# Multi-Prover-Systeme

Ben Baessler

04.Juni 2002
11.Juni 2002

# 1 Einführung

Ein MIP-System besteht aus einem Verifier und mehreren Provern die untereinander nicht kommunizieren können.
Somit kann man ein MIP-System als ein Verhör zweier Verdächtiger ansehen.
Diese versuchen natürlich die Polizei davon zu überzeugen, daß sie unschuldig sind.
Sind sie wirklich unschuldig, so ist die Wahrscheinlichkeit das sie die Polizei von ihrer Unschuld überzeugen, indem sie dieselbe Geschichte erzählen, während sie in zwei verschiedenen Räumen ohne miteinader reden zu können vernommen werden.
Wie man später sehen wird liegt in dieser Art der Bestätigung der Schlüssel zur zusätzlichen Macht von MIP-Systemen.

# 2 Definitionen

## 2.1 Vordefinitionen

1. $P_1, P_2, \ldots, P_k$ seien unendlich mächtige Maschinen
   (wobei k polynomiell in der Größe des Inputs ist)

2. V sei eine probabilistische Polynomial-Zeit Maschine

3. Alle Prover und der Verifier haben ein gemeinsames read-only Eingabeband

4. Der Verifier teilt mit jedem Prover ein Kommunikationsband, auf das kein weiterer Prover Zugriff hat.

## 2.2 Formale Sicht

Formal gesehen ist jedes $P_i$ eine Funktion des Inputs und der abgelaufenen Kommunikation. Diese Funktion ist lediglich dadurch beschränkt, daß die Länge der Nachricht, die von dieser Funktion erzeugt wird, durch ein Polynom in der Größe des Inputs beschränkt sein muss.

## 2.3 Multi-Prover-Interaktives-Protokoll

$P_1, P_2, \ldots, P_k$ und $V$ bilden ein Multi-Prover-Interaktives-Protokoll für eine Sprache L wenn gilt:

1. Wenn $x \in L$,
   dann ist $Pr(P_1, P_2, \ldots, P_k$ und V akzeptiert auf x)$> 1 - 2^{-n}$

2. Wenn $x \notin L$,dann gilt für alle Prover $P'_1, P'_2, \dots, P'_k$:
   $Pr(P'_1, P'_2, \dots, P'_k$ und V akzeptiert auf x)$< 2^{-n}$

Eine Runde eines MIP besteht aus Nachrichten des Verifiers zu einigen (oder allen) Provern, gefolgt von Nachrichten dieser Prover an den Verifier.

# 3 Probabilistische Orakelmaschinen

In diesem Abschnitt soll die Äquivalenz von MIP und einem probabilistischen Orakel-Protokoll gezeigt werden.

## 3.1 Probabilistisches Orakel-Protokoll

Eine Sprache L hat ein probabilistisches Orakel-Protokoll, wenn eine probabilistische Polynomial-Zeit Orakel-Turingmaschine M gibt, so daß für jedes $x \in \Sigma^*$ folgendes gilt:

1. Für alle $x \in L$ gibt es ein Orakel H, so daß $M^H$ auf Eingabe x mit Wahrscheinlichkeit $> \frac{3}{4}$ akzeptiert. (Completeness)

2. Für alle $x \notin L$ und alle Orakel H gilt:
   $M^H$ akzeptiert mit Wahrscheinlichkeit $< \frac{1}{4}$ (Soundness)

Bemerkung:

Hier wird als Schranke $\frac{3}{4}$ angenommen.
Es ist jedoch klar, das man diese durch Mehrfachausführung auch unter $1 - 2^{-n}$ drücken kann.

## 3.2 Theorem 1

Für jede Sprache L gilt:
L ist in MIP $\Leftrightarrow$ L hat ein probabilistisches Orakel-Protokoll

Beweis:
($\Rightarrow$)

Angenommen L ist eine Sprache in MIP. Sei V ein Verifier der dies bezeugt.

