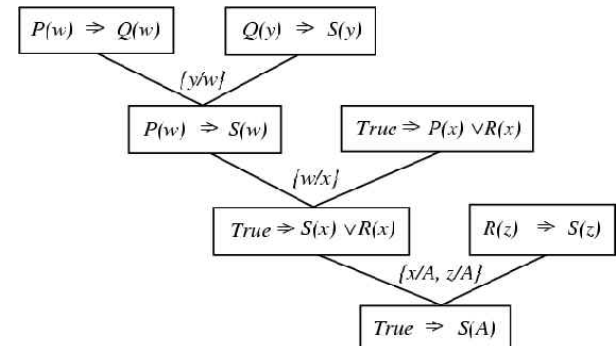


# Beweissysteme

Hartmut Klauck  
Universität Frankfurt  
WS 06/07  
25.10.



# Ein weiteres Beispiel für einen interaktiven Beweis

- QNR: Quadratic Nonresidues
  - Sei  $Z_n^* = \{x \in \mathbb{N} : 0 < x < n : \text{ggT}(x, n) = 1\}$
  - Eine Zahl  $x \in Z_n^*$  ist in  $QR(n)$ , wenn es ein  $y$  gibt, so dass  $y^2 = x \pmod n$   
[ $x$  ist ein quadratischer Rest mod  $n$ ]
  - $QNR = \{x, n : x \in Z_n^* \text{ und } x \text{ nicht } \in QR(n)\}$
- QNR liegt in co-NP, aber wahrscheinlich nicht in P
- Wir zeigen einen interaktiven Beweis für QNR

# Das Protokoll

- Ein Protokoll:
  - Der Verifizierer zieht ein Bit  $b$  sowie ein  $z$  aus  $Z_n^*$  uniform zufällig
  - Verifizierer setzt  $w =$ 
    - $z^2 \bmod n$  wenn  $b=1$
    - $x z^2 \bmod n$  wenn  $b=0$
  - Sendet  $w$
  - Der Beweiser:
    - Wenn  $w$  in  $QR(n)$  liegt, dann sendet er  $a=1$ , sonst  $0$
  - Verifizierer akzeptiert, wenn  $b=a$
- Analyse:
  - $x, n \in QNR$ :  $w \in QR(n)$  gdw  $b=1$ , denn wenn  $x$  nicht in  $QR(n)$  liegt, liegt auch  $xz^2$  nicht in  $QR(n)$ . Damit kann der Beweiser  $b$  bestimmen
  - $x, n$  nicht in  $QNR$ :  $x = y^2 \bmod n$ , d.h.  $w = y^2 z^2 \bmod n$  ist genau so verteilt wie  $z^2 \bmod n$  für  $z \in Z_n^*$  und gibt keine Information über  $b$ , d.h. Beweiser hat Erfolgswahrscheinlichkeit  $\leq \frac{1}{2}$ .

# Überblick

- Unsere Verifizierer benutzen private Zufallsbits (nicht dem Beweiser bekannt)
- Der Beweiser reagierte auf den Verifizierer, d.h. kein fester Beweis, sondern Interaktion
- Randomisierung ist wichtig

# Überblick

- Was ist die Rolle von privatem Zufall?
- Hilft Randomisierung ohne Interaktion?
- Können wir TAUT beweisen
  - randomisiert?
  - randomisiert mit Interaktion?
- Was ist die Kraft von generellen interaktiven Beweisen?

# Ein Exkurs

- Die polynomielle Hierarchie und die Klasse  $PSPACE$
- $PSPACE$  ist die Klasse der Sprachen, die durch Turingmaschinen mit polynomielltem Speicherplatz berechnet werden können

# Die polynomielle Hierarchie

- Eine Art, NP zu definieren ist so:
  - Die Menge aller Sprachen  $L$ , so dass es eine Sprache  $M \subseteq \{0,1\}^* \times \{\#\} \times \{0,1\}^*$  in P gibt, und  $k, c = O(1)$ , und  $L = \{x : \exists y : x\#y \in M \text{ und } |y| \leq cn^k\}$
  - Analog co-NP, tausche  $\exists$  gegen  $\forall$
- Wir können nun auch mehrere Quantoren betrachten
- Z.B. Die Menge aller Sprachen  $L$ , so dass es eine Sprache  $M \subseteq \{0,1\}^* \times (\{\#\} \times \{0,1\}^*)^2$  in P gibt, und  $k, c = O(1)$ , und  $L = \{x : \exists y \forall z : x\#y\#z \in M \text{ und } |y|, |z| \leq cn^k\}$
- Diese Menge nennt man  $\Sigma_2^P$
- Wenn man die Quantoren tauscht, so erhält man  $\Pi_2^P$

