

# Theoretische Informatik

Rainer Schrader

Institut für Informatik

15. Juni 2009

1/50

# Interaktive Nachweise

2/50

## Interaktives Nachweissystem

### Gliederung

- **interaktive Nachweise**
- co-Graphenisomorphie
- **IP** versus **NP**
- **IP** versus **PSPACE**

3/50

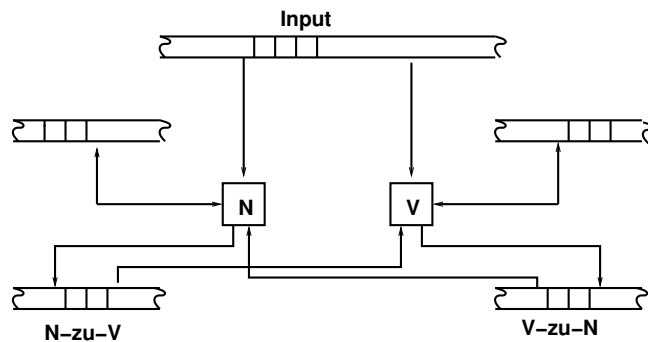
## Interaktives Nachweissystem

- das Folgende beruht auf einer Erfahrungstatsache:
- es ist
  - schwierig, einen Beweis so aufzuschreiben, dass jeder Leser ihn versteht
  - leichter, ihn zu erklären, wenn die Hörer Fragen stellen können
- wir wollen diese Interaktionsmöglichkeit übertragen auf Spracherkennungsprobleme

4/50

## Interaktives Nachweissystem

- unser **Nachweissystem** besteht aus:
  - zwei Turing-Maschinen  $N$  (**Nachweis**) und  $V$  (**Verifizierer**)
  - beide Maschinen können über spezielle Bänder wie folgt miteinander kommunizieren:

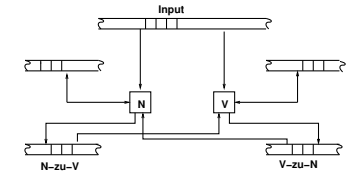


5/50

## Interaktives Nachweissystem

Es sollen folgende Bedingungen gelten:

- $V$  ist eine polynomielle zeitbeschränkte DTM
- $N$  ist eine DTM (unbeschränkt)
- $N$  und  $V$  arbeiten abwechselnd, wobei  $V$  beginnt
- jede Maschine darf zusätzlich die Kommunikationsbänder nutzen
- $V$ -zu- $N$ : write-only für  $V$ , read-only für  $N$
- $N$ -zu- $V$ : write-only für  $N$ , read-only für  $V$
- Anzahl und Länge der ausgetauschten Nachrichten sind durch Polynome beschränkt
- $V$  kann als aktive Maschine den Vorgang beenden, indem sie in den akzeptierenden oder verwerfenden Zustand übergeht
- das System akzeptiert/verwirft, falls  $V$  akzeptiert/verwirft



6/50

## Interaktives Nachweissystem

Eine Sprache  $L$  hat einen **deterministischen interaktiven Nachweis**  $DIP$ , wenn gilt:

- es existiert eine polynomielle zeitbeschränkte DTM  $V$  und
  1. es existiert eine DTM  $N^*$ , so dass  $(N^*, V)$  alle  $x \in L$  akzeptiert
  2. für alle DTM's  $N$  verwirft  $(N, V)$  alle  $x \notin L$ .
- Bedingung 1: für alle  $x \in L$  existiert ein kurzer Nachweis
- Bedingung 2: kein falscher Nachweis wird von  $V$  anerkannt
- sei **DIP** die Menge aller Sprachen, für die ein deterministischer interaktiver Nachweis existiert

7/50

## Interaktives Nachweissystem

**Lemma**

**DIP = NP**

**Beweis:**

- **NP** besteht aus genau den Sprachen, für die ein polynomieller Nachweis existiert
- wenn wir diesen auf das  $N$ -zu- $V$ -Band schreiben, zeigt dies **NP  $\subseteq$  DIP**.
- umgekehrt konstruieren wir eine NDTM, die alle möglichen (polynomiell langen) Mitteilungen von  $V$  an  $N$  rät
- ist  $L \in \mathbf{DIP}$ , so existiert eine Mitteilung, die die NDTM akzeptiert
- d.h. **DIP  $\subseteq$  NP**. □

8/50

## Interaktives Nachweissystem

Wir modifizieren unseren Ansatz und lassen für  $V$  eine **BPP**-Maschine zu:

