Seminar Interaktive Beweissysteme
$IP = PSPACE$

Daniel Rolf

26. April 2002

1 Die Klasse IP

Ein klassisches Beweissystem besteht aus einem allwissenden und in seinen Ressourcen unbeschränkte Beweiser P und einem polynomiell beschränkten Überprüfer V. Der Beweis wird geführt, indem P einen polynomiell langen Beweis vorschlägt, welcher von V in polynomieller Zeit verifiziert wird. Anhand des klassischen Beweissystems kann man die Klasse NP definieren, als Menge aller Sprachen, für die ein solches Beweissystem existiert.

Die Klasse IP wird nun als Menge aller Sprachen definiert, für die ein Beweiser und ein probabilistischer Überprüfer existiert, welche sich gegenseitig abwechselnd Nachrichten schicken. Der Beweiser antwortet auf einen Nachrichtendialog mit einer neuen Nachricht, derer Länge jedoch polynomiell beschränkt ist. Der Überprüfer muss nach einer polynomiellen Anzahl von Nachrichten eine Entscheidung fällen. Der Überprüfer kann auf einen Nachrichtendialog mit einer vom Zufall abhängigen Nachricht antworten.

Definition 1. *Ein **interaktives Beweissystem** für eine Sprache L, besteht aus einem probalistischen Überprüfer V und einem Beweiser P. Um die Mitgliedschaft von w in L zu entscheiden, stellt V Fragen, welche P beantwortet. Nach polynomiell in $|w|$ beschränkter Anzahl Fragen, muss V eine Entscheidung für oder gegen $w \in L$ fällen. V ist in seiner Laufzeit polynomiell in $|w|$ beschränkt, dementsprechend sind auch die Antworten, die P gibt, polynomiell in $|w|$ beschränkt. Insbesondere kann P sich an in diesem Dialog schon vorhandene Korrespondenz erinnern. Die Laufzeit für P ist nicht beschränkt. Die Wahrscheinlichkeit, dass V mit P ein Wort akzeptiert, wird mit $\mathbb{P}((V,P)$ akzeptiert $w)$ bezeichnet.*

IP wird nun wie folgt über ein interaktives Beweissystem definiert.

Definition 2. *Die Sprache L ist in IP genau dann, wenn ein Überprüfer V existiert, mit*

1. *es gibt einen Beweiser P, so dass für alle Worte w aus L gilt*

$$\mathbb{P}((V,P) \text{ akzeptiert } w) \geq 2/3,$$

2. *für alle Beweiser P und für alle Worte w nicht aus L gilt*

$$\mathbb{P}((V,P) \text{ akzeptiert } w) \leq 1/3.$$

2 $IP \subseteq PSPACE$

Defacto existiert für jede Sprache aus IP ein $PSPACE$-Algorithmus, welcher diese Sprache entscheidet.

Theorem 1. $IP \subseteq PSPACE$.

Beweis. Sei $L \in IP$ eine beliebige Sprache. Dann existiert ein Überprüfer V für L. Sei w ein beliebiges Wort. Ziel ist es nun,

$$z = max\{\mathbb{P}((V,P) \text{ akzeptiert } w) | P \text{ ist Beweiser für } L\} \tag{1}$$

zu berechnen. Anhand von z kann man entscheiden, ob $w \in L$. Falls $z \leq 1/3$, so ist w nicht in L, falls $z \geq 2/3$, so ist w in L.

z wird durch den im folgenden beschrieben Simulationsalgorithmus berechnet.

Sei $p(|w|)$ das Polynom, dass die Anzahl der von V gestellten Fragen, die Anzahl der von V in jedem Schritt benutzten Zufallsbits und die Länge der von P gegebenen Antworten beschränkt. Ein solches Polynom muss existieren, da $L \in IP$ ist.

Die folgende Funktion simuliert einen Schritt von V und gibt den Mittelwert der ermittelten Akzeptanz-Wahrscheinlichkeiten zurück. Der initiale Aufruf der Funktion liefert z.

Algorithmus 2.1: V-MITTELUNG()

$s \leftarrow 0$
$i \leftarrow 0$
for alle möglichen Zufallsstrings der Länge $p(|w|)$ auf dem Zufallsband
do
- Schreibe die aktuelle Konfiguration von V an das Ende des Bandes
- Simuliere einen Schritt von V
- **case**
  - **of**
    - V akzeptiert: $s \leftarrow s + 1$
    - V verweigert: $s \leftarrow s + 0$
    - V stellt eine Frage: $s \leftarrow s +$ P-MAXIMIMIERUNG()
- $i \leftarrow i + 1$
- Stelle die Konfiguration von V vom Ende des Bandes wieder her

return (s/i)

Die folgende Funktion simuliert alle möglichen Antworten von P und gibt die maximale Akzeptanz-Wahrscheinlichkeit der ausgewerteten V-Mittelungen zurück.

