Informatik III

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

Christian Schindelhauer
Wintersemester 2006/07
21. Vorlesung
18.01.2007

Komplexitätstheorie - Zeitklassen

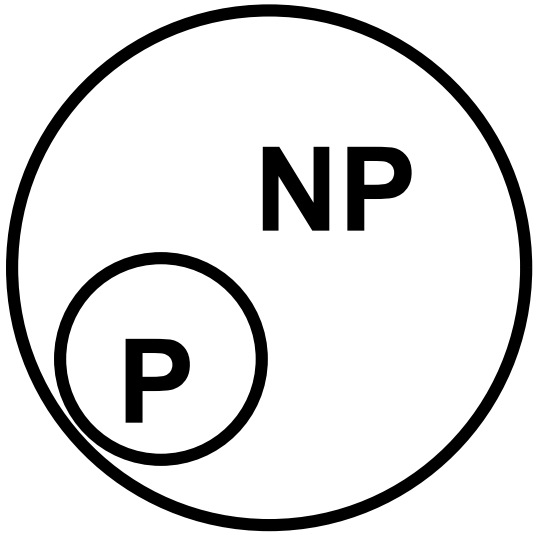
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **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 Frage P versus NP (I)

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

- **P = Klasse aller Probleme, die effizient *entschieden* werden können**
- **NP = Klasse aller Problem, die effizient *verifiziert* werden können**
  - Beispiele: Hamiltonscher Pfad, Clique, Teilsummenproblem, Koffer-Pack-Problem
- **Es gibt jetzt zwei Möglichkeiten**

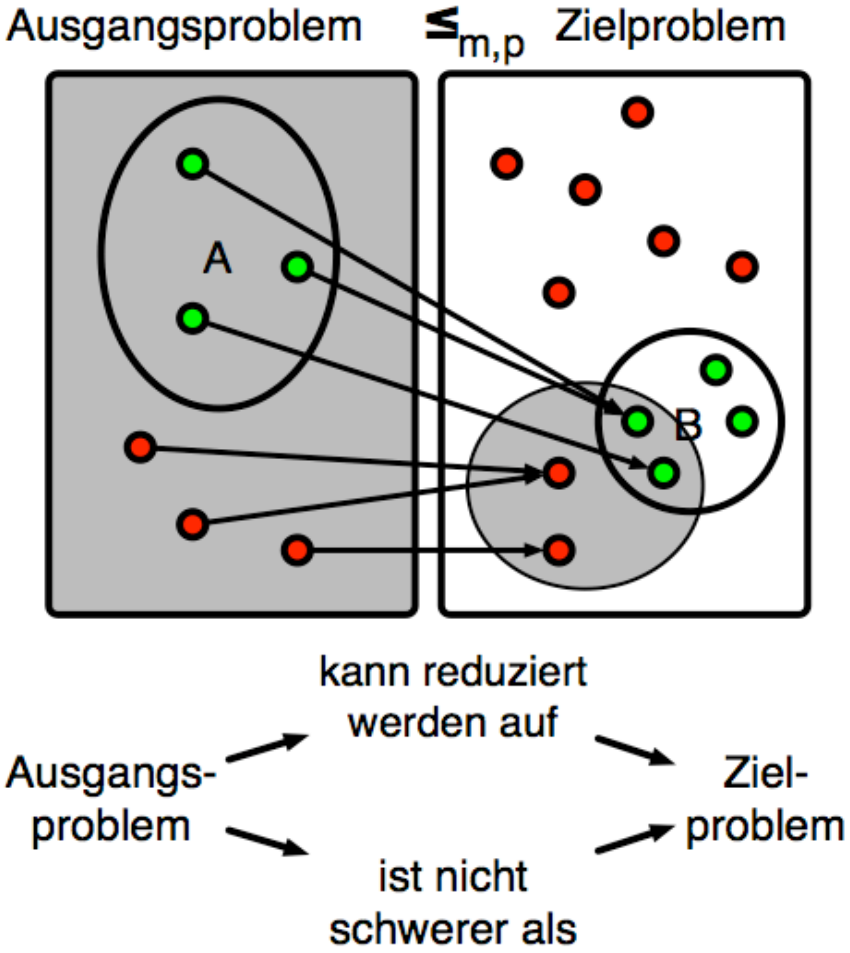


- **Was weiß man?**
  - **P ⊆ NP ⊆** $\bigcup_k \mathrm{TIME}(2^{n^k})$ **= EXPTIME**