Sei $k \geq 1$ die Anzahl Prover mit denen V kommuniziert.
O.B.d.A. kann annehmen die Prover seien deterministisch.
Modifiziere die Fragen von V. Sei x eine Eingabe ins System. Sei y die Frage die V stellen will. Seien i, $1 \leq i \leq k$, und $j \geq 1$ so daß V die Frage y an $P_i$ stellt und V bei Eingabe x bis dahin j-1 Fragen an $P_i$ gestellt hat.
Ersetze nun die Frage durch $< x, i, j, y, W >$.
W ist hier bei folgendes $W = y_1 \sharp b_1 \$ y_2 \sharp b_2 \$ \ldots \$ y_{j-1} \sharp b_{j-1}$, wobei für jedes l,$1 \leq l \leq j-1$, $y_l$ die vierte Komponente der l-ten Frage von V an $P_i$ ist und $b_l$ die Antwort die $P_i$ auf diese Frage gibt.
Die Modifikation macht die Fragen eindeutig und dem Prover wird nichts verraten was er nicht schon weiß.
Da die Prover deterministisch und die Fragen eindeutig sind, kann man nun V in eine Orakelmaschine N umwandeln indem man die Prover durch ein Orakel ersetzt.
Damit ist für alle $x \in \Sigma^*$ die größte Wahrscheinlichkeit, daß N auf Eingabe x mit einem Orakel akzeptiert gleich der größsten Wahrscheinlichkeit das V auf Eingabe x mit einer Menge von k Provern akzeptiert.
Somit hat L ein probabilistisches Orakel-Protokoll.

($\Leftarrow$)

Angenommen es gibt eine polynomiell zeitbeschränkte probabilistische Orakel-Turingmaschine N die bezeugt das L ein probabilistisches Orakel-Protokoll hat.
Wir nehmen an es gibt ein Polynom p, so daß, für jedes $x \in \Sigma^*$, N auf Eingabe x immer genau p($\mid x \mid$)Fragen stellt.
Sei V ein Verifier mit zwei Provern $P_1, P_2$ die auf Eingabe x folgenden Algorithmus ausführen:

1. Schritt: V führt das folgende 10p($\mid x \mid$) mal aus:
   (a) V simuliert N auf Eingabe x und verwendet $P_1$ als Orakel
   (b) V ruft den SamplingAlgorithmus* mit M=p($\mid x \mid$) und t=p($\mid x \mid$) auf und benutzt ihn um eine Frage $y_i$ auszuwählen und stellt diese an $P_2$
2. V akzeptiert x wenn für alle i, $1 \leq i \leq 10(\mid x \mid)$, die Antwort von $P_2$ auf $y_i$ identisch zu der von $P_1$ ist und die Anzahl der Simulationen von N auf Eingabe x die akzeptieren mindestens 5p($\mid x \mid$) ist.
   Anderenfalls weißt V x zurück.

*siehe 7.Anhang

Offensichtlich ist V polynomiell beschränkt.
Nun ist zu zeigen, daß das Protokoll sowohl complete als auch sound ist.

**Completeness des Protokolls**

Sei $x \in L$. Dann existiert ein Orakel H relativ zu dem N auf Eingabe x mit Wahrscheinlichkeit $\rho > \frac{3}{4}$ akzeptiert.
Angenommen $P_1, P_2$ antworten als wären sie H. Dann sind alle Konsistenztests zwischen $P_1, P_2$ erfolgreich.
$\Rightarrow$ Pr(V akzeptiert)=Pr(mehr als die Hälfte der Simulationen akzeptieren)
Da die Münzwürfe von V unabhängig sind ist der Erwartungswert $\alpha$ für die Anzahl akzeptierender Berechnungen die V findet $\rho(10p) > \frac{15p}{2}$ und die Varianz $\sigma$ dieser Anzahl ist $\rho(1-\rho)(10p) < \frac{15p}{2}$, wobei p für p(x) steht.
Sei A=(Anzahl akzeptierender Pfade)
$\Rightarrow_{Chebyshev}$ $Pr(A \leq 5p) \leq \frac{\sigma}{|5p-\alpha|^2} < \frac{\frac{15p}{8}}{|5p-\frac{15p}{2}|^2} \leq \frac{3}{10p} (< \frac{1}{4}$ für $p \geq 2)$
Da wir das Polynom p beliebig groß wählen können, ist die Wahrscheinlichkeit der Akzeptanz größer als $\frac{3}{4}$ für x.
Somit ist das Protokoll complete.