Algorithmus 2.2: P-MAXIMIMIERUNG()

$p \leftarrow 0$
for alle möglichen Antwortsstrings der Länge $p(|w|)$ auf dem Antwortsband
do
- Schreibe die aktuelle Konfiguration von V an das Ende des Bandes
- $q \leftarrow$ V-MITTELUNG()
- **if** $q > p$
  - **then** $p \leftarrow q$
- Stelle die Konfiguration von V vom Ende des Bandes wieder her

return (p)

Die Speicherung einer Konfiguration von V nimmt maximal die Länge $p(|w|)$ ein. Die aufgezählten Zufallsstrings sind in ihrer Länge durch $p(|w|)$ beschränkt. Damit ist die Anzahl der Mittelungen innerhalb der Schleife durch $2^{p(|w)}$ beschränkt. Dies lässt sich mit $p(|w|)$ Bits kodieren. Damit lassen sich alle lokal benutzen Variablen jeweils mit $p(|w|)$ Bits kodieren.

Durch die Anzahl der möglichen Fragen ist auch die Rekursionstiefe und damit die Anzahl der gleichzeitig gemerkten Konfigurationen von V durch $2p(|w|)$ beschränkt. □

3 $PSPACE \subseteq IP$

Da IP unter polynomieller Reduktion abgeschlossen ist, reicht es für $PSPACE \subseteq IP$ zu zeigen, dass irgendein beliebiges $PSPACE$-vollständiges Problem in IP liegt. Dies wird $TQBF$ sein.

3.1 Arithmetisierung von $TQBF$

QBF ist die Menge aller quantifizierten logischen Formeln, wobei alle Quantoren ein Präfix der Formel bilden und der Kern der Formel, welcher durch die Quantoren quantifiziert wird, als $3-KNF$-Ausdruck vorliegt. $TQBF \subset QBF$ ist die Menge aller wahren Formeln aus QBF.

Es gelten die beiden folgenden Theoreme, die hier ohne Beweis angegeben werden.

Theorem 2. *$TQBF$ ist $PSPACE$-vollständig.*

Theorem 3. *IP ist unter polynomieller Reduktion abgeschlossen.*

Gegeben eine QBF Formel Φ mit Kern $\varphi(b_1, ..., b_n)$. Die Kern-Arithmetisierung beschreibt, wie aus dem Kern $\varphi(b_1, ..., b_n)$ ein Polynom $p_\varphi(x_1, ..., x_n)$ erzeugt wird, so dass gilt

$$p_\varphi(x_1, ..., x_n) \quad = \quad \begin{cases} 0, \text{ falls } \varphi(b_1, ..., b_n) = falsch \text{ und} \\ 1, \text{ falls } \varphi(b_1, ..., b_n) = wahr \end{cases} \tag{2}$$

unter jeder Belegung von $b_1, .., b_n$ mit der Zuordnung

$$x_i \quad := \quad \begin{cases} 0, \text{ falls } b_i = falsch \text{ und} \\ 1, \text{ falls } b_i = wahr. \end{cases} \tag{3}$$

Man erhält solch ein Polynom durch folgende Zuordnung.

Als Abkürzung wird der Operator $\oplus$ für zwei Funktionen definiert als

$$f \oplus g \quad := \quad 1 - (1 - f)(1 - g). \tag{4}$$

Seien die Ausdrücke $\varphi_1(b_1, ..., b_n)$ und $\varphi_2(b_1, ..., b_n)$ geben. Dann ist

$$\begin{aligned} p_{b_i}(x_1, ..., x_n) &:= x_i && (5)\\ p_{\varphi_1 \wedge \varphi_2}(x_1, ..., x_n) &:= p_{\varphi_1}(x_1, ..., x_n) p_{\varphi_2}(x_1, ..., x_n), && (6)\\ p_{\varphi_1 \vee \varphi_2}(x_1, ..., x_n) &:= p_{\varphi_1}(x_1, ..., x_n) \oplus p_{\varphi_2}(x_1, ..., x_n), && (7)\\ p_{\neg\varphi_1}(x_1, ..., x_n) &:= 1 - p_{\varphi_1}(x_1, ..., x_n). && (8) \end{aligned}$$

Zum Beispiel ergibt $(b_1 \vee b_2) \wedge b_3$ das Polynom $(1 - (1 - x_1)(1 - x_2))x_3$.