Die Polynom-Zeit-Abbildungsreduktion

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

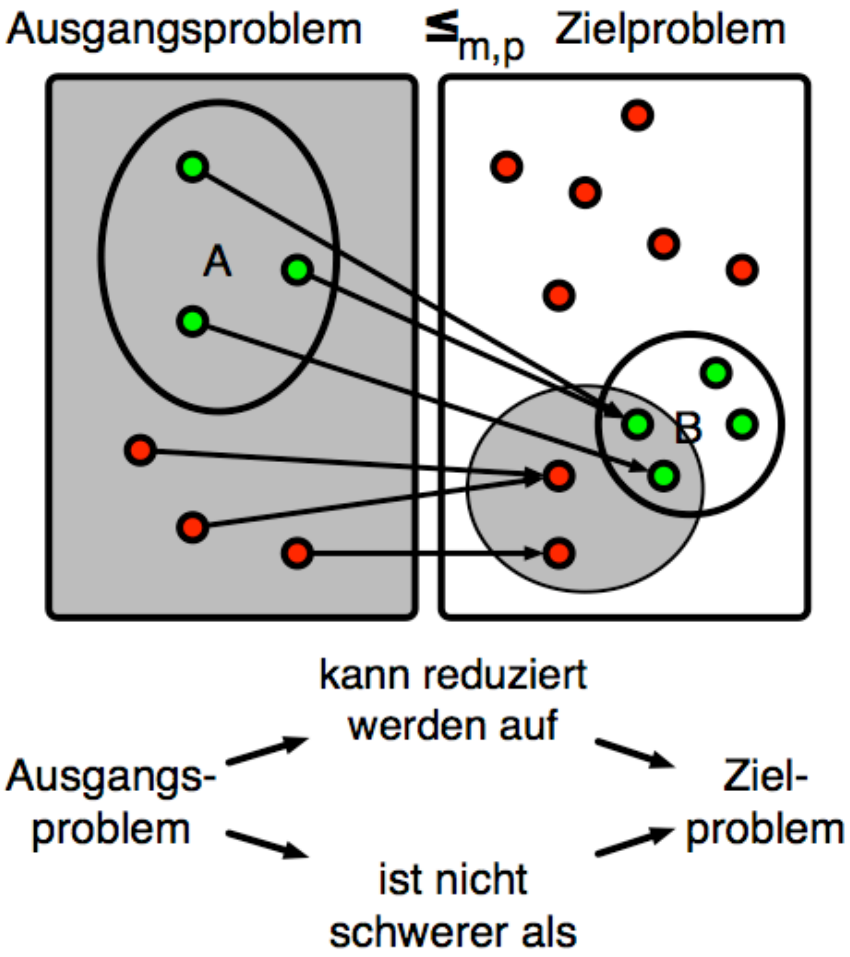
- **Definition (Abbildungsreduktion, Polynomial Time Mapping Reduction, Many-one)**
  - Eine Sprache A kann durch Abbildung auf eine Sprache B in Polynom-Zeit reduziert werden: **$A \leq_{m,p} B$**,
    - falls es eine in Polynom-Zeit berechenbare Funktion $f: \Sigma^* \to \Sigma^*$ gibt,
    - so dass 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

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

- **Literal:**
  - ist eine einfache oder negierte Variable
  - z.B.: x, y, z, ¬x, ¬y, ¬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

