Informatik III

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer

Wintersemester 2006/07

22. Vorlesung

19.01.2007

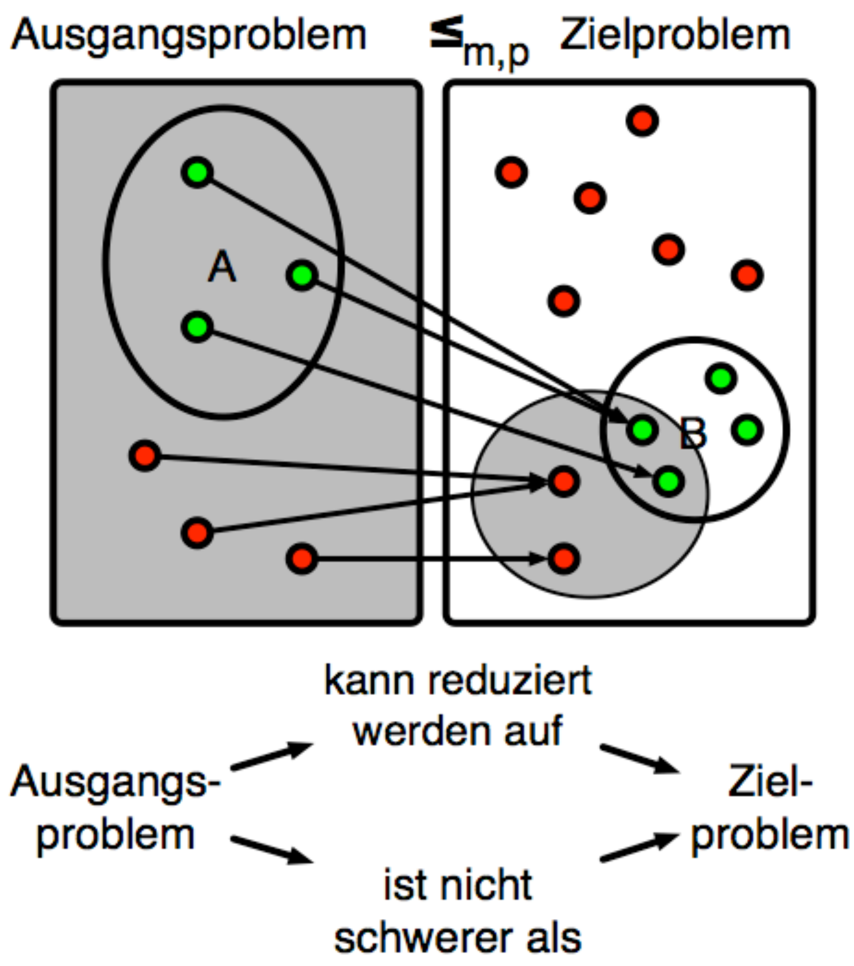
Komplexitätstheorie - Zeitklassen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Die Komplexitätsklassen TIME**
  - DTIME, NTIME
  - P
  - NP
- **Das Cook-Levin-Theorem**
  - Polynomial-Zeit-Reduktion
  - Reduktion von 3SAT auf Clique
  - NP-vollständigkeit
  - SAT ist NP-vollständig
- **Weitere NP-vollständige Probleme**
  - Knotenüberdeckung (Vertex-Cover)
  - Das Hamiltonsche Pfadproblem
  - Das ungerichtete Hamiltonsche Pfadproblem
  - Das Teilsummenproblem

Die Polynom-Zeit-Abbildungsreduktion

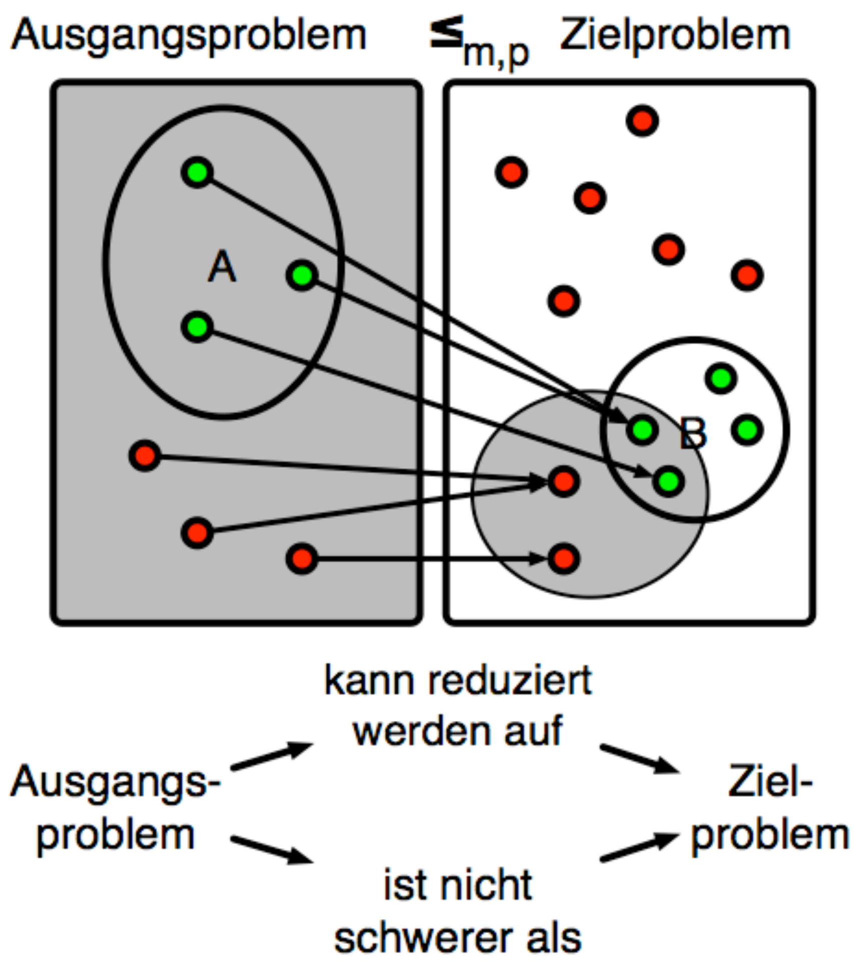
- **Definition (Abbildungsreduktion, Polynomial Time Mapping Reduction, Many-one)**
  - Eine Sprache A kann durch Abbildung auf eine Sprache B in Polynom-Zeit reduziert werden: **A $\leq_{m,p}$ B,**
    - falls es eine in Polynom-Zeit berechenbare Funktion $f: \Sigma^* \rightarrow \Sigma^*$ gibt,
    - so dass für alle w: $w \in A \Leftrightarrow f(w) \in B$
  - Die Funktion f heißt die **Reduktion** von A auf B.