Man kann leicht zeigen, dass das Schema (5)-(8) unter der Belegungszuordnung (3) die Forderung (2) erfüllt.

Die Quantoren von Φ werden durch folgende Rekursionsvorschrift arithmetisiert, wobei die Nummerierung der Variablen b_i bzw. x_i so ist, dass die Quantoren im Quantorenpräfix von Φ in aufsteigender Reihenfolge auf die b_i wirken.

$$\begin{aligned} f_{n,n}(x_1, ..., x_n) &:= p_\varphi(x_1, ..., x_n) && (9)\\ f_{i,i}(x_1, ..., x_i) &:= \begin{cases} \text{a) } f_{i+1,0}(x_1, ..., x_i, 0) f_{i+1,0}(x_1, ..., x_i, 1) \\ \quad \text{falls } x_{i+1} \text{ durch } \forall \text{ quantifiziert ist und} \\ \text{b) } f_{i+1,0}(x_1, ..., x_i, 0) \oplus f_{i+1,0}(x_1, ..., x_i, 1) \\ \quad \text{falls } x_{i+1} \text{ durch } \exists \text{ quantifiziert ist} \end{cases} && (10)\\ f_{i,j}(x_1, ..., x_i) &:= L_{j+1}(f_{i,j+1}(x_1, ..., x_i)) && (11)\\ L_j(f(x_1, ..., x_j, ..., x_i)) &:= \begin{array}{l} f(x_1, ..., 1, ..., x_i) x_j + \\ f(x_1, ..., 0, ..., x_i)(1 - x_j). \end{array} && (12) \end{aligned}$$

Der Operator L_j linearisiert f in der Variablen x_j, dabei behält die Funktion ihre Werte an den interessierenden Stellen aus $\{0, 1\}^i$ bei. Man stelle sich die Linearisierung als Ersetzen aller Exponenten von x_j größer als 1 durch 1 vor. Zum Beispiel ergibt $L_1(x_1^2 + 6x_1^5 x_2 + 7x_1^3) = 8x_1 + 6x_1 x_2$.

Würde die Linearisierung nicht vorgenommen werden, so könnte der Grad der Polynome f_i exponentiell wachsen und demnach würden exponentiell viele Koeffizienten im Polynom es unmöglich machen, das Polynom in polynomieller Zeit zu bearbeiten.

Man beobachte $f_{0,0} = 1$ bedeutet, Φ ist wahr, und $f_{0,0} = 0$ bedeutet, Φ ist falsch.

Eine für den Beweis sehr wichtige Feststellung ist, dass die Polynome $f_{n,.}$ höchstens vom Grad $3m$, wobei $m \leq |\Phi|$ die Anzahl der Klauseln ist, und die Polynome $f_{i<n,.}$ höchstens vom Grad 2 in jeder freien Variable sein können. Ersteres kommt durch die $3 - KNF$-Form des Kerns zustande und letzteres wird durch die Linearisierung erreicht.

3.2 $TQBF \in IP$

Theorem 4. *$TQBF \in IP$.*

Beweis. Sei Φ eine QBF Formel mit Kern φ. Der Kern wird wie beschrieben in p_φ übersetzt. Der Überprüfer V hangelt sich schrittweise durch die Arithmetisierungsvorschrift der Quantoren, wobei er auch die Linearisierungsschritte einzeln ausführt.

Sei der Fall betrachtet, dass $\Phi \in TQBF$. Am Anfang des Dialoges fordert V den Beweiser P auf, die kleinste Primzahl p mit $12|\Phi|^2 \leq p \leq (12|\Phi|+1)^4$ zu bestimmen, diese, einen Primzahlbeweis und $f_{0,0}$ zu senden. Die Einschränkungen von p werden am Ende des Beweises benutzt. Man erinnere sich daran, dass $PRIMES \in NP$ und deshalb diese Form der Primzahlbestimmung hier machbar ist. P sendet p, den Beweis und $f_{0,0} = 1$.

