Informatik III

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

Arne Vater

Wintersemester 2006/07

26. Vorlesung

02.02.2007

Ein Platzkomplexitätsmaß

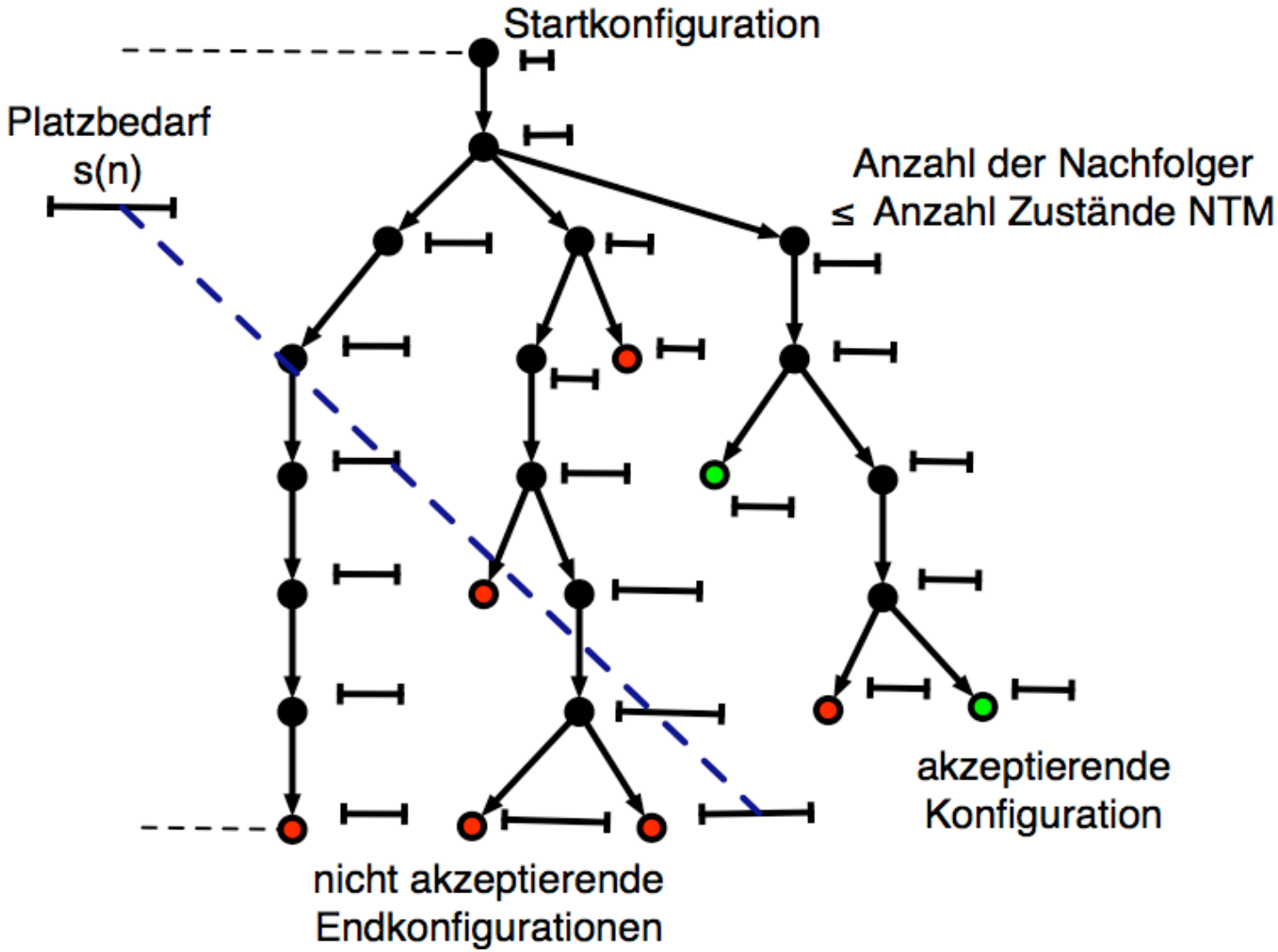
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Definition**
  - Sei M eine deterministische Turing-Maschine, die auf allen Eingaben hält.
  - Der (Speicher-) **Platzbedarf (Platzkomplexität)** von M ist eine Funktion f: N → N,
    - wobei f(n) die maximale Anzahl von Bandzellen von M ist, die M verwendet (ein Kopf der TM zeigt auf eine Bandzelle)
  - Falls f(n) der Platzverbrauch einer TM M ist, nennt man M auch eine **f(n)-Platz-Turing-Maschine**
    - z.B. Linear-Platz-TM für f(n) = c n für eine Konstante c
    - z.B. Polynom-Platz-TM für f(n) = c n^k für Konstanten c und k

- Zumeist beschreibt n die Eingabelänge.