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

- **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.:**

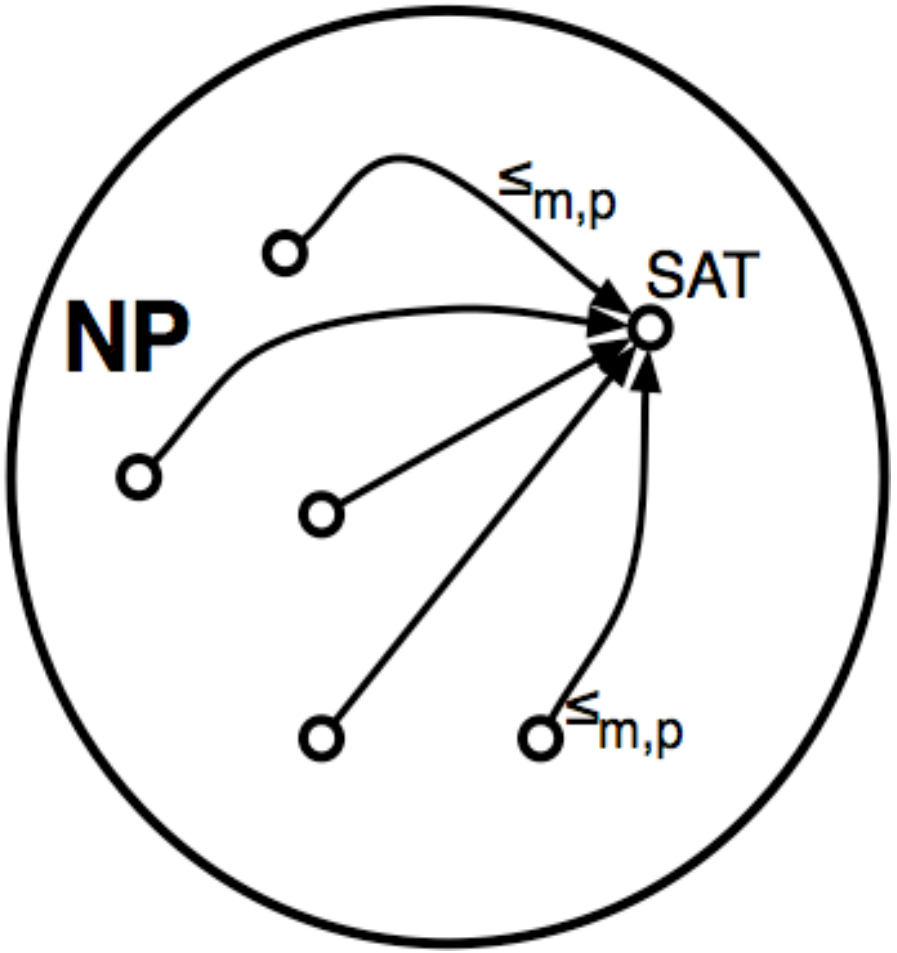
$$\psi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

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 ∈ NP
    - S ist NP-schwierig
      - d.h. für alle L ∈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 $SAT \in NP$

Es gibt ein k, so dass $L \leq_{m,p} A_{NTIME(n^k)}$

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

- **Definition**
  - Das Wortproblem von NTIME(n^k):
  - $A_{NTIME(n^k)}$ = { <M,x> | NTM M akzeptiert $x \in \Sigma^n$ in Laufzeit n^k }
- **Lemma**
  - Für alle $L \in NP$ gibt es ein $k \in \mathbf{N}$, so dass $L \leq_{m,p} A_{NTIME(n^k)}$
- **Beweis**
  - Sei $L \in NTIME(n^k)$, dann gibt es eine NTM M die in Laufzeit n^k jede Eingabe $x \in \Sigma^n$ akzeptiert.
  - Definiere Reduktionsfunktion: f(x) = <M,x>, für diese NTM M
  - Korrektheit:
    - $x \in L \Leftrightarrow$ M akzeptiert in Laufzeit n^k die Eingabe x
      $\Leftrightarrow$ <M,x> $\in A_{NTIME(n^k)}$
  - Reduktionsfunktion hat Laufzeit: |x| + O(1)
    - Da die Eingabe mit der (konstanten) Maschine M kombiniert werden muss.

Für alle $k \geq 1$: $A_{NTIME(n^k)} \leq_{m,p} A_{NTIME(n)}$

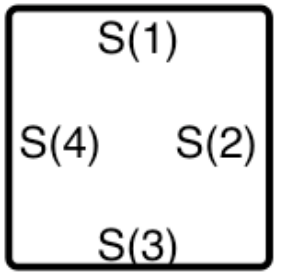
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

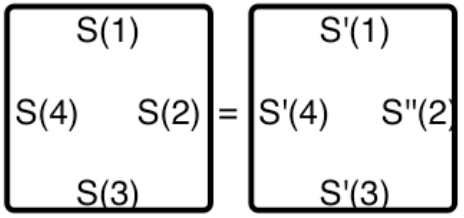
- **Definition**
  - Das Wortproblem von NTIME(n^k):
  - $A_{NTIME(n^k)} = \{ \langle M,x \rangle \mid$ NTM M akzeptiert $x \in \Sigma^n$ in Laufzeit $n^k \}$
- **Lemma**
  - Für alle $k \geq 1$: $A_{NTIME(n^k)} \leq_{m,p} A_{NTIME(n)}$
- **Beweis**
  - Betrachte h(M) = M':
    - modifiziert NTM M zu einer NTM M' mit gleicher Laufzeit und Akzeptanzverhalten nur dass das Zeichen "#" wie "_" behandelt wird
  - Reduktionsfunktion: $f(\langle M,x \rangle) = \langle h(M), x\ \#^{n^k-n} \rangle$
- **Laufzeit auf Eingabe der Länge n ist $O(n + n^k) = O(n^k)$**
- **Korrektheit:**
  - $\langle M,x \rangle \in A_{NTIME(n^k)}$ $\Leftrightarrow$ M akzeptiert in Laufzeit n^k die Eingabe x
    $\Leftrightarrow$ M' akzeptiert in Laufzeit n^k die Eingabe $x\ \#^{n^k-n}$
    $\Leftrightarrow$ M' akzeptiert in Laufzeit $n' = n^k$ die Eingabe $x\ \#^{n^k-n}$ mit Länge n'
    $\Leftrightarrow \langle M'\ x\ \#^{n^k-n} \rangle \in A_{NTIME(n)}$