V empfängt die Primzahl und einen Primzahlbeweis, verifiziert dies und stellt fest, ob $f_{0,0} = 1$. P wird nun aufgefordert, in Zukunft alle Polynome modulo p zu schicken (in $\mathbb{Z}_{(p)}$) - dadurch können die Koeffizienten mit linearer Anzahl von Bits kodiert werden. Weil der Grad der Polynome linear beschränkt ist und V immer ein Polynom in nur einer freien Variable anfordert, ist auch die Anzahl der Koeffizienten linear beschränkt, und dadurch die Nachrichtenlänge polynomiell beschränkt.

P muss jetzt V davon überzeugen, dass das berechnete Polynom $f_{0,0} = 1$ wirklich $f_{0,0}$ ist. Um das zu tun, hangeln sich P und V iterativ durch die $f_{i,j}$. Dazu wird folgende Fallunterscheidung wiederholt, bis V das Polynom $f_{n,n}$ erhalten hat. Begonnen wird mit $i = 1$, $j = 0$.

Wiederhole bis $i = n$, $j = n$:

1. Fall $j = 0$: (10)-Schritt. Quantor-Schritt.
   - V fordert $f_{i,0}(r_1, ..., r_{i-1}, x_i)$ modulo p von P.
   - P sendet $f_{i,0}(r_1, ..., r_{i-1}, x_i)$ modulo p an V.
   - (a) (10a)-Schritt, falls b_i durch $\forall$ quantifiziert ist.
     V prüft
     $$f_{i,0}(r_1, ..., r_{i-1}, 0) f_{i,0}(r_1, ..., r_{i-1}, 1) \equiv_p f_{i-1,i-1}(r_1, ..., r_{i-1}).$$
     (b) (10b)-Schritt, falls b_i durch $\exists$ quantifiziert ist.
     V prüft
     $$f_{i,0}(r_1, ..., r_{i-1}, 0) \oplus f_{i,0}(r_1, ..., r_{i-1}, 1) \equiv_p f_{i-1}(r_1, ..., r_{i-1}).$$
   - V wählt r_i zufällig. j wird auf 1 gesetzt.
2. Fall $j > 0$: (12)-Schritt. Linearisierungs-Schritt.
   - V fordert $f_{i,j}(r_1, ..., x_j, ..., r_i)$ modulo p von P.
   - P sendet $f_{i,j}(r_1, ..., x_j, ..., r_i)$ modulo p an V.
   - V prüft
     $$(1 - r_j) f_{i,j}(r_1, ..., 0, ..., r_i) + r_j f_{i,j}(r_1, ..., 1, ..., r_i) \equiv_p f_{i,j-1}(r_1, ..., r_i).$$
   - V wählt r_j zufällig. Falls $j < i$, so wird j um 1 erhöht, sonst wird i um 1 erhöht und j auf 0 gesetzt.

V prüft nun $p_\varphi(r_1...,r_n) \equiv_p f_{n,n}(r_1,...,r_n)$ und akzeptiert.

Man erkennt sofort, dass ein korrekter Prüfer den Überprüfer mit Wahrscheinlichkeit 1 überzeugen kann.

Es wird nun der Fall $\Phi \notin TQBF$ betrachtet.

Es gibt drei Klassen von Beweisern: die ehrlichen, die dummen und die kleveren. Die ehrlichen werden sofort $f_{0,0} = 0$ an den Überprüfer schicken, welcher dann sofort verweigert. Die dummen schicken inkonsistente Polynome oder Müll an den Überprüfer, welcher der mittels (10) und (12) verweigert. Deshalb kann man o.B.d.A. annehmen, dass ein kleverer Beweiser P vorliegt, der immer konsistente Polynome schickt.

Seien die Polynome, die P schickt, $\widetilde{f}_{i,j}(x_k)$, wobei x_k die freie Variable darstellt. Dann schickt er $\widetilde{f}_{0,0} = 1$ statt $f_{0,0} = 0$ und lügt somit im allerersten Polynom. Um nicht aufzufallen, muss P alle nachfolgenden Polynome so schicken, dass die obige Rekursionsvorschrift erhalten bleibt. Jedoch muss P versuchen, am Ende das wahre Polynom $f_{n,n}$ zu schicken, da dies die einzige Stelle ist, wo V die Lügen aufdecken kann. Sei $d_{i,j}(x_k) := f_{i,j}(r_1,...,x_k,...,r_i) - \widetilde{f}_{i,j}(x_k)$. Falsches und wahres Polynom können nur dann übereinstimmen, wenn der Überprüfer zufällig eine Nullstelle r_k von $d_{i,j}(x_k)$ wählt, dann kann P mit dem wahren Polynom im nächsten Schritt weitermachen, da dann V $\widetilde{f}_{i,j}(x_k)$ und $f_{i,j}(r_1,...,x_k,...,r_i)$ anhand von r_k nicht mehr unterscheiden kann.

