
$IP = PSPACE$

Ein nicht-relativierender Beweis

SE Komplexität und Kryptographie

Gliederung

1. Was ist IP?

- a) Allgemeine Beschreibung
- b) Private/Public Coins – $AM[k]$

2. $IP \subseteq PSPACE$

3. $IP \supseteq PSPACE$

- a) TQBF
- b) Arithmetisierung

4. Warum nicht-relativierend?

Was ist IP?

- Grundidee: „Natürlicher“ Beweisvorgang
- Frage-/Antwortspiel zwischen Prüfer und Beweiser
- NP: Gib mir deinen Beweis!
- IP: Überzeuge mich von deinem Beweis!

Was ist IP?

- Prüfer ist „schwach“ – maximal PTIME
- Beweiser ist „stark“ – darf beliebige Menge an Zeit / Platz verbrauchen
- Beweiser darf sogar „lügen“ / „schummeln“

- Einzige Bedingung: Beweiser muss Prüfer *irgendwie* überzeugen

Was ist IP?

- Zwei Hauptarten von IP:
 - IP mit deterministischem Prüfer (dIP)
 - IP mit probabilistischem Prüfer (IP)
- Definition von IP zunächst anhand von dIP
- Aber: $dIP = NP$, während $IP = PSPACE$
- Mächtigkeit steigt enorm durch „Zufall“

Definition dIP

Seien $f, g : \{0,1\}^* \rightarrow \{0,1\}^*$ Funktionen und $x \in \{0,1\}^*$ ihre Eingaben.

Dann ist die Folge $\langle f, g \rangle(x)$ als Sequenz aus Strings $a_1 \dots a_k \in \{0,1\}^*$ mit :

$$a_1 = f(x)$$

$$a_2 = g(x, a_1)$$

...

$$a_{2i+1} = f(x, a_1 \dots a_{2i})$$

$$a_{2i+2} = g(x, a_1 \dots a_{2i+1})$$

eine Interaktion zwischen f und g in k Runden, wobei

$out_y \langle f, g \rangle(x)$ die Ausgabe von $y \in f, g$ am Ende der Interaktion ist.

Definition dIP

- Sprache L hat dIP Beweis in k Runden wenn:
 - deterministische TM V bei Eingabe von (x, a_1, \dots, a_i) in PTIME läuft und dabei gilt:

$$x \in L \Rightarrow \exists P : \{0,1\}^* \rightarrow \{0,1\}^*, out_V \langle V, P \rangle(x) = 1$$

$$x \notin L \Rightarrow \forall P : \{0,1\}^* \rightarrow \{0,1\}^*, out_V \langle V, P \rangle(x) = 0$$

- Erweiterung davon bei IP: probabilistische TM V

$$x \in L \Rightarrow \exists P : \Pr[out_V \langle V, P \rangle(x) = 1] \geq 2/3$$

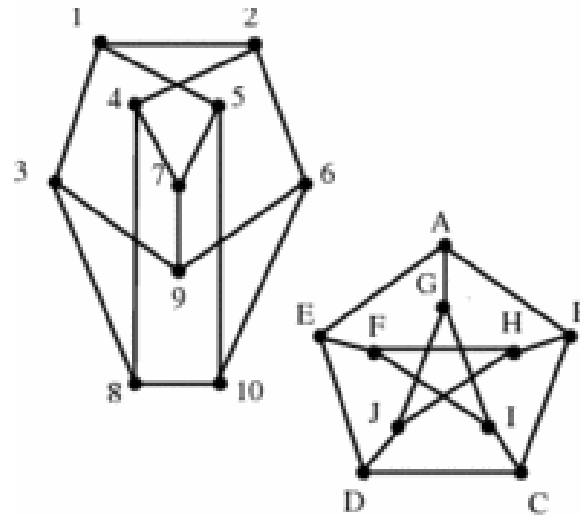
$$x \notin L \Rightarrow \forall P : \Pr[out_V \langle V, P \rangle(x) = 1] \leq 1/3$$

Private/Public Coins

- Ausreichend wenn Prüfer zufällig ist
- Kein Vorteil durch probabilistischen Beweiser
- Idee:
 - Wenn Beweiser „Recht hat“, dann hohe Akzeptanzwkt. des Prüfers (evtl. sogar = 1)
 - Aber: Kein Beweiser der „lügt“ darf Prüfer dazu bringen mit hoher Wkt. zu akzeptieren

Private/Public Coins

- Beispiel: IP Interaktion für Nichtisomorphismus zweier Graphen
- Gegeben: Zwei Graphen G_1, G_2
- Gesucht: Ist $G_1 \not\cong G_2$?