Polynom-Zeit-Abbildungsreduktion, P & NP

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Theorem**
  - Falls A $\leq_{m,p}$ B und B ist in P, dann ist A auch in P.
- **Theorem**
  - Falls A $\leq_{m,p}$ B und B ist in NP, dann ist A auch in NP.



Besondere Boolesche Funktionen

- **Literal:**
  - ist eine einfache oder negierte Variable
  - z.B.: x, y, z, $\neg x$, $\neg y$, $\neg z$
- **Klausel:**
  - ist eine Disjunktion von Literalen
  - z.B.: $x \vee y$, $z \vee \neg x \vee \neg y$, $x \vee \neg z$
- **Konjunktive Normalform (CNF)**
  - Konjunktion von Klauseln
  - z.B.: $(x \vee y) \wedge (z \vee \neg x \vee \neg y) \wedge (x \vee \neg z)$
- **k-konjunktive Normalform (k-CNF)**
  - Konjunktion von Klauseln aus k Literalen, z.B. 3-CNF
    - $(x \vee y \vee y) \wedge (z \vee \neg x \vee \neg y) \wedge (x \vee \neg z \vee \neg z)$
- **Disjunktive Normalform (DNF)**
  - Disjunktion aus Konjunktion von Literalen
    - $(x \wedge y) \vee (z \wedge \neg x \wedge \neg y) \vee (x \wedge \neg z)$

Das Erfüllbarkeitsproblem Boolescher Funktionen

- **Eine Boolesche Funktion $f(x_1,x_2,..,x_n)$ ist erfüllbar, wenn es eine Wertebelegung für $x_1,x_2,..,x_n$ gibt, so dass $f(x_1,x_2,..,x_n) = 1$**
  - $(x \vee y) \wedge (z \vee \neg x \vee \neg y) \wedge (x \vee \neg z)$ ist erfüllbar, da
    - die Belegung $x = 1$, $y = 0$, $z = 0$
  - $(1 \vee 0) \wedge (0 \vee 0 \vee 1) \wedge (1 \vee 1) = 1 \wedge 1 \wedge 1 = 1$ liefert.
- **Definition (Satisfiability Problem of Boolean Formulas)**
  - Das Erfüllbarkeitsproblem der Booleschen Funktion ist definiert als:
  - **SAT = { ϕ | ϕ ist eine erfüllbare Funktion}**, d.h.
  - **Gegeben**:
    - Boolesche Funktion ϕ
  - **Gesucht**:
    - Gibt es $x_1,x_2,..,x_n$ so dass $\phi(x_1,x_2,..,x_n) = 1$
- **Spezialfall: 3-SAT**
  - **3-SAT = { ϕ | ϕ ist eine erfüllbare Funktion in 3-CNF}**
  - **z.B.:**

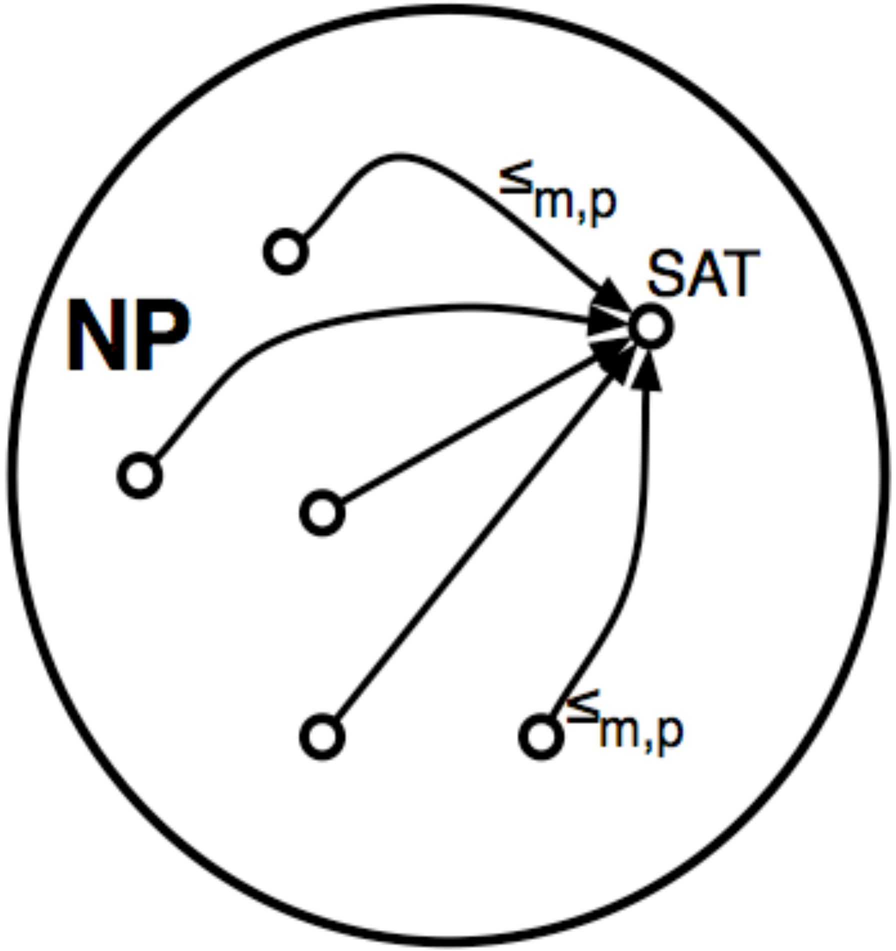
$$\psi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

Der Satz von Cook und Levin

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Definition:**
  - Eine Sprache S ist NP-vollständig (NP-complete) wenn:
    - $S \in NP$
    - S ist NP-schwierig
      - d.h. für alle $L \in NP$: $L \leq_{m,p} S$

- **Theorem (Cook/Levin)**
  - SAT ist NP-vollständig



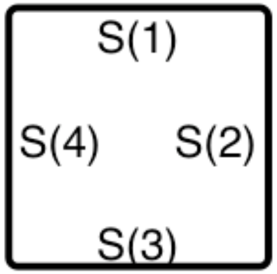
Beweis-Strategie

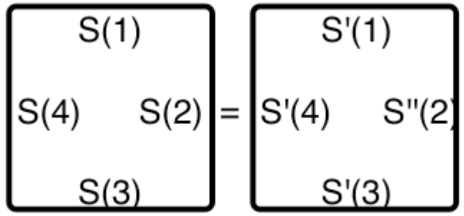
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