Ein wichtiger Satz der Algebra besagt, dass eine Polynom vom Grad n in jedem beliebigen Körper höchstens n Nullstellen besitzen kann. Seien f und g zwei unterschiedliche Polynome vom Grad höchstens n, p eine Primzahl und $0 \leq r < p$ eine ganze Zahl. Die Wahrscheinlichkeit für $f(r) \equiv_p g(r)$ ist gleich der Wahrscheinlichkeit, dass r eine Nullstelle von $f - g$ ist, also kleiner als n/p. Genau diesen Fakt macht sich der Überprüfer zum Vorteil.

Sei m die Anzahl der Klauseln in φ, dann ist $m \leq |\Phi|$. Die letzten geschickten Polynome $f_{n,.}(r_1,...,x_k,...,r_n)$ können dann höchstens vom Grad $3m$ in x_k sein. Die Polynome $f_{i<n,.}(r_1,...,x_k,...,r_n)$ sind höchstens vom Grad 2 in x_k. Alle anderen Grade kann der Überprüfer als Lüge zurückweisen. Es gibt insgesamt $(n^2+n)/2$ Polynome, von denen haben also n Maximalgrad $3m$ und die restlichen $(n^2 - n)/2$ ist der Grad höchstens 2. Für die Wahrscheinlichkeit

$$q_{i,j} := \mathbb{P}(V \text{ wählt im Schritt } (i,j) \text{ eine Nullstelle von } f_{i,j})$$

gilt somit

$$q_{i,j} \leq grad(f_{i,j})/p \leq \begin{cases} 3m/p, \text{ falls } i = n \text{ und} \\ 2/p, \text{ sonst.} \end{cases} \tag{13}$$

$q_{i,j}$ gibt also eine Obergrenze dafür an, mit welcher Wahrscheinlichkeit P im Schritt (i,j) V überlisten kann. Summiert man diese auf, so erhält man eine obere Schranke für

$$\mathbb{P}((V,P) \text{ akzeptiert } w) \leq \sum_{i,j} q_{i,j} \leq \frac{n^2 - n + 3mn}{p}. \tag{14}$$

Um $\mathbb{P}((V,P)$ akzeptiert $w) \leq 1/3$ zu erreichen, muss $p \geq 3n^2 - 3n + 9mn$ und mit $n \leq |\Phi|$ ist somit $p \geq 12|\Phi|^2$ hinreichend. Es wird nun noch gezeigt, dass eine solche Primzahl mit polynomieller Länge gefunden werden kann, damit das Protokoll funktioniert.

Ein Satz der elementaren Zahlentheorie legt folgende Schranken an die Anzahl der Primzahlen $\pi(n)$, die kleiner gleicher n sind, mit $n \geq 3$:

$$\sqrt{n} \leq \pi(n) \leq n. \tag{15}$$

Um zu zeigen, dass es immer eine Primzahl p mit $n < p \leq (n+1)^2$ gibt, reicht es zu zeigen, dass $\pi((n+1)^2) - \pi(n) \geq 1$ ist. Wenn man (15) benutzt, erhält man:

$$\pi((n+1)^2) - \pi(n) \geq \sqrt{(n+1)^2} - n = 1. \tag{16}$$

Daraus folgt nun auch die Existenz einer Primzahl zwischen $12|\Phi|^2$ und $(12|\Phi|+1)^4$, welche dann natürlich auch von polynomieller Länge ist. □

3.3 Beispiel

In diesem Kapitel wird das Protokoll anhand eines Beispiels veranschaulicht.