- eine Sprache  $L$  hat einen **interaktiven Nachweis**, wenn gilt:

es existiert eine **BPP**-Maschine  $V$ , so dass gilt:

1. es existiert eine DTM  $N^*$ , so dass

$$\Pr(f_{(N^*, V)}(x) = 1) \geq \frac{2}{3} \text{ für alle } x \in L.$$

2. für alle DTM's  $N$  gilt

$$\Pr(f_{(N, V)}(x) = 0) \geq \frac{2}{3} \text{ für alle } x \notin L.$$

- sei **IP** die Menge der Sprachen mit einem interaktiven Nachweis

9/50

## Interaktives Nachweissystem

### Gliederung

- interaktive Nachweise
- **co-Graphenisomorphie**
- **IP** versus **NP**
- **IP** versus **PSPACE**

10/50

## Interaktives Nachweissystem

- da **DIP**  $\subseteq$  **IP** und **NP** = **DIP**, folgt **NP**  $\subseteq$  **IP**.
- es wird vermutet, dass **NP**  $\subsetneq$  **IP**
- wir wollen versuchen, diese Vermutung zu motivieren
- seien  $G_1 = (V_1, E_1)$  und  $G_2 = (V_2, E_2)$  zwei Graphen,
- $G_1$  und  $G_2$  heißen **isomorph**, wenn gilt:,
- es existiert eine Bijektion  $f : V_1 \rightarrow V_2$ , so dass

$$(u, v) \in E_1 \iff (f(u), f(v)) \in E_2.$$

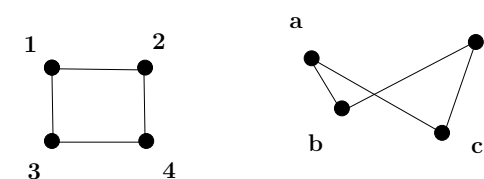
### co-Graphenisomorphie-Problem

- gegebenen zwei Graphen  $G_1, G_2$
- entscheide, ob  $G_1$  und  $G_2$  nicht isomorph sind.

11/50

## Interaktives Nachweissystem

### Beispiel:



12/50

## Interaktives Nachweissystem

- es gibt keine Ideen für einen kurzen Nachweis, dass zwei Graphen nicht isomorph sind
- daher vermutet man, dass das Problem nicht in **NP** liegt
- jedoch gilt:

### Satz

Das co-Graphenisomorphie-Problem ist in **IP**.

### Beweis:

Wir beschreiben dazu zwei Maschinen  $V$  und  $N^*$ :

13/50

## Interaktives Nachweissystem

### Verifizierer $V$

```
Input :  $G_1, G_2$ 
Antwort = ja
  wiederhole zweimal
    wähle  $i \in \{1, 2\}$  zufällig
    bilde eine isomorphe Kopie  $H$  von  $G_i$ 
    schicke  $H$  an  $N$ 
    empfangen  $j$ 
    falls  $i \neq j$ , setze Antwort = nein
  end
falls Antwort = ja, akzeptiere, andernfalls verwirf
end
```

### Nachweis $N^*$

```
Input  $G_1, G_2$ 
empfangen  $H$ 
falls  $H$  isomorph zu  $G_1$  ist, sende 1, andernfalls sende 2.
end
```

14/50

## Interaktives Nachweissystem

- offensichtlich ist die Laufzeit von  $V$  polynomiell beschränkt
- sind  $G_1$  und  $G_2$  nicht isomorph:
  - $V$  wird den Input  $(G_1, G_2)$  akzeptieren, da  $N^*$  korrekt antwortet
- sind  $G_1$  und  $G_2$  isomorph:
  - kein Nachweis  $N$  weiß, ob  $V$   $i = 1$  oder  $i = 2$  gewählt hat
  - er sendet ein zufälliges  $j$  aus  $\{1, 2\}$  zurück
- die Wahrscheinlichkeit, dass  $N$  richtig antwortet, ist somit  $\frac{1}{2}$ , bei zwei Versuchen  $\frac{1}{4}$ .  $\square$

15/50

## Interaktives Nachweissystem

### Gliederung

- interaktive Nachweise
- co-Graphenisomorphie
- **IP versus NP**
- **IP versus PSPACE**