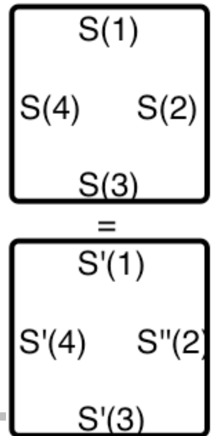
- **Definition:**
  - Eine Sprache S ist NP-vollständig (NP-complete) wenn:
    - $S \in NP$
    - S ist NP-schwierig
      - d.h. für alle $L \in NP$: $L \leq_{m,p} S$
- **Theorem (Cook/Levin)**
  - SAT ist NP-vollständig
- **Beweis-Idee:**
  - Sei $L \in NP$. Wir beweisen folgende Reduktionen:
    1. Es gibt ein k, so dass $L \leq_{m,p} A_{NTIME(n^k)}$
    2. Für alle k: $A_{NTIME(n^k)} \leq_{m,p} A_{NTIME(n)}$
    3. $A_{NTIME(n)} \leq_{m,p}$ PARKETT
    4. PARKETT $\leq_{m,p}$ FinPred
    5. FinPred $\leq_{m,p}$ SAT
  - Daraus folgt für alle $L \in NP$: $L \leq_{m,p} S$
  - Dann zeigen wir $SAT \in NP$

Das Parkett-Problem

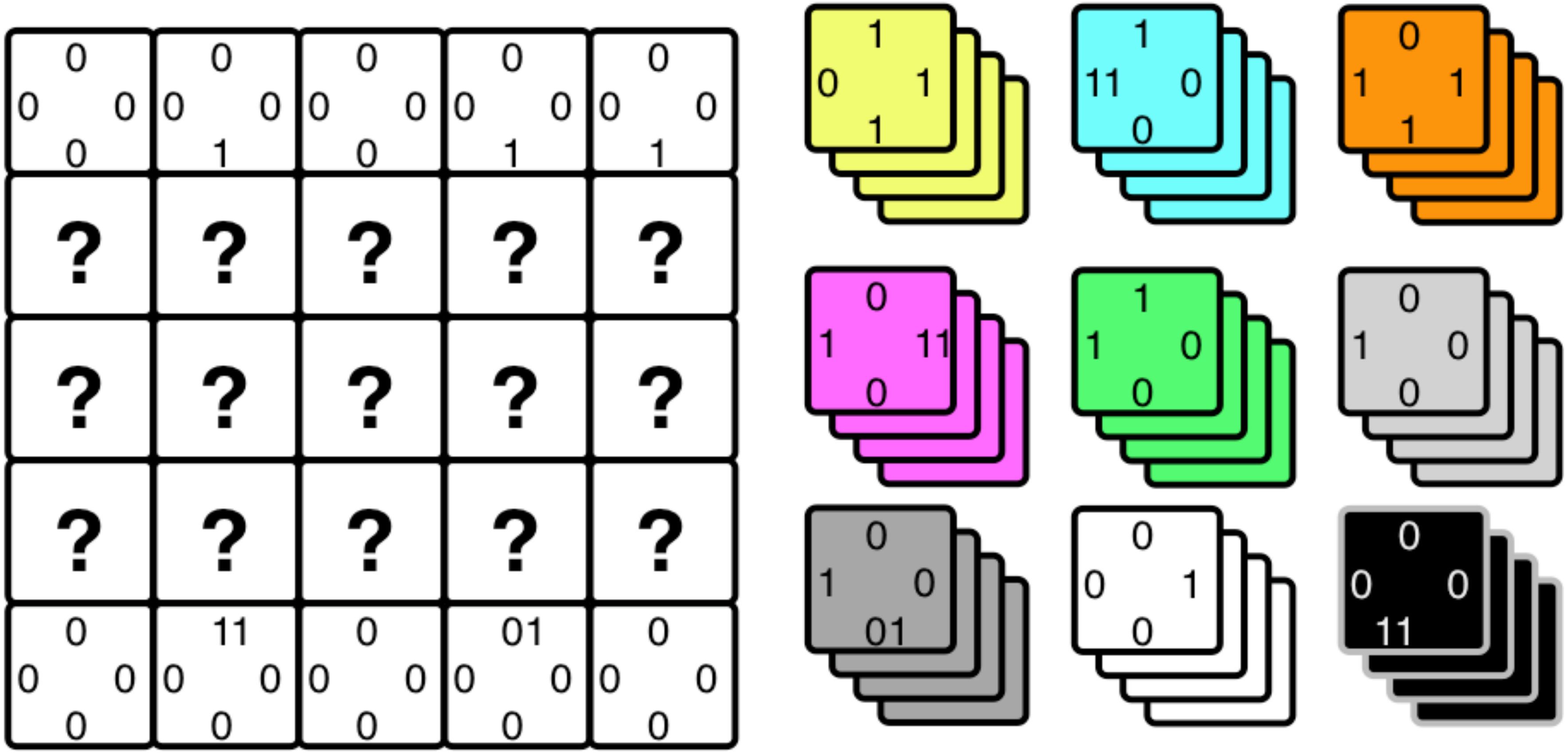
- **Definition**
  - Das Parkettproblem ist das Problem, eine gegebene quadratische Fläche mit Parkettstücken verschiedener Form lückenlos abzudecken
  - Auch bekannt als zweidimensionales Domino-Problem, also:
    - Parkett-Teil S: 4-Tupel (S(1), S(2), S(3), S(4)) mit $S(i) \in \{0,1\}^*$
    - H(S,S') := S und S' passen horizontal, d.h. S(2)=S'(4)
    - V(S,S') := S und S' passen vertikal, d.h. S(3)=S'(1)
  - Gegeben:
    - Menge von Bauteilen $M = \{B_1, B_2, B_3, \ldots, B_k\}$
    - Oberer Rand $S_{1,1}, \ldots, S_{1,m}$
    - Unterer Rand $S_{m,1}, \ldots, S_{m,m}$
  - Gesucht:
    - Gibt es eine Menge von Bauteilen $S_{i,j}$, für $i \in \{1,..,m\}$, $j \in \{2,..,m-1\}$ die passen, d.h.
      - $H(S_{i,j}, S_{i,j+1})$ für alle $i \in \{1,..,m\}$, $j \in \{1,..,m-1\}$ und
      - $V(S_{i,j}, S_{i+1,j})$ für alle $i \in \{1,..,m-1\}$, $j \in \{1,..,m\}$.