Sei die Formel $\Phi = \forall x_1 \exists x_2 \forall x_3 : x_1 \leftrightarrow (x_2 \rightarrow x_3)$ betrachtet. Dies lässt sich als quantifizierte KNF schreiben:

$$\Phi = \forall x_1 \exists x_2 \forall x_3 : \overline{x}_1 \overline{x}_2 x_3 \wedge x_1 \overline{x}_2 \overline{x}_3 \wedge x_1 \overline{x}_3 x_2 \wedge x_1 x_2 x_3.$$

Im folgenden werden die Polynome gebildet.

$$f_{3,3}(x_1,x_2,x_3) = p_\varphi(x_1,x_2,x_3) = \begin{array}{l}(1 + x_1x_2x_3 - x_3x_2)\\ (1 + x_1x_2x_3 - x_1x_2)\\ (1 - x_1x_2x_3 + x_1x_3 - x_3 + x_3x_2)\end{array} \tag{17}$$

Dieses Polynom 3.Ordnung wird nun mittels des L_j-Operators linearisiert.

$$\begin{aligned}
f_{3,2}(x_1,x_2,x_3) &= && (18)\\
L_3(f_{3,3}(x_1,x_2,x_3)) &= x_3 f_{3,3}(x_1,x_2,1) + (1-x_3) f_{3,3}(x_1,x_2,0) && (19)\\
&= \begin{array}{l}1 - x_1x_2 - x_3 - x_3x_1x_2 + x_3x_1 + x_3x_2\\ - x_3x_2^2 + 2x_3x_1x_2^2 + x_3x_1^2x_2 - x_3x_1^2x_2^2\end{array} && (20)\\
f_{3,1}(x_1,x_2,x_3) &= && (21)\\
L_2(f_{3,2}(x_1,x_2,x_3)) &= x_2 f_{3,2}(x_1,1,x_3) + (1-x_2) f_{3,2}(x_1,0,x_3) && (22)\\
&= 1 - x_3 + x_1x_3 + x_1x_2x_3 - x_1x_2 && (23)\\
f_{3,0}(x_1,x_2,x_3) &= && (24)\\
L_1(f_{3,1}(x_1,x_2,x_3)) &= x_1 f_{3,1}(1,x_2,x_3) + (1-x_1) f_{3,1}(0,x_1,x_2,x_3) && (25)\\
&= 1 - x_3 + x_1x_3 + x_1x_2x_3 - x_1x_2 && (26)
\end{aligned}$$

Nun wird der Operator $\forall x_3$ arithmetisiert.

$$\begin{aligned}
f_{2,2}(x_1,x_2) &= f_{3,0}(x_1,x_2,0) f_{3,0}(x_1,x_2,1) && (27)\\
&= x_1 - x_1^2x_2 && (28)
\end{aligned}$$

Jetzt schließt sich wieder eine Linearisierung an.

$$\begin{aligned} f_{2,1}(x_1, x_2) &= && (29)\\ L_2(f_{2,2}(x_1, x_2)) &= x_2 f_{2,2}(x_1, 1) + (1 - x_2) f_{2,2}(x_1, 0) && (30)\\ &= x_1 - x_1^2 x_2 && (31)\\ f_{2,0}(x_1, x_2) &= && (32)\\ L_1(f_{2,1}(x_1, x_2)) &= x_1 f_{2,1}(1, x_2) + (1 - x_1) f_{2,1}(0, x_2) && (33)\\ &= x_1 - x_1 x_2 && (34) \end{aligned}$$

Es wird nun der Operator $\exists x_2$ arithmetisiert.

$$\begin{aligned} f_{1,1}(x_1) &= 1 - (1 - f_{2,0}(x_1, 0))(1 - f_{2,0}(x_1, 1)) && (35)\\ &= x_1 && (36) \end{aligned}$$

Der Linearisierungsschritt ist offensichtlich.

$$\begin{aligned} f_{1,0}(x_1) &= && (37)\\ L_1(f_{1,1}(x_1)) &= x_1 && (38) \end{aligned}$$

Es bleibt noch den Operator $\forall x_1$ zu arithmetisieren. Der Linearisierungsschritt fällt hier natürlich weg, da $f_{0,0}$ eine Konstante ist.

$$f_{0,0} = f_{1,0}(1) f_{1,0}(0) = 0 \quad (39)$$

Die Formel Φ ist somit nicht in $TQBF$. Es folgt ein möglicher Dialog, in dem P versucht V davon zu überzeugen, dass $\Phi \in TQBF$.