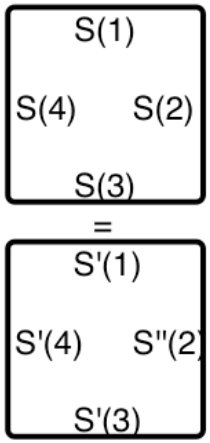
Das Parkett-Problem

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

- **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) ∈ {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 = $\{B_1,B_2,B_3,..., B_k\}$
    - Oberer Rand $S_{1,1}$, ..., $S_{1,m}$
    - Unterer Rand $S_{m,1}$, ..., $S_{m,m}$
  - Gesucht:
    - Gibt es eine Menge von Bauteilen $S_{i,j}$, für i ∈ {1,..,m} , j ∈ {2,..,m-1} die passen, d.h.
      - H($S_{i,j}$,$S_{i,j+1}$) für alle i ∈ {1,..,m}, j ∈ {1,..,m-1} und
      - V($S_{i,j}$,$S_{i+1,j}$) für alle i ∈ {1,..,m-1}, j ∈ {1,..,m}.

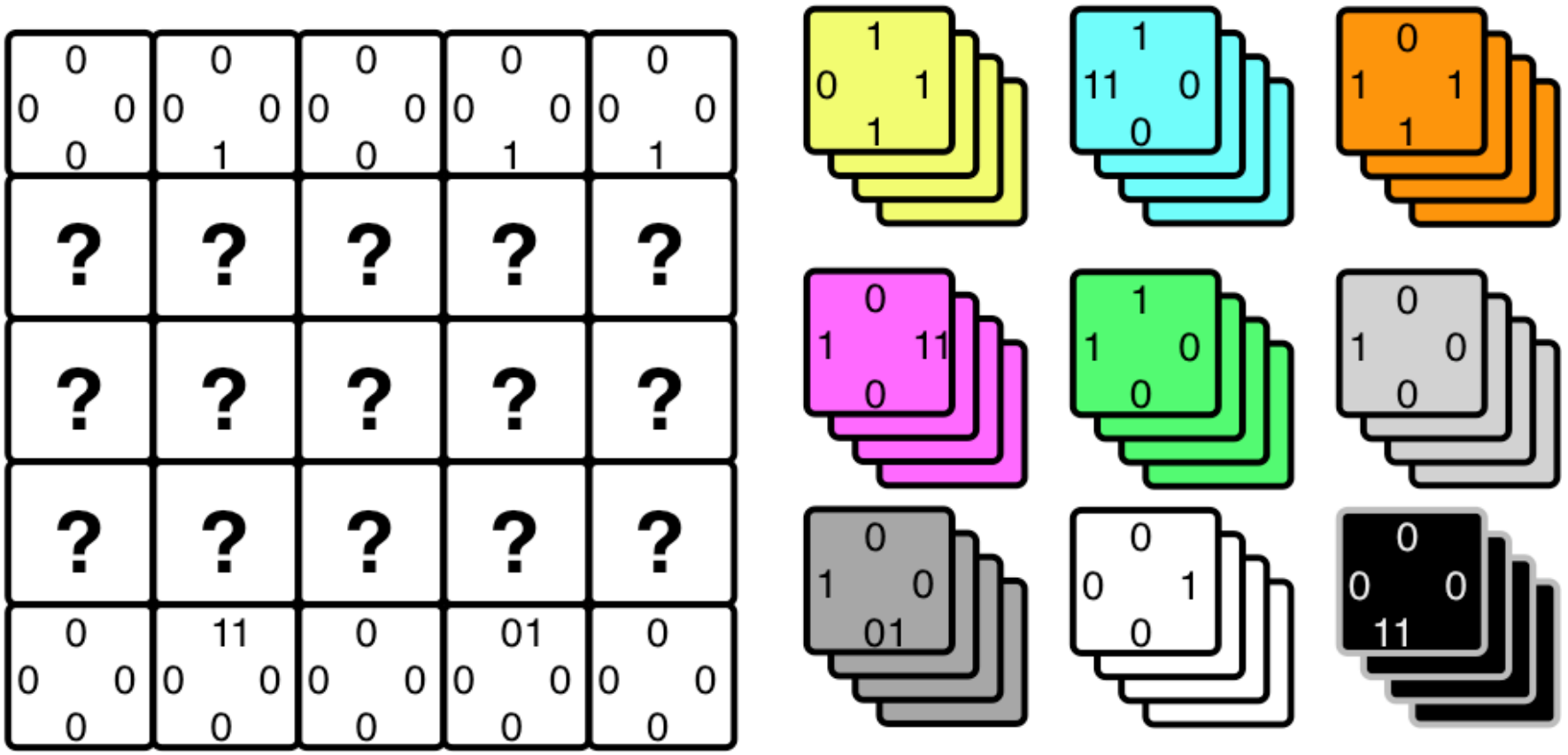






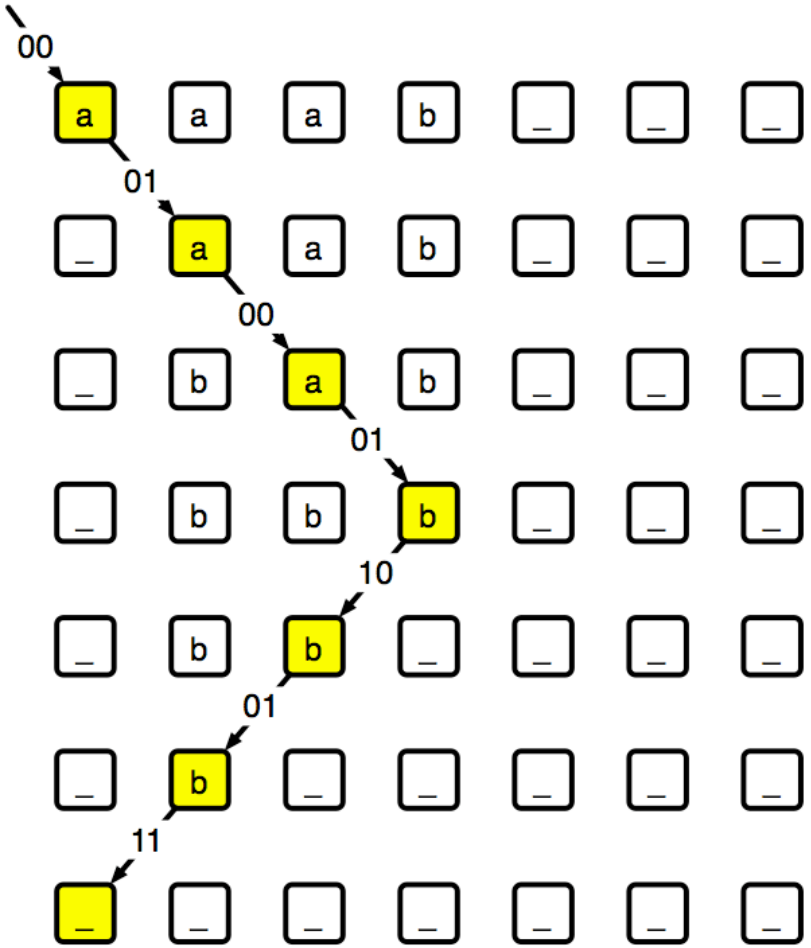
Das Parkett-Problem

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

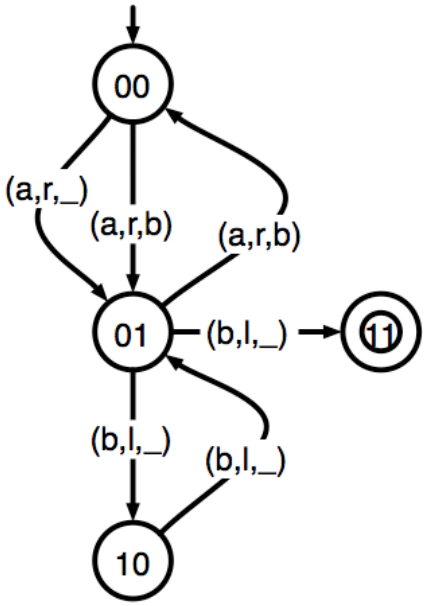


