnet:

$$\begin{aligned} B_0(\xi_1, \ldots, \xi_p) &= B(\xi_1, \ldots, \xi_p)(1 - B(\xi_1, \ldots, \xi_p)) \\ C_0(\xi_1, \ldots, \xi_{p''}) &= f_{[B]}(\xi_1, \ldots, \xi_{p''}) \end{aligned} \tag{7}$$

Somit $x \in L$ gdw $B_0 \equiv 0 \mod Q$ für alle $(a_1, \ldots a_p) \in \{0,1\}^p$ und $C_0 \equiv 0 \mod Q$ für alle $(a_1, \ldots a_p) \in \{0,1\}^{p''}$.

Weiterhin soll das Orakel noch folgende Familie von Polynomen auf dem Restklasssenring von Q berechnen:

$$\begin{aligned} B_{1\geq i\geq p}(\xi_1,\ldots,\xi_p) = B_{i-1}(\xi_1,\ldots,\xi_{i-1},0,\xi_{i+1},\ldots,\xi_p) \\ + B_{i-1}(\xi_1,\ldots,\xi_{i-1},1,\xi_{i+1},\ldots,\xi_p)\xi_i \end{aligned} \tag{8}$$

$$\begin{aligned} C_{1\geq i\geq p''}(\xi_1,\ldots,\xi_{p''}) = C_{i-1}(\xi_1,\ldots,\xi_{i-1},0,\xi_{i+1},\ldots,\xi_{p''}) \\ + C_{i-1}(\xi_1,\ldots,\xi_{i-1},1,\xi_{i+1},\ldots,\xi_{p''})\xi_i \end{aligned} \tag{9}$$

Diese Polynome haben folgende Eigenschafften:

1. Sei d der Grad von B. Dann besitzen die B_i maximal den Grad $2d+i$, die C_i einen maximal Grad von $6d+2m+i$

2. Für alle $0 \leq i \leq p$

$$B_i(\xi_1,\ldots,\xi_p) = \sum_{c_1,\ldots,c_i\in\{0,1\}} \prod_{j\leq i} B_0(c_1,\ldots,c_i,\xi_{i+1},\ldots,\xi_p)\xi_j^{c_j} \tag{10}$$

   Für alle $0 \leq i \leq p''$

$$C_i(\xi_1,\ldots,\xi_{p''}) = \sum_{c_1,\ldots,c_i\in\{0,1\}} \prod_{j\leq i} C_0(c_1,\ldots,c_i,\xi_{i+1},\ldots,\xi_{p''})\xi_j^{c_j} \tag{11}$$

3. Wenn für alle $(\xi_1,\ldots,\xi_p) \in \{0,1\}^p$ gilt $B(\xi_1,\ldots,\xi_p) \equiv 0 \mod Q$ so ist für alle $(\xi_1,\ldots,\xi_p) \in \mathbb{Z}_Q^p$ $B_p(\xi_1,\ldots,\xi_p) \equiv 0 \mod Q$
   Wenn für alle $(\xi_1,\ldots,\xi_{p''}) \in \{0,1\}^{p''}$ gilt $f_{[B]}(\xi_1,\ldots,\xi_{p''}) \equiv 0 \mod Q$ so ist für alle $(\xi_1,\ldots,\xi_{p''}) \in \mathbb{Z}_Q^{p''}$ $C_{p''}(\xi_1,\ldots,\xi_{p''}) \equiv 0 \mod Q$

Beweis. Die Eigenschafft (1) ist durch einfaches Abzählen erkennbar. Eigenschafft (2) beweisen wir induktiv. Für B_0 ist der Fall klar. Nach Induktionsvorraussetzung gelte nun:

$$B_{i-1}(\xi_1,\ldots,\xi_p) = \sum_{c_1,\ldots,c_{i-1}\in\{0,1\}} B_0(c_1,\ldots,c_{i-1},\xi_i,\ldots,\xi_p) \prod_{j\leq i-1} \xi_j^{c_j}$$