16/50

## Interaktives Nachweissystem

- das co-Graphenisomorphie-Problem motiviert, dass  $\mathbf{NP} \not\subseteq \mathbf{IP}$ .
- offensichtlich liegt co-Graphenisomorphie in  $\mathbf{coNP}$
- wir zeigen im folgenden, dass allgemein  $\mathbf{coNP} \subseteq \mathbf{IP}$ .
- wir nutzen aus, dass  $\mathbf{IP}$  abgeschlossen ist unter polynomieller Reduktion ( $L \in \mathbf{IP}, L' \leq_p L \Rightarrow L' \in \mathbf{IP}$ ) (o. Beweis)
- weiter gilt (Kap. 4):  $L \mathbf{NP}$ -vollständig  $\iff \bar{L} \mathbf{coNP}$ -vollständig
- wir werden zeigen, dass das Komplement von 3-SAT in  $\mathbf{IP}$  liegt
- da co3SAT  $\mathbf{coNP}$ -vollständig ist, folgt  $\mathbf{coNP} \subseteq \mathbf{IP}$ .

17/50

## Interaktives Nachweissystem

- sei  $f = C_1 \wedge \dots \wedge C_m$  eine 3SAT-Formel mit Klauseln  $C_1, \dots, C_m$  und Literalen  $\{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$
- wir ordnen  $f$  induktiv ein Polynom  $A_f$  in den  $\{0, 1\}$ -Variablen  $z_i$  zu:

Literal $l_i = x_i$ :	$A_{l_i} = 1 - z_i$
Literal $l_i = \bar{x}_i$ :	$A_{l_i} = z_i$
Klausel $C_i = l_1 \vee l_2 \vee l_3$ :	$A_{C_i} = 1 - A_{l_1} A_{l_2} A_{l_3}$
KNF $f = C_1 \wedge \dots \wedge C_m$ :	$A_f = A_{C_1} \cdot A_{C_2} \cdot \dots \cdot A_{C_m}$

### Beispiel

$f = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$ . Dann ist

$$A_f = (1 - (1 - z_1)(1 - z_2)(1 - z_3))(1 - z_1 \cdot z_2(1 - z_4))$$

18/50

## Interaktives Nachweissystem

### Beobachtung

Ist  $f$  erfüllbar, so ist  $A_f \neq 0$ .

- denn: sei  $f$  erfüllbar durch eine Wahrheitsbelegung  $x^*$
- setze  $z_i = 1 \iff x_i^* = \text{wahr}$
- dann gibt es in jeder Klausel mindestens ein Literal  $l_i$ , das erfüllt ist:

$$\begin{aligned} l_i = x_i &\Rightarrow A_{l_i} = 1 - z_i = 0 \\ l_i = \bar{x}_i &\Rightarrow A_{l_i} = 0 \\ &\Rightarrow A_{C_i} = 1 - A_{l_1} A_{l_2} A_{l_3} = 1 \text{ für } i = 1, \dots, m \\ &\Rightarrow A_f = 1 \end{aligned}$$

19/50

## Interaktives Nachweissystem

- somit gilt:

$$f \text{ nicht erfüllbar} \iff \sum_{z_1=0}^1 \sum_{z_2=0}^1 \dots \sum_{z_n=0}^1 A_f(z_1, \dots, z_n) = 0$$

Die Idee des interaktiven Nachweises:

- $N$  muss zeigen, dass  $A_f \equiv 0$
- $N$  könnte die einzelnen Summanden aufschreiben
- dann müsste  $V$  aber  $2^n$  Summanden testen
- und hätte damit keine polynomielle Laufzeit

20/50

## Interaktives Nachweissystem

Ausweg:

- statt des Polynoms  $A_f$  betrachten wir eine Folge von Polynomen  $A_f^i$
- für  $i = 0, \dots, n$  sei  $A_f^i(z_1, \dots, z_i)$  jeweils das Polynom

$$A_f^i(z_1, \dots, z_i) = \sum_{z_{i+1}=0}^1 \cdots \sum_{z_n=0}^1 A_f(z_1, \dots, z_n)$$

Beispiel:

- $f = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$
- $A_f = (1 - (1 - z_1)(1 - z_2)(1 - z_3))(1 - z_1 \cdot z_2(1 - z_4))$
- $A_f^4(z_1, z_2, z_3, z_4) = A_f(z_1, z_2, z_3, z_4)$
- $A_f^3(z_1, z_2, z_3) = A_f(z_1, z_2, z_3, 0) + A_f(z_1, z_2, z_3, 1)$
- $A_f^2(z_1, z_2) = A_f(z_1, z_2, 0, 0) + A_f(z_1, z_2, 0, 1) + A_f(z_1, z_2, 1, 0) + A_f(z_1, z_2, 1, 1)$