Private/Public Coins

Protokoll:

V: Wähle $i \in \{1,2\}$ zufällig. Permutiere G_i zufällig \rightarrow neuer Graph H . Sende H an P .

P: Finde Graph G_k aus dem H erzeugt wurde. Sende k an V .

V: Wenn $i = k$ akzeptiere. Sonst ablehnen.

Private/Public Coins

- Protokoll akzeptiert immer wenn $G_1 \not\equiv G_2$, da H dann eindeutig isomorph
- Wenn $G_1 \equiv G_2$, dann muss P raten, da $G_1 \equiv H \equiv G_2 \rightarrow \Pr[V_{\text{akzeptiert}}] \leq 1/2$
- Verbesserung auf $\leq 1/3$ möglich durch Wiederholung

Private/Public Coins

- Wiederum zwei Kategorien:
 1. Prüfer hält seine Zufallszahlen geheim (Private Coin / IP)
 2. Beweiser hat Zugriff auf Zufallszahlen (Public Coin / AM)
- Auf ersten Blick ist IP mächtiger, aber:
$$AM[k] \subseteq IP[k] \subseteq AM[k+2]$$
und sogar: $AM[k] = AM[2] = IP[2] = IP[k]$
$$k = \text{beliebig, aber konstant}$$



$IP \subseteq PSPACE$

IP \subseteq PSPACE

- Grundidee des Beweises:
 - Konstruktion von PSPACE Maschine, die beliebige IP Interaktionen simuliert
 - Simulation aller möglichen Pfade
- Dazu notwendig: Abschätzung wie viel Platz maximal lange IP Beweise benötigen
- PSPACE Maschine V mit:

$$\Pr[V \text{ accepts } w] = \max_p \Pr[V' \leftrightarrow P \text{ accepts } w]$$

IP \subseteq PSPACE

- Sei M_j = Sequenz an Nachrichten zwischen Prüfer und Beweiser und p die Beweislänge
- Dann kann der Beweis wie folgt zerlegt werden:

$[(V \leftrightarrow P)(w, r, M_j) = \text{accept}] \Leftrightarrow M_j$ um Nachrichten m_j erweitert werden kann sodass für alle m_j bis m_p gilt:

$$j \leq i < p \wedge i \text{ gerade} \Rightarrow m_{i+1} = V(w, r, M_i)$$

$$j \leq i < p \wedge i \text{ ungerade} \Rightarrow m_{i+1} = P(w, r, M_i)$$

$$m_p = \text{accept}; r \in \mathbf{Z}, r \text{ zufällig}$$

IP \subseteq PSPACE

- Dadurch gilt:

$$\begin{aligned} \Pr[V \leftrightarrow P \text{ accepts } w \text{ starting at } M_j] \\ = \Pr[(V \leftrightarrow P)(w, r, M_j) = \text{accept}] \end{aligned}$$

- Weitere Idee: Beweisverlauf bis zum accept/reject arithmetisieren

IP \subseteq PSPACE

- Definition der maximalen Akzeptanzwkt.

$$N_{M_j} = \begin{cases} 0 : j = p \wedge m_p = \text{reject} \\ 1 : j = p \wedge m_p = \text{accept} \\ \max_{m_{j+1}} \cdot (N_{m_{j+1}}) : j < p \wedge j \text{ ungerade} \\ \sum_{m_{j+1}} (\text{Pr}_r[V(w, r, M_j)]) : j < p \wedge j \text{ gerade} \end{cases}$$