# Die polynomielle Hierarchie

- Es ist  $\Sigma_0^P = \Pi_0^P = P$
- $\Sigma_1^P = NP$ ;  $\Pi_1^P = \text{co-NP}$
- $\Sigma_j^P$  ist die Menge aller Sprachen  $L$ : es gibt eine Sprache  $M$  aus  $\Pi_{j-1}^P$ , und
$$L = \{x : \exists y : x\#y \in M \text{ und } |y| \leq cn^k\}$$
- $\Pi_j^P$  ist die Menge aller Sprachen  $L$ : es gibt eine Sprache  $M$  aus  $\Sigma_{j-1}^P$ , und
$$L = \{x : \exists y : x\#y \in M \text{ und } |y| \leq cn^k\}$$

# Die polynomielle Hierarchie

- PH ist die Vereinigung aller  $\Sigma_j^P$  für endliche  $j$  (d.h.  $j$  ist keine Funktion der Eingabelänge)
- Beispiel für ein  $\Sigma_2^P$  Problem: Gegeben eine aussagenlogische Formel in konjunktiver Normalform. Wir wollen wissen: gibt es eine Belegung der Variablen, die strikt mehr Klauseln erfüllt als alle anderen Belegungen?
- D.h. existiert eine Belegung  $a$ , so dass für alle Belegungen  $b$  gilt:  $a=b$  oder  $a$  erfüllt mehr Klauseln als  $b$ .

# Weiteres Beispiel

- Eingabe: Eine aussagenlogische Formel  $f$
- Ausgabe: Akzeptiere, wenn es keine andere Formel gibt, die von denselben Variablenbelegungen erfüllt wird und kürzer ist
- Das Problem liegt in  $\Pi_2^P$  :
  - Für alle Formeln  $g$  die kürzer als  $f$  sind gilt: es existiert eine Eingabe  $x$  mit  $f(x) \neq g(x)$

# PSPACE

- Was passiert, wenn wir erlauben, dass die Anzahl der Alternationen mit der Eingabelänge wächst?
- Wir betrachten nur das natürliche vollständige Problem: QBF, das Problem der quantifizierten Booleschen Formeln

# QBF

- **Definition:**
  - Eine Formel der Form  $\exists x_1 \forall x_2 \dots \exists x_n F(x_1, \dots, x_n)$ , wobei  $F$  eine aussagenlogische Formel ist, und jede der Variablen durch einen Quantor gebunden ist.
  - Eine QBF ist wahr oder falsch, mit der offensichtlichen Semantik
- **Theorem:** Das Problem, den Wahrheitswert einer QBF zu entscheiden, ist PSPACE-vollständig unter Polynomialzeitreduktionen
- Zum Beweis:
  - Das Problem liegt in PSPACE:  
Durchlaufe auf demselben Arbeitsspeicher alle Belegungen der Variablen und bestimme den Wahrheitswert der Formel. Dies geht bei linearem Platzverbrauch
  - Desweiteren braucht man eine Reduktion von jedem Problem in PSPACE auf QBF

# PSPACE und Spiele

- Weitere PSPACE-vollständige Probleme sind viele Versionen von 2-Personen Spielen, z.B.  
 $n \times n$  Schach mit Zeitbeschränkung  
(d.h. mit maximal  $\text{poly}(n)$  Zügen).
- Hier spielen zwei Gegner, und am Ende entscheidet ein Schiedsrichter (das Prädikat „ist schachmatt“)
- Das zugrunde liegende Berechnungsproblem ist: Hat Spieler Weiß eine Gewinnstrategie von einer bestimmten Position aus?
- D.h. Existiert ein Zug für Weiß, so dass für alle Züge von Schwarz ein Zug von Weiß existiert, ... so dass Weiß gewinnt
- In Beweissystemen ist der Schiedsrichter der Verifizierer, aber es gibt nur einen Spieler (den Beweiser).