**Soundness des Protokolls**

Sei $x \in \overline{L}$. Sei $n =| x |$ und $m = 10p(n)$.
Es ist zu zeigen, daß für jedes Paar von Provern $(P_1, P_2)$ die Wahrscheinlichkeit das V auf Eingabe x akzeptiert indem es $P_1$ und $P_2$ verwendet streng kleiner als $\frac{1}{4}$ ist.
Der SamplingAlgorithmus wird $10p(| x |)$ mal ausgeführt.
Pr(SamplingAlgorithmus versagt)$\leq 2^{-p(|x|)}$.
Pr(Sampling-Fehler tritt auf)=$\frac{10p(|x|)}{2^{p(|x|)}} (< \frac{1}{10}$ für $p(n) \geq 10)$

Behauptung:

Ein Paar von deterministischen Provern $(P_1, P_2)$ kann die höchste Akzeptanzwahrscheinlichkeit erreichen.

Bew:

Sei das Ziel von $(P_1, P_2)$ die Akzeptanzwahrscheinlichkeit von V auf Eingabe x durch Interaktion mit ihnen zu maximieren.
Eine weitere Aufgabe von $P_2$ ist es die Antworten die $P_1$ gegeben hat

zu erraten.
Bekommt $P_2$ nun eine Frage so verwendet er seine unbeschränkte Rechenkraft um die Wahrscheinlichkeit zu berechnen, daß $P_1$ mit 0 geantwortet hat. $P_2$ kann somit in folgender Weise die Wahrscheinlichkeit maximieren:

$$P_2 \text{ antwortet } \begin{cases} 0 & \text{falls } \Pr(P_1 \text{ hat 0 geantwortet}) \geq \frac{1}{2} \\ 1 & \text{sonst} \end{cases}$$

Somit kann $P_2$ deterministisch vorgehen.

Betrachte nun den Fall, das die letzte Frage von V auf Eingabe x an $P_1$ gestellt wird.
Da auch $P_1$ unbeschränkte Rechenkraft hat und zudem das Protokoll von $P_2$ kennt, kann $P_1$ die Wahrscheinlichkeit das V auf Eingabe x akzeptiert, wenn er eine 0 antwortet, berechnen.
Seien $\alpha_1$ und $\alpha_2$ diese Wahrscheinlichkeiten.
Um die Akzeptanzwahrscheinlichkeit von V auf Eingabe x zu maximieren, wählt $P_1$ seine Antwort folgendermassen deterministisch:

$$P_1 \text{ antwortet } \begin{cases} 0 \text{ falls } \alpha_0 \geq \alpha_1 \\ 1 \text{ sonst} \end{cases}$$

Somit kann $P_1$ die letzte Frage deterministisch beantworten ohne die Akzeptanzwahrscheinlichkeit von V zu verringern.

Wiederholt man nun die Argumente für $P_1$ und $P_2$ mehrfach so ergibt sich eine deterministische Gesamtstrategie für beide, die die Akzeptanzwahrscheinlichkeit nicht verringert.

Da $P_2$ determinitsich ist und V pro Runde nur eine Frage an $P_2$ stellt, kann man die Funktion von $P_2$ in Runde i als die eines Orakels sehen.
Angenommen $P_1$ wählt am Anfang jeder Runde eine Strategie für diese Runde.
Diese Wahl legt die Wahrscheinlichkeit fest mit der $P_1$ eine andere Antwort gibt als $P_2$ geben würde. Damit kann man die Gesamtstrategie von $P_1$ mit m reellen Zahlen aus [0,1] parametrisieren.
Für jedes $\alpha = (\alpha_1, \ldots, \alpha_m) \in [0,1]^m$ sagt man $P_1$ wählt eine $\alpha$-Strategie wenn für alle $i, 1 \leq i \leq m$, die Strategie von $P_1$ in Runde i ist mit Wahrscheinlichkeit $\alpha_i$ eine andere Antwort zu geben als $P_2$ es würde.

Nun bleibt noch zu zeigen, das ein $n_0 > 0$ existiert, so daß für alle $n \geq n_0$ und für alle $\alpha \in [0,1]^m$ die Wahrscheinlichkeit das V x akzeptiert, wenn $P_1$ eine $\alpha$-Strategie wählt, streng kleiner als $\frac{1}{4}$ ist.