Nichtdeterministische Platzkomplexitätsklassen

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



Platzkomplexitätsklassen

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

➢ **Definition**

– Sei f: N → N eine Funktion. Die Platzkomplexitätsklasse SPACE(f(n)) und NSPACE(f(n)) sind wie folgt definiert

– **SPACE(f(n))** = { L | L ist eine Sprache, die durch eine O(f(n))-Platz-**DTM** entscheiden wird }

– **NSPACE(f(n))** = { L | L ist eine Sprache, die durch eine O(f(n))-Platz-**NTM** entscheiden wird }

– Wird die Anzahl der Bänder auf k beschränkt, schreiben wir

- **$SPACE_{k\text{-}Band}(f(n))$** oder einfach **$SPACE_k(f(n))$,**
- **$NSPACE_{k\text{-}Band}(f(n))$** oder einfach **$NSPACE_k(f(n))$.**

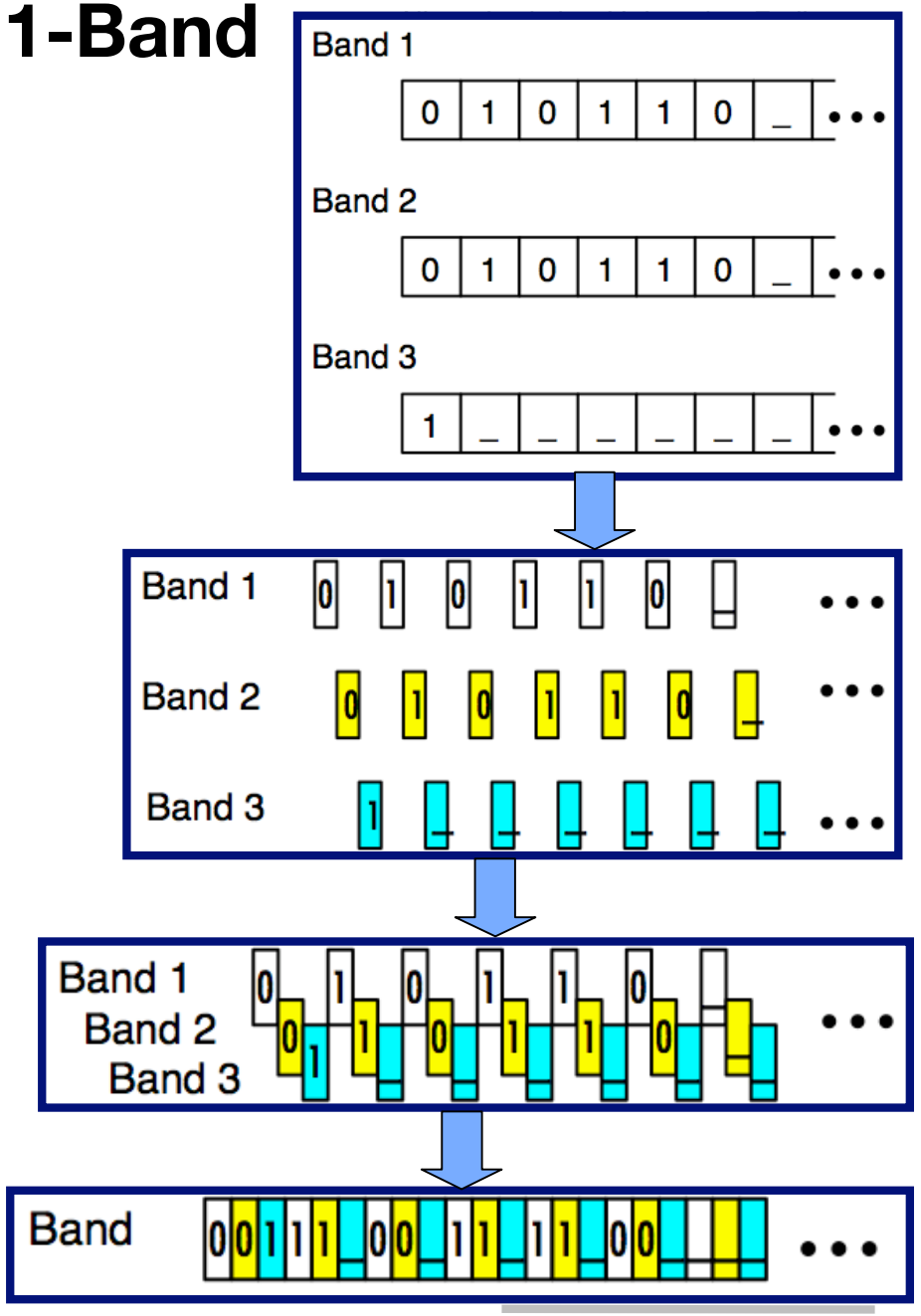
Beispiel: SAT ist in $SPACE_2(N)$

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

- **Betrachte folgende 2-Band-DTM:**
  - "Gegeben eine boolesche Formel F mit m Variablen x_1, ..., x_m
  - Für alle Belegungen von z_1, ..., $z_m \in \{0,1\}^m$
    - Setze Belegung z_1, ..., z_m in F ein
      - Ist $F(z_1, ..., z_m)$ wahr, dann halte und akzeptiere
    - sonst verwerfe"
- **Laufzeitanalyse:**
  - Auswerten der Funktion: O(n)
  - Anzahl verschiedener Belegungen ($m \leq n$): 2^n
  - Insgesamt: $O(n\, 2^n)$
- **Platzbedarf:**