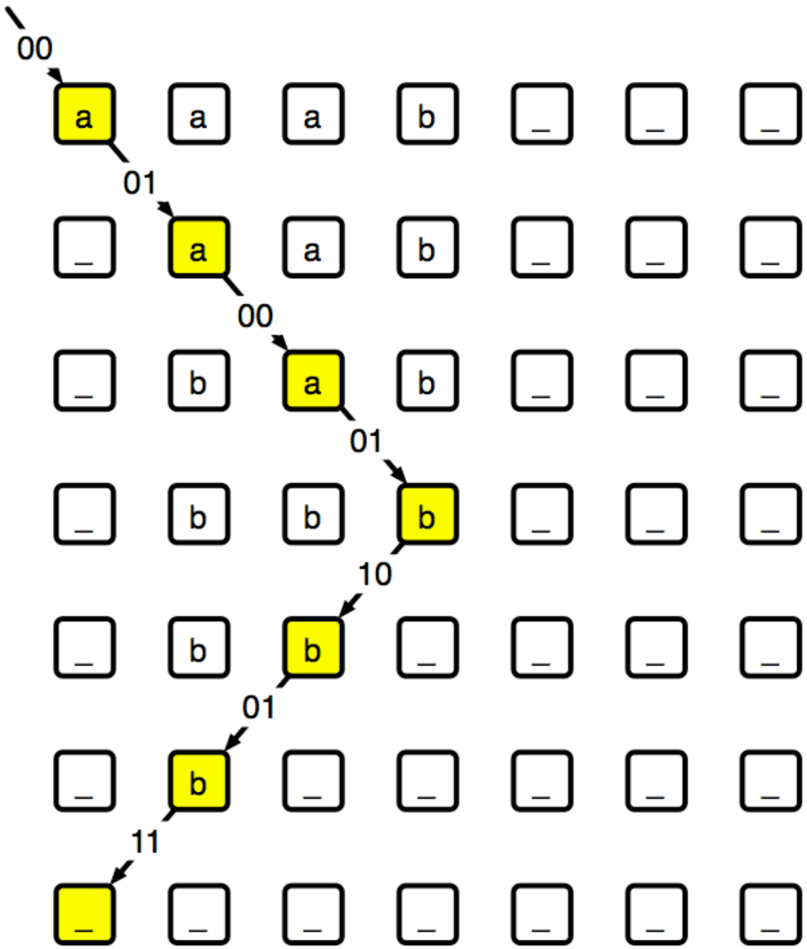
Das Parkett-Problem



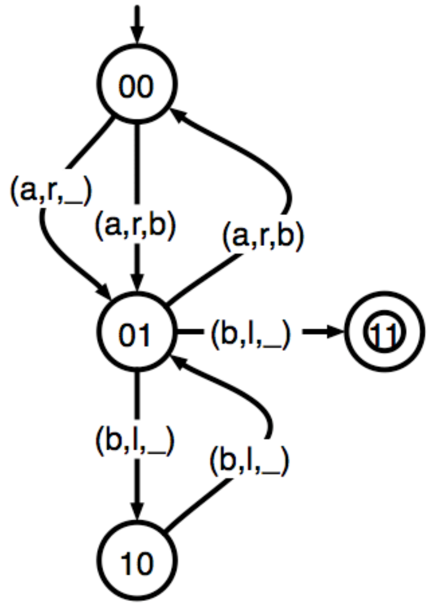
$A_{NTIME(n)} \leq_{m,p}$ PARKETT

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

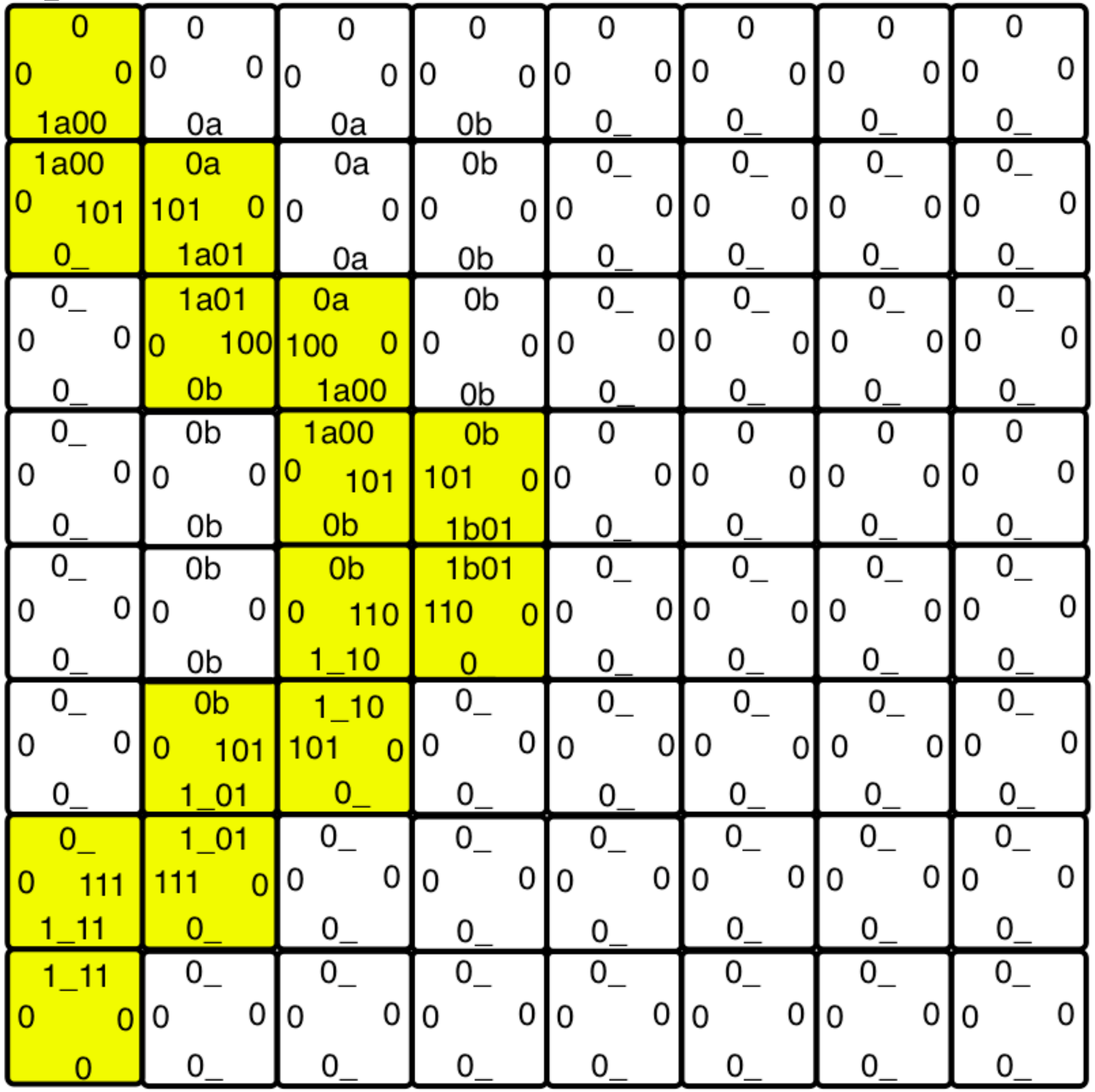
➢ **Betrachte NTM M**