21/50

## Interaktives Nachweissystem

- $f$  nicht erfüllbar  $\iff \sum_{z_1=0}^1 \sum_{z_2=0}^1 \cdots \sum_{z_n=0}^1 A_f(z_1, \dots, z_n) = 0$
- für alle  $z_1, \dots, z_i \in \mathbb{R}$  gilt:

$$\begin{aligned} A_f^i(z_1, \dots, z_i) &= \sum_{z_{i+1}=0}^1 \cdots \sum_{z_n=0}^1 A_f(z_1, \dots, z_n) \\ &= \sum_{z_{i+1}=0}^1 \left( \sum_{z_{i+2}=0}^1 \cdots \sum_{z_n=0}^1 A_f(z_1, \dots, z_n) \right) \\ &= A_f^{i+1}(z_1, \dots, z_i, 0) + A_f^{i+1}(z_1, \dots, z_i, 1) \end{aligned}$$

- somit:
  - $A_f^n = A_f$
  - $A_f^0 = 0 \iff f$  ist nicht erfüllbar
  - $A_f^i(z_1, \dots, z_{i-1}) = A_f^{i+1}(z_1, \dots, z_{i-1}, 0) + A_f^{i+1}(z_1, \dots, z_{i-1}, 1)$
  - $A_f$  ist vom Grad  $\leq 3m$ , wobei  $m$  die Anzahl der Klauseln ist

22/50

## Interaktives Nachweissystem

verbesserte Idee des interaktiven Nachweises:

- $N$  übermittelt die Koeffizienten der Polynome  $A_f^1, A_f^2, \dots, A_f^n$
- wie kann sich  $V$  davon überzeugen, dass  $N$  nicht betrügt?
  - er überprüft die Polynomidentität

$$A_f^{i-1}(z_1, \dots, z_{i-1}) = A_f^i(z_1, \dots, z_{i-1}, 0) + A_f^i(z_1, \dots, z_{i-1}, 1)$$

- wenn ein Test fehlschlägt, verwirft  $V$
- ansonsten akzeptiert  $V$

23/50

## Interaktives Nachweissystem

Wieder zwei Schwierigkeiten:

1. evtl. immer noch exponentiell viele Koeffizienten der  $A_f^i$
2. wie lässt sich testen, ob zwei Polynome übereinstimmen?

Ausweg:

1. statt der Polynome  $A_f^i(z_1, \dots, z_i)$  in  $i$  Variablen werden die Polynome  $p^i(z) = A_f^i(r_1, \dots, r_{i-1}, z)$  übertragen
  - $p^i(z)$  ist ein Polynom in einer Variable vom Grad  $\leq 3m$
  - $V$  wählt dazu sukzessiv Konstante  $r_1, \dots, r_{i-1}$
2.  $V$  prüft, ob die zwei Polynome an der Stelle  $r_1, \dots, r_i$  übereinstimmen

24/50

## Interaktives Nachweissystem

Der interaktive Nachweis geht wie folgt:

- $N$  wählt eine hinreichend große Zahl  $t$
- danach durchläuft das Verfahren  $n$  Runden
- in der  $i$ -ten Runde:
  - $V$  wählt eine Zahl  $r_i \leq t$
  - schickt  $V$  die Zahl  $r_i$  an  $N$
  - $N$  schickt die Koeffizienten eines Polynoms zurück
  - $V$  testet das Polynom
  - in Abhängigkeit vom Ausgang des Tests verwirft  $V$  oder beginnt eine neue Runde

25/50

## Interaktives Nachweissystem

- sei  $b_0 = 0$
- in der  $i$ -ten Runde: ( $1 \leq i \leq n$ )
  - $V$  hat bereits Zahlen  $r_1, \dots, r_{i-1}$  gewählt und die  $b_0, b_1, \dots, b_{i-1}$  berechnet
  - $N$  wird aufgefordert, die Koeffizienten des Polynoms

$$p_i(x) = A_i'(r_1, \dots, r_{i-1}, x)$$

in der Variable  $x \in \mathbb{R}$  zu schicken.

- $N$  darf betrügen und ein Polynom  $p_i'$  schicken
- $V$  prüft  $p_i'$ , indem er testet, ob gilt:

$$b_{i-1} = p_i'(0) + p_i'(1)$$

- falls nein, verwirft er
- falls ja, wählt  $V$  eine Zahl  $r_i \in \{0, 1, \dots, t-1\}$ , setzt  $b_i = p_i'(r_i)$  und iteriert
- am Ende der  $n$ -ten Runde prüft  $V$ , ob  $b_n = A_f(r_1, \dots, r_n)$
- falls auch dieser Test erfolgreich verläuft, akzeptiert  $V$ , andernfalls verwirft er.