Wir setzen dies in die Definition von B_i ein und erhalten:

$$\begin{aligned} B_i(\xi_1,\ldots,\xi_p) &= \sum_{c_1,\ldots,c_{i-1}\in\{0,1\}} B_0(c_1,\ldots,c_{i-1},0,\xi_{i+1},\ldots,\xi_p) \prod_{j\leq i-1} \xi_j^{c_j} \\ &+ \sum_{c_1,\ldots,c_{i-1}\in\{0,1\}} \prod_{j\leq i-1} [B_0(c_1,\ldots,c_{i-1},1,\xi_{i+1},\ldots,\xi_p)\xi_j^{c_j}]\xi_i \\ &= \sum_{c_1,\ldots,c_i\in\{0,1\}} B_0(c_1,\ldots,c_i,\xi_{i+1},\ldots,\xi_p) \prod_{j\leq i} \xi_j^{c_j} \end{aligned}$$

Der Beweis für die C_i läuft analog.

Um Eigenschafft (3) einzusehen, beachte, dass nach (2)

$$B_p(\xi_1, \ldots, \xi_p) = \sum_{c_1,\ldots,c_p \in \{0,1\}} \prod_{j \leq p} B_0(c_1, \ldots, c_p) \xi_j^{c_j}$$

Wenn nun B_0 bei allen $0, 1$ Belegungen verschwindet, so verschwindet jeder Summand in obiger Formel, deshalb verschwindet B_p unabhängig von der Wahl der Funktionsargumente.

Der Beweis für $C_{p''}$ läuft wiederum analog. □

Somit $x \in L$ gdw $B_p \equiv 0 \mod Q$ und $C_{p''} \equiv 0 \mod Q$.

Dies ist nun eine Bedingung, welche wir, wie wir sehen können, mit sehr hoher Sicherheit testen können, allerdings müssen wir vorher sicherstellen, dass das Orakel auch wirklich die Funktionen berechnet, welche wir uns wünschen

Seien nun $\hat{B}_0, \ldots, \hat{B}_p, \hat{C}_0, \ldots, \hat{C}_{p''}$ die Funktionen, von denen das Orakel behauptet, dass sie $B_0, \ldots, B_p, C_0, \ldots, C_{p''}$ entsprechen.

Wir konstruieren nun ein Protokoll welches auf folgende Bedingungen testet:

1. $\hat{Q}$ ist Primzahl und grösser als $[2^{2m}, 2^{4m}]$
2. $\hat{B}$ ist ein Polynom von Grad nicht grösser als p
3. Alle $\hat{B}_i$ sind Polynome vom Grad nicht grösser als $2p + i$
4. Alle $\hat{C}_i$ sind Polynome von Grad nicht grösser als $6p + 2m + i$
5. $\hat{B}_0(\xi_1, \ldots, \xi_p) = \hat{B}(\xi_1, \ldots, \xi_p)(1 - \hat{B}(\xi_1, \ldots, \xi_p))$
6. $\hat{C}_0(\xi_1, \ldots, \xi_{p''}) \equiv f_{[\hat{B}]}(\xi_1, \ldots, \xi_{p''}) \mod \hat{Q}$
7. Alle $\hat{B}_i$ verhalten sich zu $\hat{B}_{i-1}$ gemäss Induktionsvorschrifft.
8. Alle $\hat{C}_i$ verhalten sich zu $\hat{C}_{i-1}$ gemäss Induktionsvorschrifft.

Wie schon zuvor geschrieben, können wir annehmen, ein Zertifikat für Q in polynomieller Zeit zu erhalten.

Sei $P : \mathbb{Z}_Q^s \to \mathbb{Z}_Q$ eine Funktion, von der wir wissen wollen, ob es sich um ein Polynom von Grad nicht grösser d handelt. Wir betrachten folgenden Test:

low degree test $6(d+2)$-maliges Ausführen von

- Zufällige Wahl von $\vec{y}, \vec{z} \in \mathbb{Z}_Q^s$
- Falls $\sum_{i=o}^{d+1} (-1)^{i+1} \binom{d}{i} P(\vec{y} + i\vec{z}) \not\equiv 0 \mod Q$, verwerfe sofort

Sei $\delta(d) = \frac{1}{2(d+2)^2}$. Wenn P ist keine $\delta(d)-$Approximation eines Polynomes p vom Grad kleinergleich d ist, so verwirft der *low degree Test* die Eingabe mit Wahrscheinlichkeit grösser $\frac{7}{8}$, falls P ein Polynom von Grad nicht grösser d ist, so verwirft der Test niemals.