IP \subseteq PSPACE

- Wenn Maschine V dieses N für alle M_j berechnen kann, so gilt $IP \subseteq PSPACE$
- Zu zeigen:
 1. N_{M_0} in PSPACE berechenbar
 2. $N_{M_0} = \Pr[V \text{ accepts } w]$
- 1.) trivial, Rekursionstiefe von N_{M_0} durch p begrenzt, also in PSPACE berechenbar

IP \subseteq PSPACE

- 2.) Induktion über Länge p , M_p bis M_0
- Induktionsanfang:
 - In M_p gilt nach Def. $N_{MP} = \Pr[V \text{ accepts } w]$
- Induktion $j+1 \rightarrow j$:
 - Für $j+1$ gilt: $N_{M_{j+1}} = \Pr[V \text{ accepts } w \text{ starting at } M_j]$
 - 2 Fälle:
 - j gerade (Prüfer) oder j ungerade (Beweiser)

IP \subseteq PSPACE

- Für gerades j (Prüfer) gilt:

$$\begin{aligned} N_{M_j} &= \sum_{m_{j+1}} (\Pr_r[V(w, r, M_j) = m_{j+1}] \cdot N_{m_{j+1}}) \\ &= \sum_{m_{j+1}} (\Pr_r[V(w, r, M_j) = m_{j+1}]) \cdot N_{m_{j+1}} \\ &= \Pr[V \text{ accepts } w \text{ starting at } M_{j+1}] \end{aligned}$$

IP \subseteq PSPACE

- Für ungerades j (Beweiser) Annahme des bestmöglichen Beweisers
- Dieser wird immer N_{M_j} maximieren, da er immer die korrekte, beste Antwort wählt.

$$\begin{aligned} N_{m_j} &= \max_{m_{j+1}} \cdot (N_{m_{j+1}}) \\ &= \max_{m_{j+1}} \cdot \Pr[V \text{ accepts } w \text{ starting at } M_{j+1}] \\ &= \Pr[V \text{ accepts } w \text{ starting at } M_j] \end{aligned}$$



$IP \supseteq PSPACE$

IP \supseteq PSPACE

- Wie kann man IP \supseteq PSPACE zu zeigen?
 1. PSPACE-Complete Problem wählen
 2. IP Protokoll finden, dass dies löst
 3. Zeigen dass :

$$x \in L \Rightarrow \exists P : \Pr[out_V \langle V, P \rangle(x) = 1] \geq 2/3$$

$$x \notin L \Rightarrow \forall P : \Pr[out_V \langle V, P \rangle(x) = 1] \leq 1/3$$

TQBF

- Gewähltes Problem:
True Quantified Boolean Formula (TQBF)
- Gegeben eine quantifizierte boolesche Formel, entscheide ob diese erfüllbar ist

- d.h.: $[Q_1x_1Q_2x_2 \dots Q_nx_n\varphi : Q \in \{A, E\}] \Leftrightarrow$
 $\forall Q_i \text{ gilt: } \begin{cases} Q_ix_i\varphi = (\varphi_{xi=0} \wedge \varphi_{xi=1}) = \textit{true}, \text{ wenn } Q_i = \forall \\ Q_ix_i\varphi = (\varphi_{xi=0} \vee \varphi_{xi=1}) = \textit{true}, \text{ wenn } Q_i = \exists \end{cases}$

TQBF

- Bemerkung: IP muss nur „einfache“ QBFs lösen, da jede QBF in „einfache“ QBF umformbar (höchstens quadr. Wachstum)
- Einfache QBF:
 1. Geschlossene Form (Alle x_i sind gebunden)
 2. Zwischen Variablen und ihren Quantoren steht höchstens ein Allquantor

$$\forall x_1 \forall x_2 \exists x_3 [(x_1 \wedge x_2) \wedge \forall x_4 (x_2 \wedge x_3 \wedge x_4)]$$

Arithmetisierung

- Problem:
 - Wie kann sich der Prüfer in PTIME vom Ergebnis überzeugen?
 - Wie kann er einen lügenden Beweiser ausschließen?
- Idee: Andere Darstellung der QBF
→ Arithmetisierung

Arithmetisierung

■ Methode:

1. Ersetze Variable x_i durch neue Variable $z_i \in \mathbb{Z}$
2. Ersetze:
 - $\neg x_i$ durch $(1 - z_i)$
 - \wedge durch Multiplikation
 - \vee durch Addition,
 - \forall durch $\prod_{z_i \in \{0,1\}^*}$
 - \exists durch $\sum_{z_i \in \{0,1\}^*}$

Arithmetisierung

- Beispiel:

$$B = \forall x_1 \exists x_2 [(x_1 \wedge x_2) \wedge \exists x_3 (\neg x_2 \wedge x_3)]$$

ergibt :

$$A = \prod_{z_1 \in \{0,1\}} \sum_{z_2 \in \{0,1\}} \left[(z_1 \cdot z_2) + \sum_{z_3 \in \{0,1\}} (1 - z_2) \cdot z_3 \right] = 2$$

- Es gilt: $A \neq 0 \Leftrightarrow$ QBF erfüllbar

Arithmetisierung

- Problem für Prüfer: Wenn B wahr ist, kann A sehr große Werte annehmen.
- Obere Schranke:

$$B = \forall x_1 \forall x_1 \dots \forall x_{n-1} \exists x_n (x_n \vee \neg x_n) \Rightarrow A = O\left(2^{2^n}\right)$$

- Lösung: Modulo-Rechnung
→ Es kann gezeigt werden, dass:

$$B \text{ ist wahr} \Leftrightarrow A \not\equiv 0 \pmod{p} : \text{länge}(p) = O(n^c)$$

IP \supseteq PSPACE

- Es verbleibt zu zeigen:
 - Prüfer erkennt korrekte Arithmetisierung eines korrekten Beweisers
 - Prüfer erkennt falsche Arithmetisierung **aller** betrügenden Beweiser

$$B \text{ ist wahr} \Leftrightarrow A \neq 0 \pmod{p}$$

IP \supseteq PSPACE

- Grundidee: Eine Arithmetisierung A lässt sich immer in 2 Teile teilen und als Polynom darstellen:

$$B = \forall x_1 [\neg x_1 \vee \exists x_2 \forall x_3 (x_1 \wedge x_2) \vee x_3]$$

$$A = \prod_{z_1 \in \{0,1\}} \left[(1 - z_1) + \sum_{z_2 \in \{0,1\}} \prod_{z_3 \in \{0,1\}} (z_1 \cdot z_2 + z_3) \right] = 2$$

$$A_2 = \left[(1 - z_1) + \sum_{z_2 \in \{0,1\}} \prod_{z_3 \in \{0,1\}} (z_1 \cdot z_2 + z_3) \right] = q(z_1) = z_1^2 + 1$$

IP \supseteq PSPACE

■ Idee des IP-Protokolls:

1. Beweiser berechnet den Wert $a=(A \bmod p)$
2. Prüfer testet ob a korrekt ist, indem er A in $A_1+A_2 (\Sigma)$ bzw. $A_1 * A_2 (\Pi)$ teilt. Dabei ist A_1 ein von V auswertbares Polynom mit dem Wert a_1 und A_2 beginnt beim ganz links stehende Quantor.
3. P und V wiederholen dann die folgenden Vereinfachungsschritte

IP \supseteq PSPACE

1. Wenn A_2 leer ist, so hält V und akzeptiert wenn $a = a_1$
2. Wenn A_1 nicht leer ist, so ersetzt V A durch A_2 (dabei verschwindet ein Quantor) und ersetzt a durch $[a - a_1 \pmod{p}]$ bzw. $[a / a_1 \pmod{p}]$ abhängig vom Quantor
3. Bei Teilung $0/0$ akzeptiert V , bei $x/0$ lehnt es ab.
4. Ansonsten sendet P das Polynom $q(z_i)$ von A_2
5. V testet \pmod{p} ob: $a = q(0) + q(1)$ oder $a = q(0) \cdot q(1)$, sendet ein zufälliges r an P und ersetzt A durch $A'(z_i = r)$ und a durch $q(r)$