$A_{NTIME(n)} \leq_{m,p}$ PARKETT

➢ **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

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

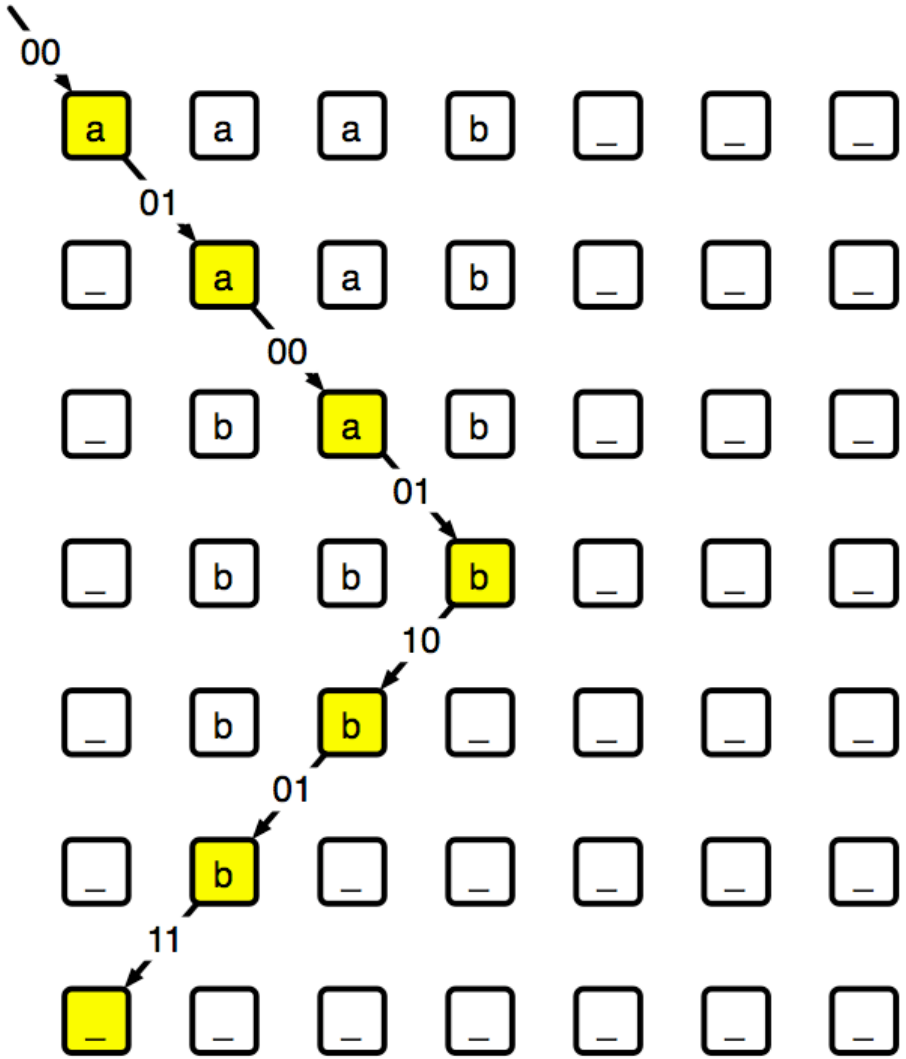
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Sei bin(x) eine Kodierung jeweils der endlichen Zustände und Eingabesymbole als eindeutige Binärzahl mit der festen Länge 1+ log max{|Σ|,|Q|}**