Angenommen $P_1$ wählt eine $\alpha$-Strategie.
Da $x \notin L$ gilt: $\forall$ Orakel Q Pr($N^Q(x)$ akzeptiert) $< \frac{1}{4}$
$\Rightarrow \forall$ i:
Pr(Sim. von N(x) in Runde i mit $P_1$ akzept.)$< min\{1, \frac{1}{4} - \alpha_i\} \leq \frac{1}{4} + \alpha_i$
$\Rightarrow$ E($\sharp$ generierter akzept. Sim.)$< (\frac{m}{4}) + \Sigma_{1 \leq i \leq m} \alpha_i$
$\Rightarrow$ Var($\sharp$ generierter akzept. Sim.)$< (\frac{m}{4}) + \Sigma_{1 \leq i \leq m} \alpha_i$

Beh: Pr(V(x) akzeptiert)$< \frac{1}{4}$

Bew:

a) Ann: $\Sigma_{1 \leq i \leq m} \alpha_i \leq \frac{9m}{40}$
$\Rightarrow_{Chebyshev}$

$$\begin{aligned} Pr(\sharp \text{ akzept. Sim.} \geq \frac{m}{2}) &< \frac{\frac{m}{4} + \frac{9m}{40}}{(\frac{m}{2} - \frac{m}{4} - \frac{9m}{40})^2} = \frac{7600}{m} = \frac{760}{p} \\ &< \frac{1}{8} \text{ für } p(n) \geq 6081 \end{aligned} \quad (1)$$

$\Rightarrow$ Pr(V(x) akzeptiert, wenn kein Sampling-Fehler auftritt)$< \frac{1}{8}$

b) Ann: $\Sigma_{1 \leq i \leq m} \alpha_i > \frac{9m}{40}$
$\Rightarrow$ E($\sharp$ Sim. in den $P_1 = P_2$ in mind. einer Frage)$=\frac{9m}{40}$
$\Rightarrow$ Var($\sharp$ Sim. in den $P_1 = P_2$ in mind. einer Frage)$< \frac{9m}{40}$
$\Rightarrow_{Chebyshev}$

$$\begin{aligned} Pr(\sharp \text{ Sim. in den} P_1 \neq P_2 \leq \frac{m}{5}) &< \frac{\frac{9m}{40}}{(\frac{9m}{40} - \frac{m}{5})^2} = \frac{3600}{m} = \frac{360}{p} \\ &< \frac{1}{16} \text{ für } p(n) \geq 5761 \end{aligned} \quad (2)$$

Wenn ($\sharp$ Sim. in denen $P_1 \neq P_2$)=2p(n)=$\frac{m}{5}$ ist
dann

$$\begin{aligned} Pr(\text{Widerspruch wird nicht entdeckt}) &\leq (1 - \frac{1}{p(n)})^{2p(n)} \\ &< \frac{1}{16} \text{ für } p(n) \geq 5761 \end{aligned} \quad (3)$$

Somit ist für n ausreichend groß die Wahrscheinlichkeit das V(x) akzeptiert, wenn kein Sampling-Fehler auftritt, kleiner als $\frac{1}{8}$
Da ein Sampling-Fehler V akzeptieren läßt und mit Wahrscheinlichkeit

kleiner $\frac{1}{8}$ auftritt, ist die Gesamtwahrscheinlichkeit das V(x) akzeptiert kleiner als $\frac{1}{4}$.
Damit ist das Protokoll sound.

Dies beweißt das Theorem.

$\square$

# 4 MIP⊆NExp

In diesem Abschnitt soll eine obere Schranke für die Komplexität von MIP-Systemen gezeigt werden.

## 4.1 Theorem 2

Wenn es ein MIP-System gibt, das eine Sprache L akzeptiert, so kann L in nichtdeterministischer Exponential-Zeit berechnet werden.

Beweis:

Aus Theorem1 folgt:
Es gibt eine probabilistische Orakelmaschine M die L akzeptiert, wobei M bei Eingabe der Länge n dazu Zeit $n^k$ für ein $k > 0$ benötigt.