26/50

## Interaktives Nachweissystem

Sei  $f$  nicht erfüllbar:

- dann kann  $N$  stets korrekt antworten mit

$$p_i'(x) = p_i(x) = A_i'(r_1, \dots, r_{i-1}, x)$$

- d.h.  $V$  wird mit Wahrscheinlichkeit 1 akzeptieren

Sei  $f$  erfüllbar:

- $N$  darf betrügen und ein falsches Polynom angeben
- er hat dazu aber immer weniger Freiheitsgrade zur Verfügung
- denn er muss stets sicherstellen:  $b_{i-1} = p_{i-1}'(r_{i-1}) = p_i'(0) + p_i'(1)$
- d.h. das von  $N$  gewählte Polynom  $p_i'$  hängt von  $p_{i-1}'$  und von  $r_{i-1}$  ab

27/50

## Interaktives Nachweissystem

- bezeichne  $\Pr(p_i \neq p_i')$  die Wahrscheinlichkeit dafür, dass
  - $N$  im  $i$ -ten Schritt mit einem falschen Polynom erwischt wird,
  - bis dahin aber alle Tests positiv verlaufen sind
- per Induktion über  $i$  lässt sich zeigen:

$$\Pr(p_i \neq p_i') \geq \left(1 - \frac{3m}{t}\right)^i$$

- dann ist die Wahrscheinlichkeit, dass  $N$  spätestens im letzten Schritt erappt wird, mindestens

$$\left(1 - \frac{3m}{t}\right)^n > \frac{2}{3} \quad \text{für } t \text{ hinreichend groß}$$

28/50

## Interaktives Nachweissystem

Damit folgt:

### Satz

$\text{coNP} \subseteq \text{IP}$ .

□

Weiter gilt:

### Korollar

$\text{NP} \cup \text{coNP} \subseteq \text{IP}$

□

29/50

## Interaktives Nachweissystem

### Gliederung

- interaktive Nachweise
- co-Graphenisomorphie
- IP versus NP
- IP versus PSPACE

30/50

## Interaktives Nachweissystem

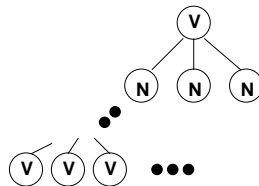
Wir zeigen weiter, dass **IP** nicht größer werden kann als **PSPACE**.

### Satz

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

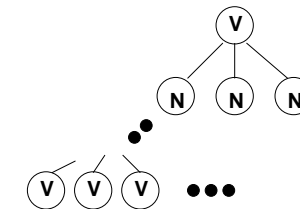
### Beweisidee:

- ähnlich wie der Beweis zu  $\text{NP} \subseteq \text{PSPACE}$
- die Gesamtlänge der Nachrichten, die zwischen  $V$  und  $N$  ausgetauscht werden, ist durch ein Polynom  $p$  beschränkt
- zu gegebenem Input  $x$  stellen wir einen Baum auf, in dem alle möglichen Kommunikationen der Länge  $p(x)$  zwischen  $N$  und  $V$  dargestellt sind



31/50

## Interaktives Nachweissystem



- entlang eines Pfades von der Wurzel zu einem Blatt alternieren  $V$ - und  $N$ -Knoten
- jeder Pfad entspricht einer Kommunikation der Länge  $p(x)$
- die Verzweigungen über einem  $V$ -Knoten entsprechen dem probabilistischen Verhalten der **BPP**

32/50

## Interaktives Nachweissystem

- Wir werten den Baum mit einer **PSPACE**-Maschine wie folgt aus:
  - führe Tiefensuche durch
  - ein Blatt erhält den Wert
$$\begin{cases} 1, & \text{falls es ein akzeptierender Endknoten ist;} \\ 0, & \text{sonst} \end{cases}$$
  - ein  $N$ -Knoten erhält den Maximalwert seiner Söhne (der dann dem Verhalten von  $N^*$  entspricht),
  - ein  $V$ -Knoten erhält den Mittelwert seiner Söhne
- dann ist  $x \in L \iff$  der Wert am Wurzelknoten ist größer als  $\frac{2}{3}$
- Tiefensuche benötigt polynomiell viel Platz
- die  $N$ -Maschine muss nicht simuliert werden,
- das Verhalten von  $V$  hängt nur von der Nachricht abhangt und nicht davon, wie die Nachricht erzeugt wurde.  $\square$