Der Beweis ist sehr aufwendig, deshalb muss hier auf die Literatur verwiesen werden.

Weiterhin gilt, dass das p eindeutig bestimmt ist:

Falls P $\delta(d)-$Approximation zweier Polynome $p, q : \mathbb{Z}_Q^s \to \mathbb{Z}_Q$ vom Grad $\leq d$ ist, so ist $p = q$.

Beweis. Sei $r = p - q$. Da p und q d-Approximationen von P sind, ist p eine $2\delta(d)$-Approximation von q, somit bestitzt r mindestens $(1-2\delta(d))Q^s$ Nullstellen im Restklassenring. r ist ein Polynom von maximalen Grad d. Ein solches besitzt maximal $\frac{d}{Q}Q^s$ Nullstellen im Restklassenring, es sei denn alle Koeffizienten von r sind 0. Da $1-2\delta(d) > \frac{d}{Q}$, ist somit $r = 0$ und es gilt $p = q$ □

Wenn nun eine der Bedingungen 1-4 nicht zumindest $\delta-$approximativ erfüllt ist, so verwirft der Verifier mit Wahrscheinlichkeit grösser als $\frac{7}{8}$.

Wir für alle folgenden Betrachtungen nehmen wir nun an, dass

- $\hat{B}$ ist eine $\delta(p)-$Approximation eines Polynomes b vom Grad $\leq p$
  Jedes $\hat{B}_i$ ist $\delta(2p+i)-$Approximation eines Polynomes b_i vom Grad $\leq 2p+i$
  Jedes $\hat{C}_i$ ist $\delta(6p+2m+i)-$Approximation eines Polynomes c_i vom Grad $\leq 6p+2m+i$

Um Bedingung 5 zu prüfen, testet der Verifier, ob für ein zufällig gewähltes $\vec{y} \in \mathbb{Z}_Q^p$ gilt, das

$$\hat{B}(\vec{y})(1-\hat{B}(\vec{y})) \equiv \hat{B_0}(\vec{y}) \mod Q$$

anderenfalls verwirft er die Eingabe sofort.

Unter Annahme $(*)$: Falls $b_0(\xi_1,\ldots,\xi_p) \not\equiv b(\xi_1,\ldots,\xi_p)(1-b(\xi_1,\ldots,\xi_p)) \mod Q$, so verwirft der Verifier mit Wahrscheinlichkeit grösser als $\frac{7}{8}$

Beweis. Angenommen, $b_0 \not\equiv b(1-b) \mod Q$. Betrachte $\beta = b_0-(b(1-b))$. Da es sich hierbei um nichttriviales Polynom vom Grad $\leq 2p$ handelt, besitzt β maximal $\frac{2p}{Q}Q^p$ Nullstellen im Restklassenring. Die Wahrscheinlichkeit, das bei Wahl eines zufälligen $\vec{y} \in \mathbb{Z}_Q^p$ gilt, dass $\beta(\vec{y}) \not\equiv 0 \mod Q$ ist somit mindestens $1-\frac{2p}{Q}$.

Nun ist $\hat{B}$ eine $\delta(p)$-Approximation von b und $\hat{B}_0$ eine $\delta(2p)$-Approximation von b_0, somit beträgt die Wahrscheinlichkeit des Verwerfens $1-\frac{2p}{Q}-\delta(p)-\delta(2p)$ Dies ist für hinreichend lange x grösser als $\frac{7}{8}$. □

Um Bedingung 6 zu prüfen, testet der Verifier, ob für ein zufällig gewähltes $\vec{y} \in \mathbb{Z}_Q^{p''}$ gilt, das

$$\hat{C_0}(\vec{y}) \equiv \beta_1\beta_2\beta_3\prod_{i=1}^{m}\alpha_i \mod \hat{Q}$$

wobei $\beta_i = (\hat{B}(y_{(i-1)p+1},\ldots,y_{ip}) - y_{3p+i})^2$

Unter Annahme $(*)$: Falls $c_0(\xi_1,\ldots,\xi_{p''}) \not\equiv f_{[b]}(\xi_1,\ldots,\xi_{p''}) \mod \hat{Q}$ so verwirft der Verifier mit Wahrscheinlichkeit grösser als $\frac{7}{8}$.