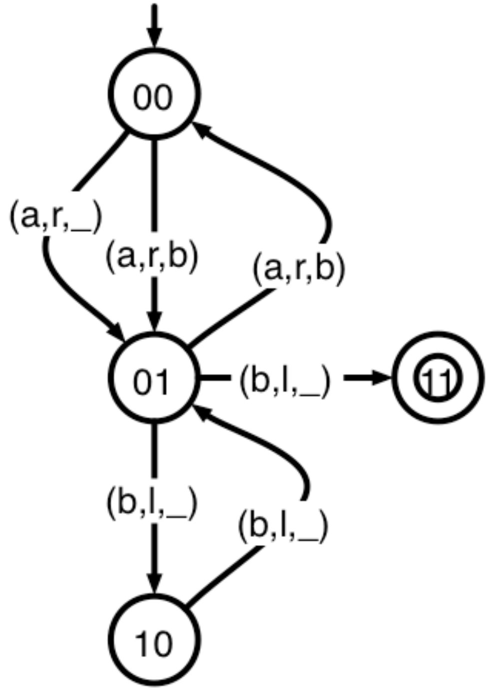


Akzeptiert auf leerer Eingabe



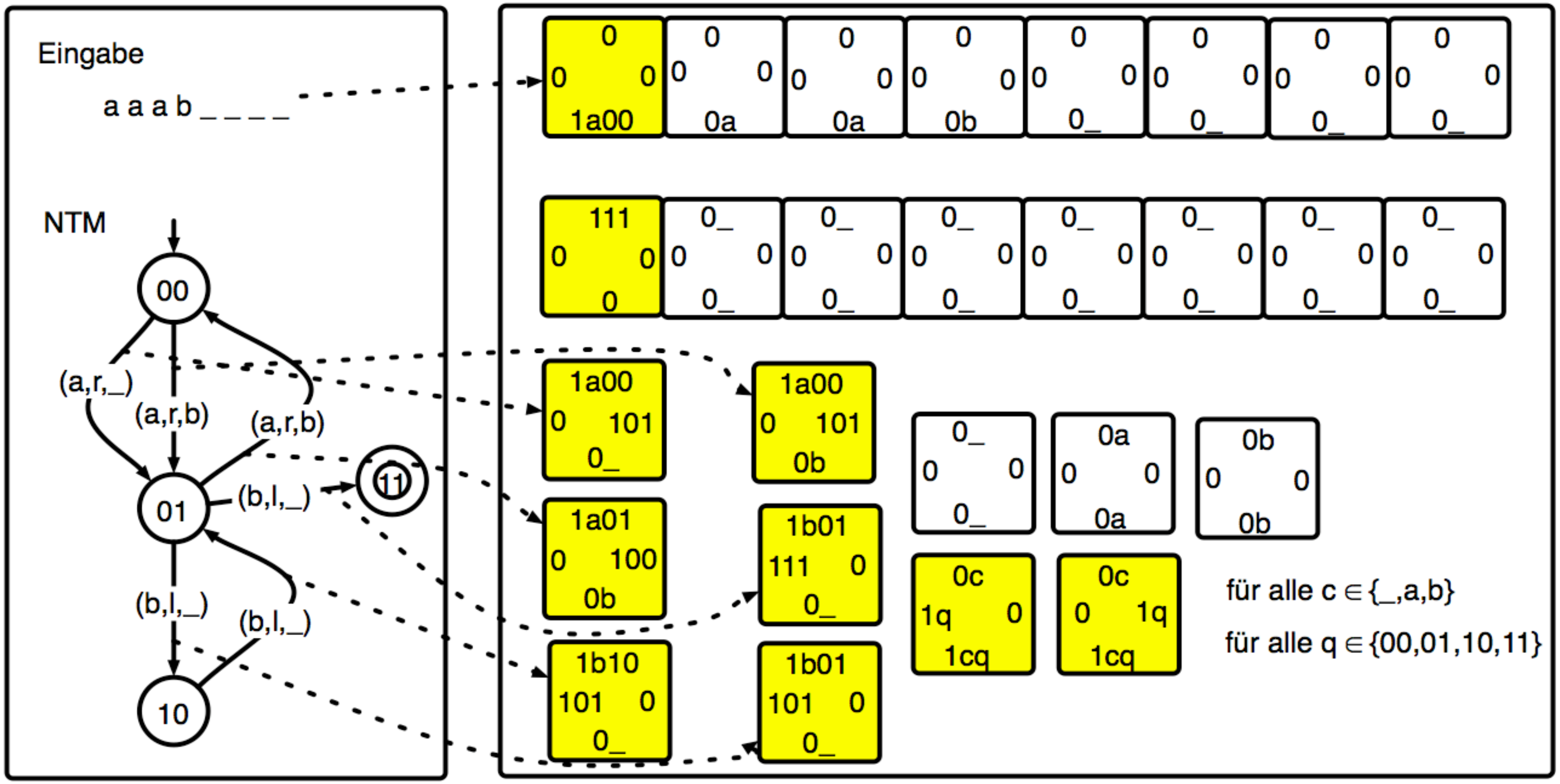
- die nach der Berechnung das gesamte Band mit _ beschreibt
- mit eindeutigen akzeptierenden Zustand q_{akz}
- und die am linken Rand der Eingabe akzeptiert