Baue folgendermassen eine NExp-Maschine:
Bei Eingabe x der Länge n, rate den Wert des Orakels H auf alle Fragen der Länge höchstens $n^k$.
Es gibt genau $2^{n^k+1} - 1$ solche Fragen.
Für einen String r der Länge $n^k$ sei

$$\text{f(x,H,r)}=\begin{cases} 1 & \text{wenn M(x) akzeptiert und dazu r Münzwürfe verwendet} \\ & \text{und die Orakelantworten von H bekommt} \\ 0 & \text{sonst} \end{cases}$$

Berechne nun

$$\text{S}=\Sigma_{r\in\{0,1\}^{n^k}} f(x,H,r).$$

Akzeptiere wenn $S > \frac{2^{n^k}}{2}$
Nach der Definition von probabilistischen Orakel-Maschinen gibt es für $x \in L$ eine Belegung des Orakels so daß $S \geq (1-2^{-n})2^{n^k}$ und für $x \notin L$ eine Belegung des Orakles so daß $S < (2^{-n})2^{n^k}$.
Dies beweist die Korrektheit der obigen Berechnung.

$\square$

Theorem2 zeigt NICHT, das jede Sprache L aus MIP mit Provern bewiesen werden kann die NExp-Komplexität haben. Denn die Prover müssen die nichtdeterministischen Schätzungen, die die NExp-Maschine akzeptieren lassen, erstmal finden.
Es ist nicht bekannt ob man zeigen kann das NExp-Komplexität für die Prover ausreicht. Babai,Fortnow und Lund haben jedoch gezeigt, das jede Sprache aus Exp nur Prover mit Exp-Komplexität benötigt.

# 5 Protokolle mit beschränkter Rundenzahl

## 5.1 Ergebnisse

1. Jede Sprache die durch ein Ein-Prover-IP-System bewiesen werden kann, hat eine Zwei-Prover-Protkoll das nur eine Runde verwendet.

2. Jede Sprache die durch ein MIP-System bewiesen werden kann, hat ein 3-Prover-Protokoll das nur zwei Runden verwendet.

Dies läßt sich beweisen, indem man spezielle Eigenschaften des MIP-Protokolls ausnutzt.
Hier soll aber die anfängliche und falsche Annahme widerlegt werden, die im ursprünglichen Beweis verwendet wurde.

## 5.2 Falsche Beweisannahme

Im Beweis nahmen Fortnow,Rompel und Simpser an das man die Fehlerwahrscheinlichkeit bestimmter Protokolle verringern kann, indem man sie parallel laufen läßt.
Sie nahmen an, daß wenn man die Kommunikation zwischen den Provern verhindert, sich die parallelen Abläufe des Protokolls unabhängig verhalten, so wie es parallele Abläufe des IP-Protokolls tun.
Diese Annahme ist falsch.

## 5.3 Gegenbeispiel

Angenommen man hat folgendes Zwei-Prover-Protokoll:

$$\begin{aligned} V &: \text{Wählt zufällig zwei Bits a und b} \\ V \rightarrow P_1 &: a \\ V \rightarrow P_2 &: b \\ P_1 \rightarrow V &: c \\ P_2 \rightarrow V &: d \\ V &: \text{Akzeptiert wenn } (a \vee c) \neq (b \vee d) \end{aligned}$$

Für jeden Prover gibt es zwei Strategien:

$$P_1^1(a) = \begin{cases} 0 & a = 0 \\ 1 & a = 1 \end{cases} \text{ oder } P_1^2(a) = \begin{cases} 1 & a = 0 \\ 0 & a = 1 \end{cases}$$
$$\text{und}$$
$$P_2^1(b) = \begin{cases} 0 & b = 0 \\ 1 & b = 1 \end{cases} \text{ oder } P_2^2(b) = \begin{cases} 1 & b = 0 \\ 0 & b = 1 \end{cases}$$

Daraus ergeben sich 4 Situationen (mögliche Prover-Startegien):

1. $P_1^1, P_2^1$ :

| $a$ | $b$ | $c$ | $d$ | $(a \vee c) \neq (b \vee d)$? |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | *nein* |
| 0 | 1 | 0 | 1 | *ja* |
| 1 | 0 | 1 | 0 | *ja* |
| 1 | 1 | 1 | 1 | *nein* |

2. $P_1^1, P_2^2$ :

| $a$ | $b$ | $c$ | $d$ | $(a \vee c) \neq (b \vee d)$? |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | *ja* |
| 0 | 1 | 0 | 0 | *ja* |
| 1 | 0 | 1 | 1 | *nein* |
| 1 | 1 | 1 | 0 | *nein* |