33/50

## Interaktives Nachweissystem

- wir wollen als nachstes zeigen, dass **IP** = **PSPACE**
- wir betrachten dazu das folgende Problem:

### QBF (quantified Boolean formula)

- sei  $f(x_1, \dots, x_n)$  eine Boolesche Formel in konjunktiver Normalform
- sei  $F = (Q_1, x_1)(Q_2, x_2) \dots (Q_n, x_n)f(x_1, \dots, x_n)$  mit Quantoren  $Q_i \in \{\forall, \exists\}$
- Frage: ist  $F$  erfullbar?

### Beispiel:

$\forall x_1 \exists x_2$ , so dass  $\forall x_3$  und  $\forall x_4$  gilt:  $f = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$ .

34/50

## Interaktives Nachweissystem

Ohne Beweis setzen wir den folgenden Satz voraus:

### Satz

QBF ist **PSPACE**-vollstandig.  $\square$

- wir werden zeigen, dass  $\text{QBF} \in \text{IP}$  und damit **PSPACE**  $\subseteq$  **IP**
- wir werden wie im Beweis zum vorigen Satz vorgehen und  $F$  ein Polynom zuordnen
- wir wissen bereits, wie wir einer KNF ein Polynom zuordnen
- wir mussen uns also etwas fur die Quantoren uberlegen

35/50

## Interaktives Nachweissystem

- sei  $p(x_1, \dots, x_n)$  ein Polynom in  $n$  Variablen
- dem Ausdruck  $\forall x_1 p(x_1, \dots, x_n)$  konnen wir wie folgt ein Polynom in  $n - 1$  Variablen zuordnen

$$\text{AND}_{x_1}^p(x_2, \dots, x_n) = p(0, x_2, \dots, x_n)p(1, x_2, \dots, x_n)$$

- ist  $p$  ein binares Polynom, so ist auch  $\text{AND}_{x_1}^p$  binar
- uber Booleschen Variablen gilt:

$$\text{AND}_{x_1}^p(x_2, \dots, x_n) = 1 \iff p(0, x_2, \dots, x_n) = p(1, x_2, \dots, x_n) = 1$$

36/50

## Interaktives Nachweissystem

- entsprechend:
- dem Ausdruck  $\exists x_1 p(x_1, \dots, x_n)$  können wir wie folgt ein Polynom in  $n - 1$  Variablen zuordnen

$$\text{OR}_{x_1}^p(x_2, \dots, x_n) = p(0, x_2, \dots, x_n) + p(1, x_2, \dots, x_n) - \text{AND}_{x_1}^p(x_2, \dots, x_n)$$

- ist  $p$  ein binäres Polynom, so ist auch  $\text{OR}_{x_1}^p$  binär
- über Booleschen Variablen gilt:  
$$\text{OR}_{x_1}^p(x_2, \dots, x_n) = 1 \iff p(0, x_2, \dots, x_n) = 1 \text{ oder } p(1, x_2, \dots, x_n) = 1$$

37/50

## Interaktives Nachweissystem

$$\text{AND}_{x_1}^p(x_2, \dots, x_n) = p(0, x_2, \dots, x_n)p(1, x_2, \dots, x_n)$$

$$\text{OR}_{x_1}^p(x_2, \dots, x_n) = p(0, x_2, \dots, x_n) + p(1, x_2, \dots, x_n) - \text{AND}_{x_1}^p(x_2, \dots, x_n)$$

- sei  $F = (Q_1, x_1)(Q_2, x_2) \dots (Q_n, x_n) p(x_1, \dots, x_n)$  eine QBF
- dann gilt

$$F \text{ erfüllbar} \iff (Q_1, x_1)(Q_2, x_2) \dots (Q_{n-1}, x_{n-1}) F' \text{ erfüllbar,}$$

wobei

$$F' = \begin{cases} \text{AND}_{x_n}^p(x_1, \dots, x_{n-1}), & \text{falls } Q_n = \forall \\ \text{OR}_{x_n}^p(x_1, \dots, x_{n-1}), & \text{falls } Q_n = \exists \end{cases}$$

- per Induktion folgt:  $F$  erfüllbar  $\iff$  das resultierende konstante Polynom  $F'$  ist nicht Null