Beweis. Angenommen, $c_0 \not\equiv f_{[b]} \mod Q$. Betrachte $\zeta = c_0 - f_{[b]}$. Dies ist ein nichttriviales Polynom vom Grad $\leq 6p+2m$ welches maximal $\frac{6p+2m}{Q}Q^{p''}$ Nullstellen im Restklassenring besitzt. Die Wahrscheinlichkeit, das bei Wahl eines zufälligen $\vec{y} \in \mathbb{Z}_Q^{p''}$ gilt, dass $\zeta(\vec{y}) \not\equiv 0 \mod Q$ ist somit mindestens $1-\frac{6p+2m}{Q}$, bzw. $1-\frac{6p+2m}{Q}-\delta(6p+2m)-3\delta(p)$ bei Betrachtung von $\hat{C_0}$ und $\hat{B}$. Für hinreichend lange x ist auch dies grösser als $\frac{7}{8}$. □

Nun testen wir die Bedingungen 7 und 8. Da die B_i und C_i nach gleicher Induktionsvorschrifft aus ihrer Vorgängerfunktion gebildet werden, können wir bei beiden Tests den *selbstkorrigierenden Polynomtest* verwenden.

Sei $F_i : \mathbb{Z}_Q^s \longrightarrow \mathbb{Z}_Q$ ein Polynom vom Grad d und $F_{i+1} : \mathbb{Z}_Q^s \longrightarrow \mathbb{Z}_Q$ eines vom Grad $d+1$ welches aus F_i entsteht gemäss:

$$F_{i+1}(x_1, \ldots, x_s) = F_i(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots x_s) + F_i(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots x_s)x_i.$$

selbstkorrigierender Polynomtest (Skizze):

- wähle $\vec{z} \in \mathbb{Z}_Q^s$.
- Erfrage für $d+1$ zufälligen Zahlen $x_1, \ldots, x_{d+1}$ die Werte $y_1, \ldots, y_{d+1}$ von $F_i(z_1, \ldots, z_{i-1}, x_j, z_{i+1}, \ldots z_s)$
- Berechne anhand der Punkte (x_j, y_j) die $d+1$ eindeutig bestimmten Koeffizienten eines univarianten Polynomes $f(x)$ vom Grad d
- Berechne anhand dieser Koeffizenten die eindeutig bestimmte Funktion $f'(x) = f(0) + f(1)x$
- Erfrage für $d+2$ zufälligen Zahlen $x'_1, \ldots, x'_{d+2}$ die Werte $y_1, \ldots, y_{d+2}$ von $F_{i+1}(z_1, \ldots, z_{i-1}, x'_j, z_{i+1}, \ldots z_s)$
- Berechne anhand der Punkte (x'_j, y'_j) die $d+2$ eindeutig bestimmten Koeffizienten eines univarianten Polynomes $g(x)$ vom Grad $d+1$
- Teste ob $f'(x) \equiv g(x) \mod Q$, verwerfe anderenfalls

Falls $x \in L$, d.h das Orakel lügt nicht, entsprechen die berechneten Funktionen $f(x), g(x)$ nun genau den univarianten Polynomen $F'_i(x), F'_{i+i}(x)$, die aus F_i, F_{i+i} durch Festlegen von $s-1$ Argumenten entstehen, somit verwirft der Verifier in diesem Fall niemals.

Die Betrachtung des anderen Falls gestaltet sich komplizierter, denn im Gegensatz zu den anderen Tests können wir hier nicht einfach mit den approximierten Polynomen b_i, c_i rechnen (Für diese würde der Test ja immer gelingen oder immer scheitern), sondern müssen die Approximationsfehler direkt mit in die Rechnung einbeziehen.

Somit wird auch hier auf die Literatur verwiesen werden, es sei nur gesagt, dass

Unter Annahme $(*)$: Falls Bedingung 7 oder 8 für b_i, c_i verletzt sind, so verwirft der Verifier mit Wahrscheinlichkeit grösser $\frac{7}{8}$

Nun kann der Verifier sich der Entscheidungsfrage zuwenden, indem er testet, ob $\hat{B}_p \equiv 0 \mod \hat{Q}$ und $\hat{C}_{p''} \equiv 0 \mod \hat{Q}$ für beliebige Wertebelegungen.