3. $P_1^2, P_2^1$ :

| $a$ | $b$ | $c$ | $d$ | $(a \vee c) \neq (b \vee d)$? |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | *ja* |
| 0 | 1 | 1 | 1 | *nein* |
| 1 | 0 | 0 | 0 | *ja* |
| 1 | 1 | 0 | 1 | *nein* |

4. $P_1^2, P_2^2$ :

| $a$ | $b$ | $c$ | $d$ | $(a \vee c) \neq (b \vee d)$? |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | *nein* |
| 0 | 1 | 1 | 0 | *nein* |
| 1 | 0 | 0 | 1 | *nein* |
| 1 | 1 | 0 | 0 | *nein* |

Wie man daran nun leicht sieht bringt die beste Strategie der Prover den Verifier dazu mit Wahrscheinlichkeit $\frac{1}{2}$ zu akzeptieren.
Beachte: Keiner der beiden Prover weiß welches Bit der Verifier dem anderen Prover geschickt hat.

Betrachten wir nun die Zwei-Runden-Version des Protokolls:

$$\begin{array}{rcl} V & : & \text{Wählt zufällig die Bits } a_1, a_2 \text{ und } b_1, b_2 \\ V \to P_1 & : & a_1, a_2 \\ V \to P_2 & : & b_1, b_2 \\ P_1 \to V & : & c_1, c_2 \\ P_2 \to V & : & d_1, d_2 \\ V & : & \text{Akzeptiert wenn } (a_1 \vee c_1) \neq (b_1 \vee d_1) \text{ und } (a_2 \vee c_2) \neq (b_2 \vee d_2) \end{array}$$

Wenn parallele Läufe des Protokolls sich unabhängig verhalten, so sollte die optimale Strategie für die Prover den Verifier mit einer Wahrscheinlichkeit von $(\frac{1}{2})^2 = \frac{1}{4}$ akzeptieren lassen.
Die folgende Strategie läßt den Verifier aber mit der Wahrscheinlichkeit $\frac{3}{8}$ akzeptieren:

$P_1$: Wenn $a_1 = a_2 = 0$ antworte $c_1 = c_2 = 0$ anderenfalls $c_1 = c_2 = 1$
$P_2$: Wenn $b_1 = b_2 = 0$ antworte $d_1 = d_2 = 0$ anderenfalls $d_1 = d_2 = 1$

Bew:

| $a_1$ | $b_1$ | $a_2$ | $b_2$ | $c_1$ | $d_1$ | $c_2$ | $d_2$ | $(a_1 \vee c_1) \neq (b_1 \vee d_1)$? | $(a_2 \vee c_2) \neq (b_2 \vee d_2)$? |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *nein* | *nein* |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | *ja* | *ja* |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | *ja* | *ja* |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | *nein* | *nein* |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | *ja* | *ja* |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | *ja* | *ja* |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | *nein* | *nein* |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | *nein* | *nein* |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | *ja* | *ja* |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | *nein* | *nein* |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | *ja* | *ja* |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | *nein* | *nein* |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | *nein* | *nein* |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | *nein* | *nein* |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | *nein* | *nein* |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | *nein* | *nein* |

Hieran sieht nun leicht die Korrektheit der obigen Aussage.

# 6 Literatur


1. On the Power of Multi-Prover Interactive Protocols (Paper im Netz)
   L.Fortnow/J.Rompel/M.Sipser

2. The Complexity Theory Companion (Buch)
   L.A.Hemaspaandra/M.Ogihara


# 7 Anhang

## SamplingAlgorithmus

Eingabe: Zwei Integer $M \geq 2$ und $t \geq 1$
Ausgabe: Eine Zahl in $\{0, \ldots, M-1\}$
Fehlerwahrscheinlichkeit: $2^{-t}$

Sei l die kleinste Zahl, so daß $2^l \geq M$.
Wiederhole das folgende höchstens t mal:

> Benutze l Münzwürfe um eine Zahl Y zwischen 0 und $2^l - 1$ auszuwählen.
> Wenn $Y \leq M$, dann stoppe die Schleife und gebe Y aus.

Wenn keiner der Versuche erfolgreich ist, dann akzeptiere x.