38/50

## Interaktives Nachweissystem

- Nachteil dieses Vorgehens:
  - durch wiederholte Anwendung von AND und OR kann der Grad des Polynoms exponentiell anwachsen
  - die Auswertung von  $F'$  kann aufwendig sein, obwohl  $F$  einfach auszuwerten war, da in konjunktiver Normalform
- damit kann keine polynomielle Laufzeitschranke gewährleistet werden
- um dem Anwachsen der Grade entgegenzuwirken, definieren wir einen dritten Typ von Polynomen

39/50

## Interaktives Nachweissystem

Sei  $p(x_1, \dots, x_n)$  ein Polynom in  $n$  Variablen

$$\text{RED}_{x_1}^p(x_1, \dots, x_n) = p(0, x_2, \dots, x_n) + (p(1, x_2, \dots, x_n) - p(0, x_2, \dots, x_n)) \cdot x_1$$

offensichtlich gilt für  $x_2, \dots, x_n \in \mathbb{R}$ :

1.  $\text{RED}_{x_1}^p(0, x_2 \dots x_n) = p(0, x_2 \dots x_n)$
2.  $\text{RED}_{x_1}^p(1, x_2 \dots x_n) = p(1, x_2 \dots x_n)$
3.  $\text{RED}_{x_1}^p$  reduziert den Grad von  $x_1$  auf 1
4. ist  $p$  binär, so auch  $\text{RED}_{x_1}^p$

40/50

## Interaktives Nachweissystem

### Beispiel

- sei  $f(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$
- dann gilt:

$$A_f(x_1, \dots, x_4) = \left(1 - (1 - x_1)(1 - x_2)(1 - x_3)\right) \left(1 - x_1 x_2 (1 - x_4)\right)$$

$$\text{AND}_{x_1}(x_2 \dots x_4) = \left(1 - (1 - x_2)(1 - x_3)\right) \left(1 - x_2(1 - x_4)\right)$$

$$\text{OR}_{x_1}(x_2 \dots x_4) = 1 - (1 - x_2)(1 - x_3) + \left(1 - x_2(1 - x_4)\right) - \text{AND}_{x_1}$$

$$\text{RED}_{x_1}(x_1, \dots, x_4) = \left(1 - (1 - x_4)(1 - x_3)\right) + \left(x_2(1 - x_4) + (1 - x_2)(1 - x_3)\right) x_1$$

41/50

## Interaktives Nachweissystem

Für ein Polynom  $p(x_1, \dots, x_n)$  in  $n$  Variablen gilt somit:

- dem Ausdruck  $\forall x_1 p(x_1, \dots, x_n)$  entspricht das Polynom  $\text{AND}_{x_1}^p$
- dem Ausdruck  $\exists x_1 p(x_1, \dots, x_n)$  entspricht das Polynom  $\text{OR}_{x_1}^p$
- ist  $p$  binär, so sind auch AND, OR und RED binär
- $\text{RED}_{x_1}^p(0, x_2 \dots x_n) = p(0, x_2 \dots x_n)$
- $\text{RED}_{x_1}^p(1, x_2 \dots x_n) = p(1, x_2 \dots x_n)$
- der Grad von  $x_1$  in AND und OR ist Null
- der Grad von  $x_i$  in AND und OR kann sich verdoppeln
- der Grad von  $x_1$  in RED ist Eins
- der Grad von  $x_i$  in RED bleibt unverändert

42/50

## Interaktives Nachweissystem

- beim Nullstellenproblem für Polynome haben wir gesehen, dass es aufwendig sein kann,  $p$  an einer bestimmten Stelle auszuwerten
- der Aufwand hängt davon ab, wie das Polynom gegeben ist
- wir betrachten die allgemeine Fragestellung:

### Polynomtest zu einem gegebenen Polynom $p$ ( $p$ -Test)

- gegeben  $a_1, \dots, a_n$  und  $b$
- ist  $p(a_1, \dots, a_n) = b$  ?

43/50

## Interaktives Nachweissystem

Ohne Beweis verwenden wir die folgenden Aussagen:

- Ist  $p\text{-TEST} \in \mathbf{IP}$  für ein Polynom  $p$ , dann sind auch die Tests für die Polynome  $\text{AND}_{x_1}^p$ ,  $\text{OR}_{x_1}^p$  und  $\text{RED}_{x_1}^p$  in  $\mathbf{IP}$ .
- Ist  $f$  eine Formel in KNF und  $A_f$  das zugehörige Polynom, so ist der  $p$ -TEST für  $A_f$  in  $\mathbf{IP}$ .