Parkettierung gelungen

Beispielreduktion



$A_{\text{NTIME(n)}} \leq_{m,p}$ PARKETT

- **Die Reduktionsfunktion:**
  - Auf Eingabe 1-Band NTM M mit Zustandsmenge Q, Startzustand q_0 und akzeptierenden Zustand q_{akz} welche in Linearzeit rechnet und die Eingabe löscht
  - und Eingabe $x \in \Sigma^*$
- **1. Erzeuge oberen und unteren Rand durch die Teile**
  - $S_{1,1}$ = (0, 0, 1 bin(x_1) bin(q_0), 0), $S_{1,i}$ = (0, 0, 0 bin(x_i), 0) für $i \geq 2$
  - $S_{m,1}$ = (1 bin("_") bin(q_{akz}) 0, 0, 0, 0), $S_{m,i}$ = (0 bin("_"), 0, 0 , 0) für $i \geq 2$
- **2. Erzeuge die folgenden Parkett-Teile**
  - Standardzellen: (0z,0,0z,0) für alle $z \in \Sigma$
  - Schiebezellen:
    - (0 bin(z), 0, 1 bin(z) bin(q), 1 bin(q)) für alle $z \in \Sigma$, $q \in Q$
    - (0 bin(z), 1 bin(q), 1 bin(z) bin(q), 0) für alle $z \in \Sigma$, $q \in Q$
  - Rechenzellen: Falls $(q,z) \rightarrow (q',z,l)$ in der Übergangsfunktion von M:
    - (1 bin(z) bin(q) , 0, 0 bin(z'), 1 bin(q'))
  - Falls $(q,z) \rightarrow (q',z,r)$ in der Übergangsfunktion von M
    - (1 bin(z) bin(q) , 1 bin(q'), 0 bin(z'), 0)
- **Laufzeit: O(n)** (Reduktion besteht hauptsächlich aus Kopieren)
- **Korrektheit:**
  - Jede akzeptierende Berechnung ergibt eine vollständige Parkettierung
  - Jede vollständige Parkettierung kann einer akzeptierenden Berechnung zugeordnet werden

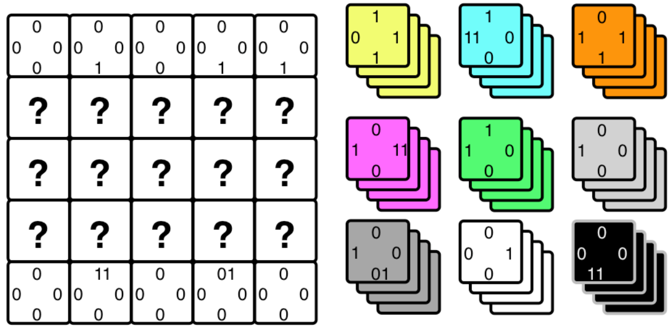
PARKETT $\leq_{m,p}$ FinPred

- **Endliches Prädikat**
  - Ist eine Funktion $P(x_1,\ldots,x_k)$ die für Variablen $x_1,\ldots,x_k \in \{0,1\}^p$ die Wert 0 (falsch) oder 1 (wahr) liefert
  - Wir beschränken uns auf die folgenden Prädikate auf Konstanten oder Variablen
    - **`Element`**: Für $x,y_1,\ldots,y_k \in \{0,1\}^p$: für Konstanten $y_1,\ldots,y_k$ und Variable x
      - **`Element`**$(x, y_1,\ldots,y_k) := 1$ gdw $x \in \{y_1,\ldots,y_k\}$
    - Teile sind gleich: Für Variablen $x,y \in \{0,1\}^p$ mit $x= x_1,\ldots,x_p$ und $y= y_1,\ldots,y_p$
      - **`Teilgleich`**$_{a,b,c}(x,y)$
        $= (x_a = y_b) \wedge (x_{a+1} = y_{b+1}) \wedge \ldots \wedge (x_{b+c-1} = y_{b+c-1})$
- **Definition FinPred (Endliche Prädikate)**
  - Gegeben:
    - Eine Menge von endlichen Prädikaten
  - Gesucht:
    - Gibt es eine Belegung der Variablen $x_1,\ldots,x_k \in \{0,1\}^p$ so dass alle Prädikate erfüllt werden
- **Lemma**
  - PARKETT $\leq_{m,p}$ FinPred

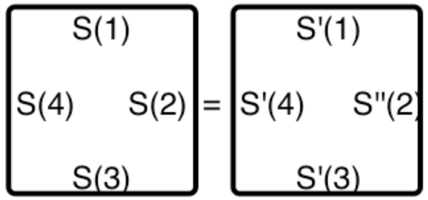
- **Beispiel:**
  - **`Element`**(x,010,100,111)
  - **`Element`**(y,110,000)
  - **`Teilgleich`**$_{2,3,1}$ (x,y)
  - **`Element`**(z,110,011,101)
  - **`Teilgleich`**$_{1,2,2}$ (x,z)
- **Lösung:**
  - x = 100
  - y = 110
  - z = 110

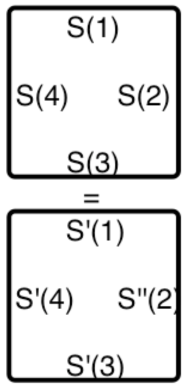
  - Teilgleich$_{2,3,1}$ (x,y)
    - x= 100, y=110
  - Teilgleich$_{1,2,2}$ (x,z)
    - x= 100, z=110