IP \supseteq PSPACE

Beispiel:
$$A = \sum_{z_1 \in \{0,1\}} \prod_{z_2 \in \{0,1\}} (3 \cdot z_2 + z_3)$$

$$a = 10 - (-2) = 12$$

$$A_2 = \prod_{z_2 \in \{0,1\}} (3 \cdot z_2 + z_3) = q(z_2) = 9z_2^2 + 3z_2$$

- V erhält nun dieses q , testet ob:
 $q(0) + q(1) = 0 + 12 = 12 = a$, und wählt zufällig $z_2 = 2$.
- Anmerkung: Wären die QBF's nicht einfach, wäre evtl. $\deg(q) = 2^{n-1} \rightarrow$ nicht in PTIME auswertbar

IP \supseteq PSPACE

- Durch $z_2=2$ ergibt sich für P dann:

$$A = \prod_{z_2 \in \{0,1\}} (3 \cdot 2 + z_3)$$

$$a = q(2) = 9 \cdot 4 + 3 \cdot 2 = 42$$

$$A_2 = z_3 + 6 = q(z_3)$$

- V testet ob $q(0) \cdot q(1) = 6 \cdot 7 = 42 = a$ und wählt dann zufällig $z_3 = 5$ und testet alleine, ob:

$$A_2(z_3 = 5) = 6 + 5 = q(5) = a$$

IP \supseteq PSPACE

- Erfüllt diese Abfolge die Anforderungen an korrekten IP Beweis?
- Es gilt: $x \in L \Rightarrow \exists P : \Pr[out_V \langle V, P \rangle(x) = 1] = 1$
da ehrlicher Beweiser immer korrekte Polynome und Werte für a findet.

IP \supseteq PSPACE

- Es gilt: $x \notin L \Rightarrow \forall P : \Pr[out_V \langle V, P \rangle(x) = 1] \leq 1/3$
da der Beweiser sonst Polynome q finden müsste, die sein falsches a unterstützen.

Aber:

- Grad von $q = t \rightarrow [q(0) \otimes q(1) = a]$ kann nur an höchstens t Stellen in \mathbb{Z}_p gelten
- Da aber $p = O(2^n)$, $t = O(n^c)$ ist $q(r)$ nur sehr unwahrscheinlich korrekt, was aber von V spätestens im letzten Schritt getestet wird.

Warum nicht-relativierend?

Warum nicht-relativierend?

- Hauptgrund analog zu relativierenden Beweisen von $P \stackrel{?}{=} NP$:
- Für gewisse Orakel A, B gilt:
 $P^A = NP^A$, und $P^B \neq NP^B$
- Für $IP = IPSPACE$ gilt ebenso:
- $IP^A = PSPACE^A$, und $IP^B \neq PSPACE^B$
→ Trotzdem ist $IP = PSPACE$ beweisbar

Warum nicht-relativierend?

- Genauer:
Für ein zufällig gewähltes Orakel A gilt:

$$\text{coNP} \subseteq \text{IP} = \text{PSPACE}$$

$$\text{PSPACE}^A \supseteq \text{coNP}^A \not\subseteq \text{IP}^A$$

$$\rightarrow \text{IP}^A \neq \text{PSPACE}^A$$

Warum nicht-relativierend?

- Interessant: Für IPP gilt:

$$x \in L \Rightarrow \exists P : \Pr[out_V \langle V, P \rangle(x) = 1] \geq 1/2$$

$$x \notin L \Rightarrow \forall P : \Pr[out_V \langle V, P \rangle(x) = 1] < 1/2$$

$$\forall A \text{ gilt : } IPP^A = PSPACE^A$$

- Für IP[2]/rpoly (IP[2] mit zufälligem polynomial großer Beraterfunktion) gilt:

$$IP[2] / rpoly = ALL$$

Quellen

Adi Shamir. „IP = PSPACE“. Journal of the ACM, volume 39, issue 4, p.869-877. October 1992.

Arora, Sanjeev, und Boaz Barak.

„Complexity Theory: A Modern Approach“.
Web draft. Princeton University, 2007.

R. Chang, B. Chor, et.al. “The random oracle hypothesis is false”, Journal of Computer and System Sciences 49(1):24-39, 1994