44/50

## Interaktives Nachweissystem

### Satz

**IP = PSPACE.**

### Beweis:

- sei  $F = (Q_1, x_1) \dots (Q_n, x_n) f(x_1 \dots x_n)$  eine prädikatenlogische Formel
- sei wie vorher  $A_f$  das zu  $f$  gehörende Polynom
- wir wissen:  $A_f$ , AND, OR und RED haben einen  $p$ -Test in **IP**
- wir modifizieren  $A_f$  wie folgt:

45/50

## Interaktives Nachweissystem

**Phase 1** wende nacheinander  $RED_{x_1}, \dots, RED_{x_n}$  an

**Phase 2** do  $i = n$  to 1

ist  $Q_i = \forall$ : wende  $AND_{x_i}$  an

ist  $Q_i = \exists$ : wende  $OR_{x_i}$  an

do  $j = 1$  to  $i - 1$

wende  $RED_{x_j}$  an

end do

end do

- Phase 1 reduziert  $A_f$  auf ein Polynom von Grad 1
- in Phase 2 kann der Grad auf höchstens 2 anwachsen, wird danach aber wieder reduziert
- am Ende erhalten wir ein konstantes Polynom  $p$  mit  $p = 0$  oder  $p = 1$
- mit dem **IP**-Test können wir entscheiden, ob  $p = 0$  oder  $p = 1$ .  $\square$

46/50

## Interaktives Nachweissystem

### Veranschaulichung:

- Nachweise in **PSPACE** können exponentielle Länge haben
- lediglich ihre „Breite“ ist klein
- wenn wir einen Nachweis als Folge von äquivalenten Umformungen auffassen, also
  - $a_1 \Leftrightarrow a_2, a_2 \Leftrightarrow a_3, \dots, a_{k-1} \Leftrightarrow a_k$
  - dann kann  $k$  exponentiell groß sein
  - aber der Platzbedarf für eine Äquivalenz ist polynomiell beschränkt
- der Satz besagt, dass solche langen Nachweise trotzdem in polynomieller Zeit randomisiert überprüft werden können

47/50

## Interaktives Nachweissystem

- abschließend eine kurze Bemerkung zu dem Spezialfall von zero-knowledge-Nachweisen
- wir verzichten auf eine formale Definition von „zero knowledge“
- informell: **IP**-Protokolle, bei denen  $V$  nichts über den Nachweis lernt
- solche Protokolle können u.a. zur Zugangskontrolle eingesetzt werden
- das Wissen von  $N$  dient als Nachweis seiner Identität
- $V$  kann sich von diesem Wissen überzeugen können,
- ohne dass er in der Lage wäre, sich danach als  $N$  auszugeben
- es gilt: jedes Problem in **NP** hat einen zero-knowledge-Nachweis
- wir illustrieren den Ansatz an der Graphenisomorphie

48/50

# Interaktives Nachweissystem

## Graphenisomorphie

Input  $G_1, G_2$   
wiederhole zweimal

$N$ : wählt zufällig  $i \in \{1, 2\}$  sowie eine Permutation  $\pi$   
bildet die isomorphe Kopie  $H = G_i(\pi)$   
sendet  $H$  an  $V$

$V$ : wählt  $j \in \{1, 2\}$  zufällig  
sendet  $j$  an  $N$

$N$ : falls  $G_1 \sim G_2$ :  
bestimmt Permutation  $\sigma$ , so dass  $G_j(\sigma) = G_i(\pi)$   
sendet  $\sigma$  an  $V$   
falls  $G_1 \not\sim G_2$ :  
sendet  $\sigma = \pi$  an  $V$

$V$ : akzeptiert  $\iff G_j(\sigma) = G_i(\pi)$   
end

# Interaktives Nachweissystem

- das Verfahren ist ein **IP-Protokoll**:
- falls  $G_1 \sim G_2$ :
  - es existiert immer ein  $\sigma$  mit  $G_j(\sigma) = G_i(\pi)$
  - $V$  akzeptiert dann mit Wahrscheinlichkeit 1
- falls  $G_1 \not\sim G_2$ :
  - $\sigma$  existiert nur, wenn  $i = j$
  - die Wahrscheinlichkeit dafür ist  $\frac{1}{2}$
  - bei zwei Versuchen  $\frac{1}{4}$
- das Verfahren ist ein **zero-knowledge-Protokoll**:
- $V$  erfährt nichts über den Isomorphismus zwischen  $G_1$  und  $G_2$  (wenn er existiert)