PARKETT $\leq_{m,p}$ FinPred



- **Lemma**
  - PARKETT ≤m,p FinPred
- **Reduktion**
  - PARKETT ist gegeben als
    - Menge von Bauteilen $M = \{B_1, B_2, B_3, \dots, B_k\}$
    - Ränder $S_{1,1}, \dots, S_{1,m}$ und $S_{m,1}, \dots, S_{m,m}$
  - Verwende für alle Ränder Zeichenketten gleicher Länge p
  - Kodiere ein Bauteil S als Zeichenkette S= S(1)S(2)S(3)S(4)
  - Seien $V_{i,j}$ Variablen aus $\{0,1\}^{4p}$
  - Füge Prädikate ein:
    - `Element`$(V_{1,i}, S_{1,i})$ für alle $i \in \{1,..,m\}$
    - `Element`$(V_{m,i}, S_{m,i})$ für alle $i \in \{1,..,m\}$
    - `Element`$(V_{i,i}, B_1, B_2, B_3, \dots, B_k)$ für alle $i \in \{2,..,m-1\}$, $j \in \{1,..,m\}$
    - `Teilgleich`$_{p+1,3p+1,p}(V_{i,i}, V_{i,i+1})$ für alle $i \in \{1,..,m\}$, $j \in \{1,..,m-1\}$
      - d.h. horizontal passend S(2)= S'(4)
    - `Teilgleich`$_{2p+1,1,p}(V_{i,i}, V_{i+1,i})$ für alle $i \in \{1,..,m-1\}$, $j \in \{1,..,m\}$
      - d.h. vertikal passend: S(3)=S'(1)
- **Korrektheit:**
  - folgt aus der einfachen aussagenlogischen Umformung
- **Laufzeit:**
  - $O(n^2)$, hauptsächlich Erstellen durch der $O(m^2)$ Prädikate `Teilgleich`





FinPred $\leq_{m,p}$ SAT

- **Definition SAT**
  - Gegeben
    - Eine Menge von Booleschen Variablen $x_1,\ldots, x_n$
    - Eine Boolesche Funktion ϕ
  - Gesucht
    - Gibt es eine Belegung von $x_1,\ldots, x_n$, für welche $\phi(x_1,\ldots, x_n)=1$
- **Lemma**
  - FinPred $\leq_{m,p}$ SAT
- **Beweis: Reduktion:**
  - Forme jedes Prädikat als Boolesche Funktion um
    - Ersetze **`Element`**$(x, y_1,\ldots,y_k)$ für $x \in \{0,1\}^p$ durch
      - **`Teilgleich`**$_{1,1,p}(x, y_1) \vee$ **`Teilgleich`**$_{1,1,p}(x, y_2) \vee \ldots \vee$ **`Teilgleich`**$_{1,1,p}(x, y_k)$
    - Ersetze **`Teilgleich`**$_{a,b,c}(x, y)$ für $x= x_1,\ldots,x_p$ und $y= y_1,\ldots,y_p$ durch
      - $[x_a = y_b] \wedge [x_{a+1} = y_{b+1}] \wedge\ldots \wedge[x_{a+c-1} = y_{b+c-1}]$
    - Ersetze $[x_i = y_i]$ durch
      - $(x_i \wedge y_i) \vee (\neg x_i \wedge \neg y_i)$
  - Ver-"unde" alle entstehenden Booleschen Prädikate $T_1, T_2, \ldots, T_r$:
    - $T_1 \wedge T_2 \wedge \ldots \wedge T_r$
- **Laufzeit: O(n)**
- **Korrektheit: folgt aus der Äquivalenz der Umformungen**

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

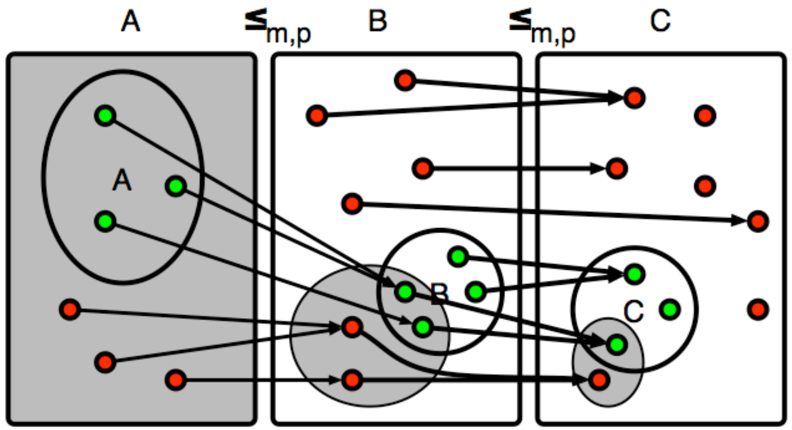
Beweis-Strategie

- **Definition:**
  - Eine Sprache S ist NP-vollständig (NP-complete) wenn:
    - $S \in NP$
    - S ist NP-schwierig
      - d.h. für alle $L \in NP$: $L \leq_{m,p} S$
- **Theorem (Cook/Levin)**
  - SAT ist NP-vollständig
- **Beweis-Idee:**
  - Sei $L \in NP$. Wir beweisen folgende Reduktionen:
    1. Es gibt ein k, so dass $L \leq_{m,p} A_{NTIME(n^k)}$
    2. Für alle k: $A_{NTIME(n^k)} \leq_{m,p} A_{NTIME(n)}$
    3. $A_{NTIME(n)} \leq_{m,p}$ PARKETT
    4. PARKETT $\leq_{m,p}$ FinPred
    5. FinPred $\leq_{m,p}$ SAT
  - Daraus folgt für alle $L \in NP$: $L \leq_{m,p}$ SAT ?
  - Jetzt zeigen wir $SAT \in NP$

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Transitivität von Abbildungsreduktion

- **Lemma:**
  - Aus $A \leq_{m,p} B$ und $B \leq_{m,p} C$ folgt $A \leq_{m,p} C$.
- **Beweis:**
  - Sei f die Reduktionsfunktion von A nach B
  - Sei g die Reduktionsfunktion von B nach C
  - Dann ist $g \circ f$ die Reduktionsfunktion von A nach C.
  - Korrektheit:
    - $x \in A \Leftrightarrow f(x) \in B \Leftrightarrow g(f(x)) \in C$
  - Laufzeit:
    - f(x) kann in Zeit $O(|x|^k)$ berechnet werden
    - g(y) kann in Zeit $O(|y|^{k'})$ berechnet werden
    - $|f(x)| \leq |x|^k$
    - Damit hat g(f(x)) die Laufzeit $O(|f(x)|^{k'}) = O(|x|^{k\,k'})$



SAT ∈ NP

- **Folgende NTM löst das SAT-Problem:**
  - “Auf Eingabe der Funktion $\Phi(x_1,x_2,...,x_n)$
  - Rate nichtdeterministisch eine Belegungen aller Variablen $x_1,x_2,...,x_n$
  - Setze diese Belegung ein
  - Falls $\Phi(x_1,x_2,...,x_n) = 1$ akzeptiere
  - Sonst, verwerfe”
- **Laufzeit:**
  - Die Laufzeit der NTM ist polynomiell, da Boolesche Funktion in Linearzeit ausgewertet werden können
- **Korrektheit**
  - Falls die Formel erfüllbar ist, wird die Belegung durch “Nichtdeterministisches Raten” gefunden
  - Ist die Formel nicht erfüllbar, wird niemals akzeptiert