Auch dieses testen wir durch eine einfache Stichprobe.

Unter Annahme (*): Falls $b_p \not\equiv 0 \mod Q$ oder $c_{p''} \not\equiv 0 \mod Q$, so verwirft der Verifier mit Wahrscheinlichkeit grösser $\frac{7}{8}$.

Beweis. Angenommen, eines der Polynome ist nichttrivial. Die Wahrscheinlichkeit, dass wir keine Nullstelle von b_p (bzw von $c_{p''}$) treffen, ist mindestens $1 - \frac{p}{Q}$ (bzw $1 - \frac{p''}{Q}$). Nun ist $\hat{B_p}$ eine $\delta(p)$-Approximation von b_p (bzw $\hat{C}_{p''}$ ist eine $\delta(p'')$-Approximation von $c_{p''}$) Somit ist die Wahrscheinlichkeit mindestens $1 - \frac{p}{Q} - \delta(p)$ (bzw $1 - \frac{p''}{Q} - \delta(p'')$), dass wir die Nichttrivialität bemerken. Dies ist für alle hinreichend langen x grösser als $\frac{7}{8}$ □

6 Betrachtung der Akzeptanzwahrscheinlichkeiten

Bevor wir die oben erarbeiteten Ergebnisse zusammenfassen, müssen wir noch das Vorgehen bei der Auswahl der Zufallszahlen betrachten. Da wir in unserer Rechnung angenommen haben, dass alle gezogen Zahlen unabhängig und gleichverteilt in $\mathbb{Z}_Q$ sind, können diese nicht durch einfaches Münzwerfen bestimmt werden. Stattdessen muss jede Zahl mit dem bereits vorgestellten *Sampling Algorithmus* ermittelt werden, bei dem wir die Anzahl der Iterationen so von der Eingabelänge abhängig wählen, dass dieser nur noch mit einer Wahrscheinlichkeit von weniger 2^{-p} zum sofortigen Akzeptieren der Eingabe führt (und in polynomieller Zeit arbeitet).

Die totale Anzahl benötigter Zufallszahlen liegt in $\mathcal{O}((p + p'' + m)^4)$ somit beträgt die Wahrscheinlichkeit eines Samplingfehlers $\mathcal{O}(\frac{(p+p''+m)^4}{2^p}) \leq \frac{1}{8}$ für hinreichend lange x.

Falls $x \in L$ akzeptiert der Verifier mit Wahrscheinlichkeit 1, da alle Tests erfolgreich sind und bei eventuell auftretenden Samplingfehlern ohnehin akzeptiert wird.

Falls $x \notin L$, so ist das Orakel gezwungen entweder:

- mindestens eine Funktion $\hat{B}_0, \ldots \hat{B}_p, \hat{C}_0, \ldots, \hat{C}_{p''}$ liefern, die keine (eindeutige) Approximation eines Polynomes von entsprechendem Grad ist, was der Verifier mit der Wahrscheinlichkeit von weniger als $\frac{1}{8}$ nicht bemerkt.

oder die approximierten Polynome so wählen, dass:

- mindestens eine der Bedingungen 5 bis 8 nicht erfüllt ist, was der Verifier mit der Wahrscheinlichkeit von weniger als $\frac{1}{8}$ nicht bemerkt

oder:

- Die korrekt gebildeten, jedoch nicht nullwertigen Polynome $\hat{B}_p$ und $\hat{C}_{p''}$ zu berechnen, was der Verifier mit Wahrscheinlichkeit von weniger als $\frac{1}{8}$ nicht bemerkt.

Zussammen mit der Wahrscheinlichkeit des Samplefehlers akzeptiert der Verifier x mit Wahrscheinlichkeit von weniger als $\frac{1}{4}$.

Dies beweist die Existenz von probablistischen Orakel-Protokollen für alle Sprachen in $NEXP$.

7 Literatur

L. Hemaspaandra, M. Ogihara The Complexity Theory Companion, *Springer*, 2002

L. Fortnow, J. Rompel, M.Sipser On the Power of Multi-Prover Interactive Protocols, *Cambridge*, 1990

L. Babai, L. Fortnow, C.Lund Non-Deterministic Exponential Time has Two-Prover Interactive Protocols, *Cambridge*, 1991