  - Auswerten der Funktion: O(n)
  - Speichern der Belegung: O(n)
  - Insgesamt: O(n)

k-Band-DTMs → 1-Band DTMs

- **Theorem**
  - **$SPACE_k(s(n)) \subseteq SPACE_1(O(s(n))$,** d.h.
  - Für $s(n) \geq n$ kann jede Berechnung einer **k-Band-s(n)-Platz-DTM** von einer **1-Band-(k s(n))-Platz DTM** berechnet werden.
- **Beweis(anfang):**
  - Betrachte k-Band-DTM M mit Arbeitsalphabet Γ
  - und konstruiere 1-Band-DTM mit Alphabet $\Gamma \times \{\_, kopf\}$,
    - Speichere das i-te Symbol von Band j an der Position j + i k.
    - Verwende Markierung *kopf* nur wenn der Kopf der k-Band-TM an der entsprechenden Stelle steht.
  - ...



k-Band-DTMs → 1-Band DTMs

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

- **Theorem**
  - **$SPACE_k(s(n)) \subseteq SPACE_1(s(n))$**
- **Beweis (Fortsetzung)**
  - Arbeitsweise: 1-Band-DTM
    1. Kodiere Eingabe passend um
    2. Für jeden Schritt der k-Band-DTM
    3. Für jedes Band j
    4. Suche Kopfposition an den Stellen {j,j+k,j+2k,...}
    5. Lies Eingabe
    6. Berechne den Folgezustand, neues Symbol und Bewegungsrichtung
    7. Für jedes Band
    8. Suche Kopfposition
    9. Schreibe neues Zeichen
    10. Bewege Kopfmarkierung um k Zeichen nach links oder rechts
    11. Falls M hält, halte und akzeptiere/verwerfe entsprechend