1.
   - V wünscht sich eine Primzahl p und fordert P auf, $f_{0,0}$ zu senden.
   - P sendet die Primzahl $p = 11$ und lügt $\widetilde{f}_{0,0} = 1$.
2.
   - V erhält $p = 11$ und einen Primzahlbeweis und verifiziert 11 als Primzahl.
     V erhält $\widetilde{f}_{0,0} = 1$ und stellt fest, dass $\Phi \in TQBF$ behauptet wird.
     V fordert nun P auf, $f_{1,0}(x_1)$ zu senden.
   - P kann nun aber nicht das wahre Polynom $f_{1,0}(x_1) = x_1$ senden, da V sofort die Inkonsistenz aufdecken würde.
     P sendet deshalb $\widetilde{f}_{1,0}(x_1) = 1$.
3.
   - V erhält $\widetilde{f}_{1,0}(x_1) = 1$ und stellt fest, dass
     $$\widetilde{f}_{0,0} = \widetilde{f}_{1,0}(0)\widetilde{f}_{1,0}(1).$$
     V wählt $r_1 = 10$ und fordert P auf, $f_{1,1}(x_1)$ zu senden.
   - P muss wiederholt lügen, da das wahre Polynom $f_{1,1}(x_1)$ inkonsistent zur vorherigen Lüge wäre.
     P sendet statt dessen $\widetilde{f}_{1,1}(x_1) = 1$.

4.
   - V erhält $\widetilde{f}_{1,1}(x_1) = 1$ und stellt fest, dass
   $$\widetilde{f}_{1,0}(r_1) = 1 = 10\widetilde{f}_{1,1}(1) + (1-10)\widetilde{f}_{1,1}(0).$$
   V wählt die Zahl $r_1 = 3$ und fordert P auf, $f_{2,0}(r_1, x_2)$ zu senden.
   - P berechnet $f_{1,1}(r_1) = 3 \neq 1 = \widetilde{f}_{1,1}(r_1)$. P muss weiterhin lügen und sendet $\widetilde{f}_{2,0}(r_1, x_2) = 3 - 2x_2$ statt $f_{2,0}(r_1, x_2) = 3 - 3x_2$.

5.
   - V erhält $\widetilde{f}_{2,0}(r_1, x_2)$ und prüft gegen:
   $$\widetilde{f}_{1,1}(r_1) = 1 = 1 - (1 - \widetilde{f}_{2,0}(r_1, 0))(1 - \widetilde{f}_{2,0}(r_1, 1)).$$
   V wählt die Zahl $r_2 = 9$ und wünscht sich $f_{2,1}(x_1, r_2)$ von P.
   - P berechnet $f_{2,0}(r_1, r_2) = 9 \neq 7 = \widetilde{f}_{2,0}(r_1, r_2)$ und muss seine Lügen fortsetzen. P sendet daraufhin $\widetilde{f}_{2,1}(x_1, r_2) = 2x_1$.

6.
   - V erhält $\widetilde{f}_{2,1}(x_1, r_2) = 2x_1$ und stellt fest, dass
   $$\widetilde{f}_{2,0}(r_1, r_2) = 7 = (1 - r_1)\widetilde{f}_{2,1}(0, r_2) + r_1\widetilde{f}_{2,1}(1, r_2).$$
   V wählt die Zahl $r_1 = 5$ und fordert P auf, $f_{2,2}(r_1, x_2)$ zu senden.
   - P berechnet $\widetilde{f}_{2,1}(r_1, r_2) = 1 \neq 0 = f_{2,1}(r_1, r_2)$ und muss auch dieses Mal lügen. P sendet $\widetilde{f}_{2,2}(r_1, x_2) = 3 + x_2$.

7.
   - V erhält $\widetilde{f}_{2,2}(r_1, x_2) = 3 + x_2$ und stellt fest, dass
   $$\widetilde{f}_{2,1}(r_1, r_2) = 1 = (1 - r_2)\widetilde{f}_{2,2}(r_1, 0) + r_2\widetilde{f}_{2,2}(r_1, 1).$$
   V wählt die Zahl $r_2 = 8$ und fordert P auf, $f_{3,0}(r_1, r_2, x_3)$ zu senden.
   - P berechnet $\widetilde{f}_{2,2}(r_1, r_2) = 0 \neq 3 = f_{2,2}(r_1, r_2)$ und muss auch dieses Mal lügen. P sendet $\widetilde{f}_{3,0}(r_1, r_2, x_3) = 3 + x_3$.

8.
   - V erhält $\widetilde{f}_{3,0}(r_1, r_2, x_3)$ und prüft gegen:
   $$\widetilde{f}_{2,2}(r_1, r_2) = 0 = \widetilde{f}_{3,0}(r_1, r_2, 0)\widetilde{f}_{3,0}(r_1, r_2, 1).$$
   V wählt die Zahl $r_3 = 2$ und wünscht sich $f_{3,1}(x_1, r_2, r_3)$ von P.
   - P berechnet $f_{3,0}(r_1, r_2, r_3) = 5 \neq 3 = \widetilde{f}_{3,0}(r_1, r_2, r_3)$ und muss auch diesmal lügen. P sendet nun $\widetilde{f}_{3,1}(x_1, r_2, r_3) = 3x_1 - 1$.