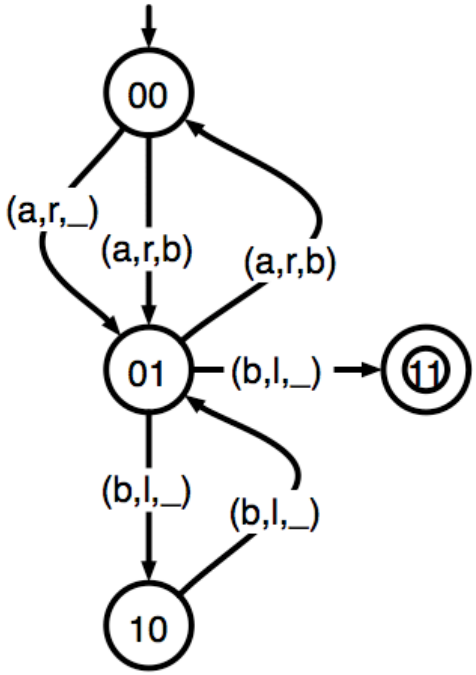
- **Kodiere die Linearzeit-Berechnung wie folgt:**
  - Die Bandzelle j im Schritt i wird als Bauteil $S_{i,j}$ kodiert
  - Ist der Kopf der NTM auf der Bandzelle wird in die Bandzelle zusätzlich der Zustand kodiert
  - Für die obere Zellen ergeben sich folgende Teile
    - $S_{1,1}$ = (0, 0, 1 bin(x_1) bin(q_0), 0)
    - $S_{1,i}$ = (0, 0, 0 bin(x_i), 0) für $i \geq 2$
  - Für die untere Zeile ergeben sich die Teile
    - $S_{m,1}$ = (1 bin("_") bin(q_{akz}) 0, 0, 0, 0)
    - $S_{m,i}$ = (1 bin("_"), 0, 0 , 0) für $i \geq 2$

Berechnung einer 1-Band-NTM

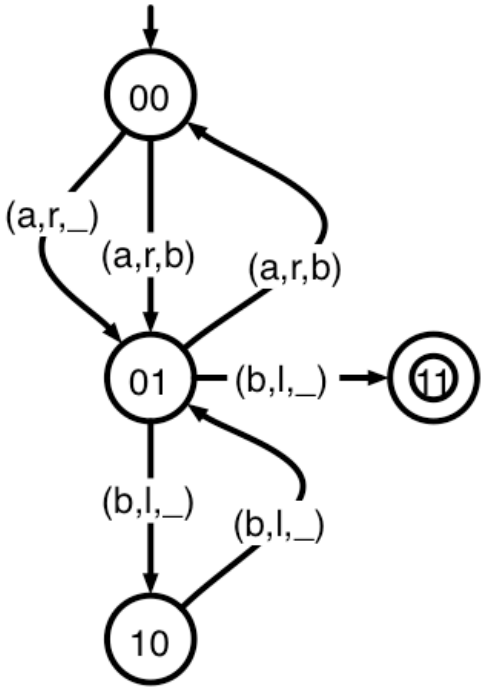
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer



Akzeptiert auf leerer Eingabe



0 0 0 1a00	0 0 0 0a	0 0 0 0a	0 0 0 0b	0 0 0 0_	0 0 0 0_	0 0 0 0_	0 0 0 0_
1a00 0 101 0_	0a 101 0 1a01	0a 0 0 0a	0b 0 0 0b	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_
0_ 0 0 0_	1a01 0 100 0b	0a 100 0 1a00	0b 0 0 0b	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_
0_ 0 0 0_	0b 0 0 0b	1a00 0 101 0b	0b 101 0 1b01	0 0 0 0_	0 0 0 0_	0 0 0 0_	0 0 0 0_
0_ 0 0 0_	0b 0 0 0b	0b 0 110 1_10	1b01 110 0 0	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_
0_ 0 0 0_	0b 0 101 1_01	1_10 101 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_
0_ 0 111 1_11	1_01 111 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_
1_11 0 0 0	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_	0_ 0 0 0_