k-Band-DTMs → 1-Band DTMs

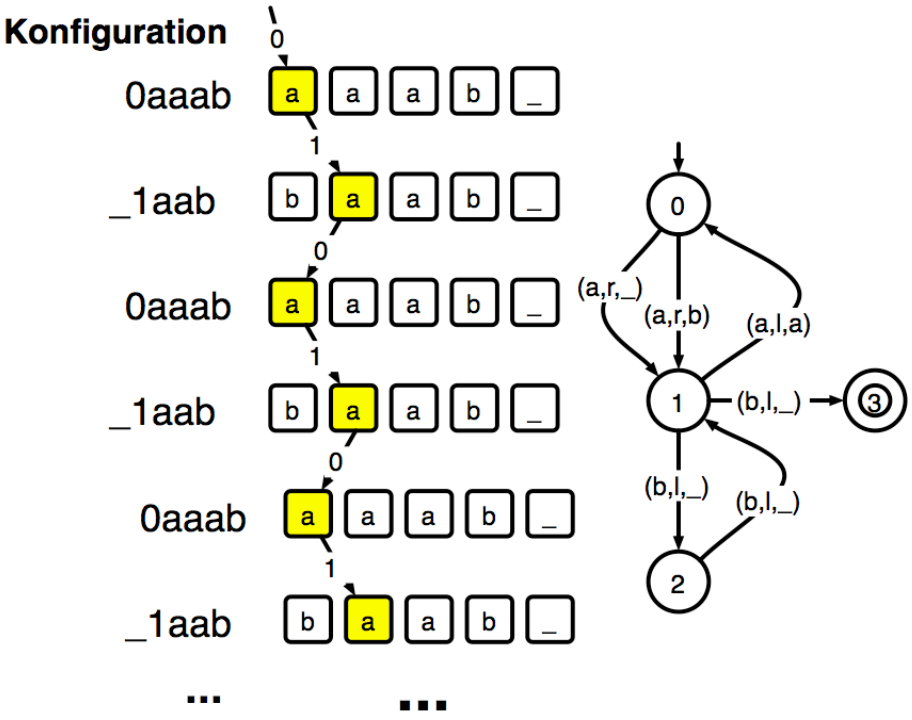
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Theorem**
  - **$SPACE_k(s(n)) \subseteq SPACE_1(k\ s(n))$**
- **Beweis (Speicherplatz):**
  - Da die k-Band-DTM höchstens s(n) Zellen besucht, wird die 1-Band-DTM höchstens k s(n) Bandzellen besuchen.
- **Theorem**
  - **Jede Berechnung einer s(n)-Platz-DTM kann durch eine max{n,s(n)/k}-Platz-DTM durchgeführt werden.**
- **Beweis**:
  - Erweitere das Bandalphabet von Σ auf Σ^k.
  - Jeweils k benachbarte Zeichen $a_1,a_2,..,a_k$ werden in das Zeichen $(a_1,a_2,..,a_k)$ umgeformt
  - Die Zustandsmenge wird entsprechend vergrößert
  - und die Zustandsübergänge angepasst:
    - Zuerst kompriminiert die neue DTM die Eingabe um den Faktor k
    - Dann führt sie die Berechnung analog auf dem komprimierten Zeichensatz durch.