9.
   - V erhält $\widetilde{f}_{3,1}(x_1, r_2, r_3) = 3x_1 - 1$ und stellt fest, dass
   $$\widetilde{f}_{3,0}(r_1, r_2, r_3) = 3 = (1 - r_1)\widetilde{f}_{3,1}(0, r_2, r_3) + r_1\widetilde{f}_{3,1}(1, r_2, r_3).$$
   V wählt die Zahl $r_1 = 1$ und fordert P auf, $f_{3,2}(r_1, x_2, r_3)$ zu senden.

- P berechnet $\widetilde{f}_{3,1}(r_1, r_2, r_3) = 1 \neq 8 = f_{3,1}(r_1, r_2, r_3)$ und muss auch dieses Mal lügen. P sendet $\widetilde{f}_{3,2}(r_1, x_2, r_3) = 7x_2 + 1$.

10. • V erhält $\widetilde{f}_{3,2}(r_1, x_2, r_3) = 7x_2 + 1$ und stellt fest, dass

$$\widetilde{f}_{3,1}(r_1, r_2, r_3) = 2 = (1 - r_2)\widetilde{f}_{3,1}(r_1, 0, r_3) + r_2\widetilde{f}_{3,1}(r_1, 1, r_3).$$

  V wählt die Zahl $r_2 = 7$ und fordert P auf, $f_{3,3}(r_1, r_2, x_3)$ zu senden.

- P berechnet $\widetilde{f}_{3,2}(r_1, r_2, r_3) = 0 \neq 8 = f_{3,2}(r_1, r_2, r_3)$ und muss auch dieses Mal lügen. P sendet $\widetilde{f}_{3,3}(r_1, x_2, r_3) = 7x_3 + 1$.

11. • V erhält $\widetilde{f}_{3,2}(r_1, x_2, r_3) = 7x_2 + 1$ und stellt fest, dass

$$\widetilde{f}_{3,1}(r_1, r_2, r_3) = 2 = (1 - r_2)\widetilde{f}_{3,2}(r_1, 0, r_3) + r_2\widetilde{f}_{3,2}(r_1, 1, r_3).$$

  V wählt die Zahl $r_2 = 7$ und fordert P auf, $f_{3,3}(r_1, r_2, x_3)$ zu senden.

- P berechnet $\widetilde{f}_{3,2}(r_1, r_2, r_3) = 0 \neq 8 = f_{3,2}(r_1, r_2, r_3)$ und muss auch dieses Mal lügen. P sendet $\widetilde{f}_{3,3}(r_1, r_2, r_3) = 4x_3 - 2$.

12. • V erhält $\widetilde{f}_{3,3}(r_1, x_2, r_3) = 4x_3 - 2$ und stellt fest, dass

$$\widetilde{f}_{3,2}(r_1, r_2, r_3) = 2 = (1 - r_3)\widetilde{f}_{3,2}(r_1, r_2, 0) + r_3\widetilde{f}_{3,3}(r_1, r_2, 1).$$

  V wählt die Zahl $r_3 = 4$ und rechnet

$$\widetilde{f}_{3,3}(r_1, r_2, r_3) = 3 \neq 7\varphi(r_1, r_2, r_3)$$

  und lehnt ab.

Im letzten Schritt hätte V auch $r_3 = 5$ wählen können und hätte damit nicht erkannt, dass P gelogen hat. Im 5. Schritt hätte die Wahl von $r_2 = 0$ dazu geführt, dass P das wahre Polynom hätte schicken können. In diesem Fall hätte V keine Chance gehabt, die Lüge zu erkennen.

Literatur

[BM??] Mares, B.J.: *A detailed proof that $IP = PSPACE$.* Brown University.
http://www.cs.brown.edu/courses/gs019/papers/ip.pdf

[LT01a] Trevisan, Luca: *Lecture 18 of Computational Complexity.* U.C. Berkeley, 12.04.2001.
http://www.cs.berkeley.edu/ luca/cs278/notes/lecture18.ps

[LT01b] Trevisan, Luca: *Lecture 19 of Computational Complexity.* U.C. Berkeley, 17.04.2001.
http://www.cs.berkeley.edu/ luca/cs278/notes/lecture19.ps