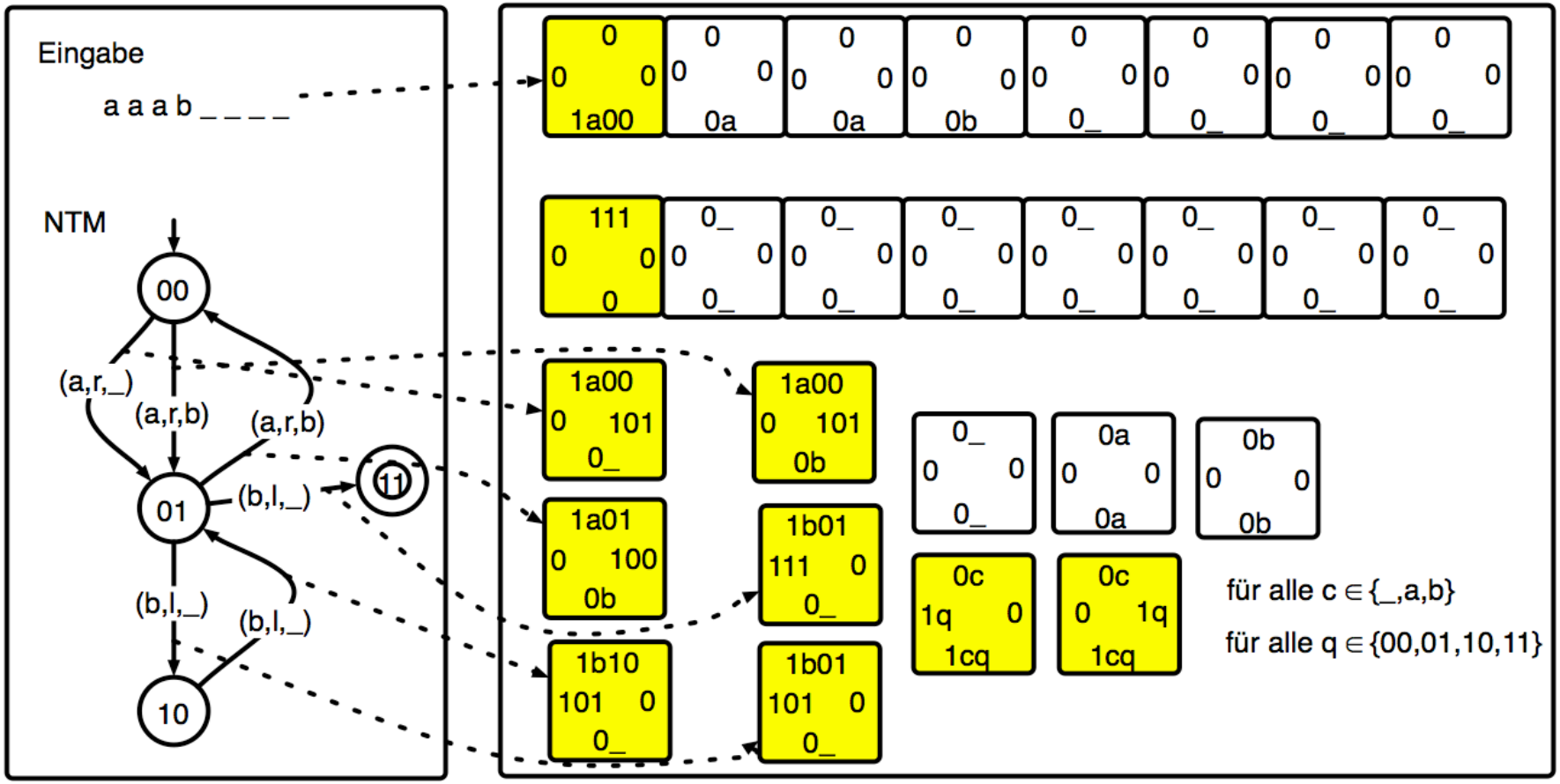
Parkettierung gelungen

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

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

- **Kodiere die Berechnungsschritte wie folgt:**
  - Die Bandzelle j im Schritt i wird als Bauteil $S_{i,j}$ kodiert
  - Ist der Kopf der NTM auf der Bandzelle wird in die Bandzelle zusätzlich der Zustand kodiert
  - Für die inneren Zellen gibt es folgende Fälle:
    - Der Kopf der NTM ist vor und nach einem Berechnungsschritt i nicht in der Zelle j, sei z das Zeichen auf dem Band und bin(z) die Binärdarstellung
      - entspricht Baustein (0 bin(z), 0, 0 bin(z), 0)
    - Der Kopf der NTM mit Zustand q bewegt sich von links in die Zelle mit Zustand
      - entspricht Baustein (0 bin(z), 0, 1 bin(z) bin(q), 1 bin(q))
    - Analog von rechts:
      - entspricht Baustein (0 bin(z), 1 bin(q) , 1 bin(z) bin(q), 0 , 0)
    - Der Kopf war im Schritt zuvor in der Zelle.
    - Die Berechnung ergibt aus Zustand q mit Symbol z, Nachfolgezustand q' mit Symbol z' und Bewegungsrichtung links
      - entspricht Baustein (1 bin(z) bin(q) , 0, 0 bin(z'), 1 bin(q'))
    - Entsprechend rechts
      - entspricht Baustein (1 bin(z) bin(q) , 1 bin(q'), 0 bin(z'), 0)

Beispielreduktion



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

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

- **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

Ende der 21. Vorlesung

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

Christian Schindelhauer

Wintersemester 2006/07

21. Vorlesung

18.01.2007