Maximale Berechnungszeit einer s(n)-Platz-DTM/NTM

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

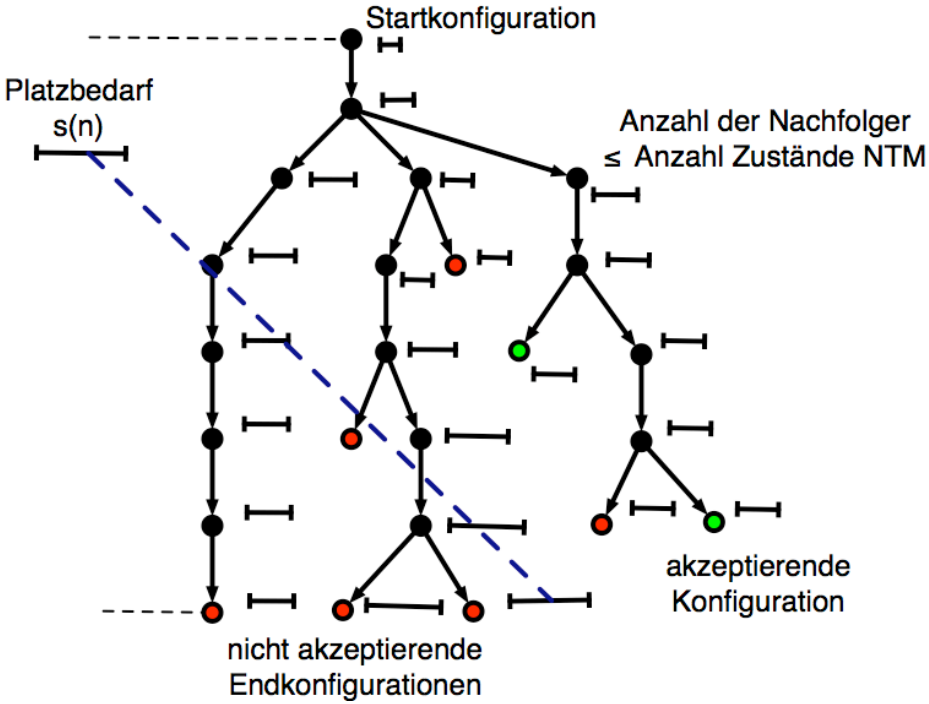
- **Lemma**
  - Jede s(n)-Platz-DTM hat eine Laufzeit von $2^{O(s(n))}$.
- **Beweis:**
  - Es gibt höchstens $2^{O(s(n))}$ verschiedene Konfigurationen der DTM
  - Wiederholt sich eine Konfiguration, so wiederholt sich diese immer wieder und die DTM hält niemals
- **Lemma**
  - Jede s(n)-Platz-NTM hat eine Laufzeit von $2^{O(s(n))}$.
- **Beweis**
  - Analog zur DTM:
  - Wenn sich auf einem Berechnungspfad eine Konfiguration wiederholt, dann wird sie sich immer wieder und wieder und wieder wiederholen.



Nicht der Satz von Savitch

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

- **Schwacher Satz: Für $s(n) \geq n$**
  - **$NSPACE_1(s(n)) \subseteq SPACE_3(2^{O(s(n))})$, d.h.**
  - Jede Berechnung einer s(n)-Platz 1-Band-NTM kann von einer 3-Band DTM in Platz $2^{O(s(n))}$ durchgeführt werden.
- **Beweis:**
  - Simuliere alle Berechnungspfade der NTM durch
  - Dadurch wird die Berechnung deterministisch
- **Warum ist die Platzschranke so schlecht?**
  - Der Baum kann exponentiell tief sein: $2^{O(s(n))}$
  - Dann braucht man allein zum Abspeichern, welchen Ast man gerade berechnet ein exponentiell langes Wort



Der Satz von Savitch

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

- **Theorem: Für jede Funktion $s(n) \geq n$**
  - **$NSPACE_1(s(n)) \subseteq SPACE_3(s^2(n))$, d.h.**
  - Jede Berechnung einer s(n)-Platz 1-Band-NTM kann von einer 3-Band DTM in Platz **$s^2(n)$** durchgeführt werden.
- **Beweis:**
  - Betrachte NTM für $L \in NSPACE_1(s(n))$, die mit einer **eindeutigen Konfiguration C_{akz}** akzeptiert, d.h.
    - Es gibt nur einen akzeptierenden Zustand
    - Am Ende der Berechnung wird das Band gelöscht
  - Betrachte das Prädikat **`Erreicht-Konf`(C,C',S,T):**
    - Dieses Prädikat ist wahr, wenn die S-Platz-NTM M ausgehend von der Konfiguration C die Konfiguration C' innerhalb von T Schritten erreicht.
  - **Lemma: `Erreicht-Konf`(C,C',S,T) kann von einer 2-Band-(S log T)-Platz-DTM entschieden werden**
    - noch zu beweisen
  - Nun ist $T \leq 2^{O(s(n))}$ und damit ist $\log T = O(s(n)) \leq c\, s(n)$ für eine Konstante $c>0$.
  - **Sei C_{start} die Startkonfiguration**
  - **Das Prädikat `Erreicht-Konf`(C_{start}, C_{akz}, s(n), $2^{c\, s(n)}$) entscheidet L.**
  - Dann kann eine 3-Band-DTM L in Platz
    $c\, s(n)\, s(n) = c\, s(n)^2 = O(s^2(n))$ die Sprache L entscheiden