- **Damit ist SAT als erstes NP-vollständiges Problem identifiziert worden.**

SAT $\leq_{m,p}$ 3-SAT

- **Reduktionsfunktion**
  - Füge für jede Konjunktion oder Disjunktion zweier Teilterme eine neue Variable ein
  - Stelle für alle die Gleichungen auf
    - x = y ∨ z oder x = y ∧ z, wobei x,y,z Literale sind
  - Stelle alle Gleichungen als Klauseln dar mit Literalen x,y,z:
    - x = y ∨ z
      - äquivalent zu (x ∧ (y ∨ z)) ∨ (¬x ∧ ¬(y ∨ z))
      - wird dargestellt als Klauseln: (x ∨ x ∨ ¬y) ∧ (x ∨ x ∨ ¬z) ∧ (¬x ∨ y ∨ z)
    - x = y ∧ z
      - äquivalent zu (x ∧ (y ∧ z)) ∨ (¬x ∧ ¬(y ∧ z))
      - wird dargestellt als Klauseln: (x ∨ ¬y ∨ ¬z) ∧ (¬x ∨ ¬x ∨ y) ∧ (¬x ∨ ¬x ∨ z)

$$\exists x_1, x_2, x_3, x_4 \in \{0,1\} \ : \overline{(((x_1 \wedge \overline{x_2}) \vee \overline{x_3 \vee \overline{x_4}}) \wedge (\overline{x_1} \wedge \overline{x_3})}$$

$$\exists x_1, \ldots, x_4, h_1, \ldots h_5 \in \{0,1\}$$

$$h_4 = x_1 \wedge \overline{x_2} \qquad \overline{h_5} = x_3 \wedge \overline{x_4} \qquad h_3 = \overline{x_1} \wedge \overline{x_3}$$

$$h_2 = h_4 \vee h_5$$

$$\overline{h_1} = h_2 \wedge h_3$$

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

$$\exists x_1, x_2, x_3, x_4 \in \{0,1\} \;:\; \overline{(((x_1 \wedge \overline{x_2}) \vee \overline{x_3 \vee \overline{x_4}}) \wedge (\overline{x_1} \wedge \overline{x_3})}$$

$$\exists x_1, \ldots, x_4, h_1, \ldots h_5 \in \{0,1\}$$

$$h_4 = x_1 \wedge \overline{x_2}$$

$$\overline{h_5} = x_3 \wedge \overline{x_4}$$

$$h_3 = \overline{x_1} \wedge \overline{x_3}$$

$$h_2 = h_4 \vee h_5$$

$$\overline{h_1} = h_2 \wedge h_3$$

$$(h_1 \vee h_1 \vee h_2) \wedge (h_2 \vee h_2 \vee h_3) \wedge (\overline{h_1} \vee \overline{h_2} \vee \overline{h_3})$$

$$(h_2 \vee h_2 \vee \overline{h_4}) \wedge (h_2 \vee h_2 \vee \overline{h_5}) \wedge (\overline{h_2} \vee h_4 \vee h_4)$$

SAT $\leq_{m,p}$ 3-SAT

- **Reduktionsfunktion**
  - Füge für jede Konjunktion oder Disjunktion zweier Teilterme eine neue Variable ein
  - Stelle für alle die Gleichungen auf
    - $x = y \lor z$ oder $x = y \land z$, wobei x,y,z Literale sind
  - Stelle alle Gleichungen als Klauseln dar mit Literalen x,y,z:
    - $x = y \lor z$
      - wird dargestellt als Klauseln: $(x \lor x \lor \neg y) \land (x \lor x \lor \neg z) \land (\neg x \lor y \lor z)$
    - $x = y \land z$
      - wird dargestellt als Klauseln: $(x \lor \neg y \lor \neg z) \land (\neg x \lor \neg x \lor y) \land (\neg x \lor \neg x \lor z)$
- **Korrektheit:**
  - folgt aus der Äquivalenz der Teilformeln
- **Zeitaufwand zur Berechnung der Reduktion:**
  - Zerlegen der Teilformeln in Gleichungssysteme: $O(n^2)$
  - Umwandeln der Gleichungen in 3-KNF: $O(n)$

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Theorem: 3-SAT ist NP-vollständig

- **Beweis:**
  - SAT $\leq_{m,p}$ 3-SAT
    - impliziert dass jedes NP-Problem auf 3-SAT reduziert werden kann
    - daher ist 3-SAT NP-schwierig
  - 3-SAT $\leq_{m,p}$ SAT
    - ist trivial: jede 3-KNF ist eine Boolesche Funktion
    - Da SAT $\in$ NP folgt: 3-SAT $\in$ NP
  - Damit ist 3-SAT NP-vollständig

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Theorem: PARKETT ist NP-vollständig

- **Beweis:**
  - Wir haben bewiesen:
    - Für alle $L \in NP$ gilt: $L \leq_{m,p}$ PARKETT
    - Damit ist PARKETT NP-schwierig (nach Definition)
  - Andererseits:
    - PARKETT $\leq_{m,p}$ SAT
    - Da SAT $\in$ NP, folgt PARKETT $\in$ NP
  - Damit ist PARKETT NP-vollständig

Theorem: CLIQUE ist NP-vollständig

- **Definition CLIQUE:**
  - Gegeben:
    - Ein ungerichteter Graph G
    - Eine Zahl k
  - Gesucht:
    - Hat der Graph G eine Clique der Größe k?
- **Beweis:**
  - Wir haben bereits bewiesen: 3-SAT $\leq_{m,p}$ CLIQUE
    - 3-SAT ist NP-schwierig
    - Dann ist CLIQUE auch NP-schwierig
  - Zu beweisen: CLIQUE ist in NP
    - Konstruiere Verifizierer:
      - Sei $v_1,v_2,..,v_k$ eine beliebige Knotenmenge der Größe k
      - Sei <G,k> die Instanz von CLIQUE
      - Verifiziere, ob die Knoten $v_1,v_2,..,v_k$ verschieden sind und ob alle Kanten zwischen den Knoten existieren
    - Laufzeit: $O(n^2)$

Ende der 22. Vorlesung

CoNe
Freiburg

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer
Wintersemester 2006/07
22. Vorlesung
19.01.2007