Beweis des Lemmas

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

- **Lemma: Das Prädikat Erreicht-Konf(C,C',S,T) kann eine DTM mit 2 Bändern mit Platzbedarf s(n) log T entscheiden.**
  - Diese Prädikat **Erreicht-Konf(C,C',S,T)** ist wahr, wenn die S-Platz-NTM M ausgehend von der Konfiguration C die Konfiguration C' innerhalb von T Schritten erreicht.
- **Beweis**
  - Betrachte folgende DTM M' auf Eingabe (C,C',S,T)
    - Falls T=0 dann
      - Akzeptiere falls C=C' und verwerfe falls C≠C'
    - Falls T=1 dann
      - Akzeptiere falls C' eine erlaubte Nachfolgekonfiguration von C ist oder C=C', ansonsten verwerfe
    - Falls T>1 dann
      - Für alle Konfigurationen Z der Länge S
        - Berechne rekursiv r_1 = Erreicht-Konf(C,Z,S,$\lfloor T/2 \rfloor$)
        - Berechne rekursiv r_2 = Erreicht-Konf(Z,C',S,$\lceil T/2 \rceil$)
        - Falls r_1 und r_2 gilt, dann halte und akzeptiere
      - Verwerfe

- **Analyse Platzverbrauch**
- **Platzverbrauch = Eingabelänge = s(n)**
- **Platzverbrauch: zusätzlich s(n)+1 in jeder Rekursionstiefe**
- **Anzahl Rekursionstiefen: log T**

Der Satz von Savitch (Nachschlag)

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

- **Theorem: Für jede Funktion $s(n) \geq n$**
  - **$NSPACE_1(s(n)) \subseteq SPACE_3(s^2(n))$, d.h.**
- **Beweis:**
  - Zusammenfassung
    - Sei C_{start} die Startkonfiguration
    - Sei C_{akz} die eindeutige akzeptierende Endkonfiguration
    - Das Prädikat **`Erreicht-Konf`**$(C_{start}, C_{akz}, s(n), 2^{c\ s(n)})$ entscheidet L in Platz $O(s^2(n))$

Drei wichtige Komplexitätsklassen

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

- **Definition:**
  - $$\mathrm{P} = \bigcup_k \mathrm{TIME}(n^k)$$
  - $$\mathrm{NP} = \bigcup_k \mathrm{NTIME}(n^k)$$

$$\mathrm{PSPACE} = \bigcup_k \mathrm{SPACE}(n^k)$$

- **Noch mal:**
  - P: Klasse aller Sprachen, die von einer Polynom-Zeit DTM entschieden werden
  - NP: Klasse aller Sprachen, die von einer Polynom-Zeit NTM entschieden werden können.
  - PSPACE: Klasse aller Sprachen, die von einer Polynom-Platz-DTM oder Polynom-Platz-NTM etnschieden werden können.

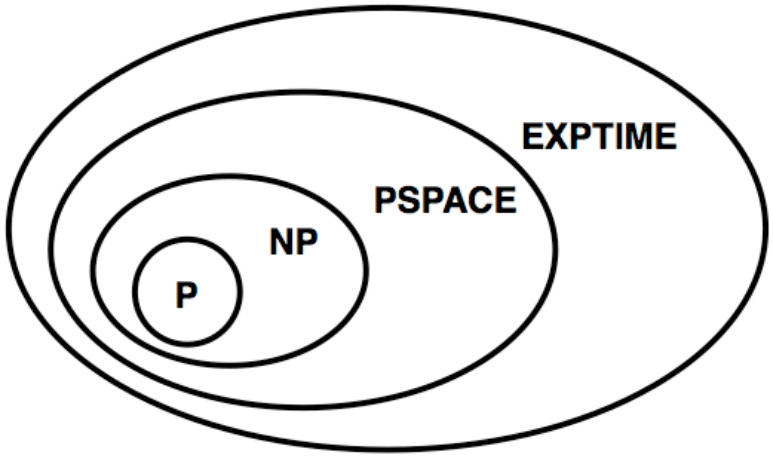
Die Frage P versus NP versus PSPACE

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 Probleme, die effizient *verifiziert* werden können
- **PSPACE =** Klasse aller Probleme, die auf polynomiellem Platz entschieden werden können
- **EXPTIME =** $\bigcup_k \mathbf{TIME}(2^{n^k})$
- **Man weiß nur, dass P $\neq$ EXPTIME und**

$$\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXPTIME}$$

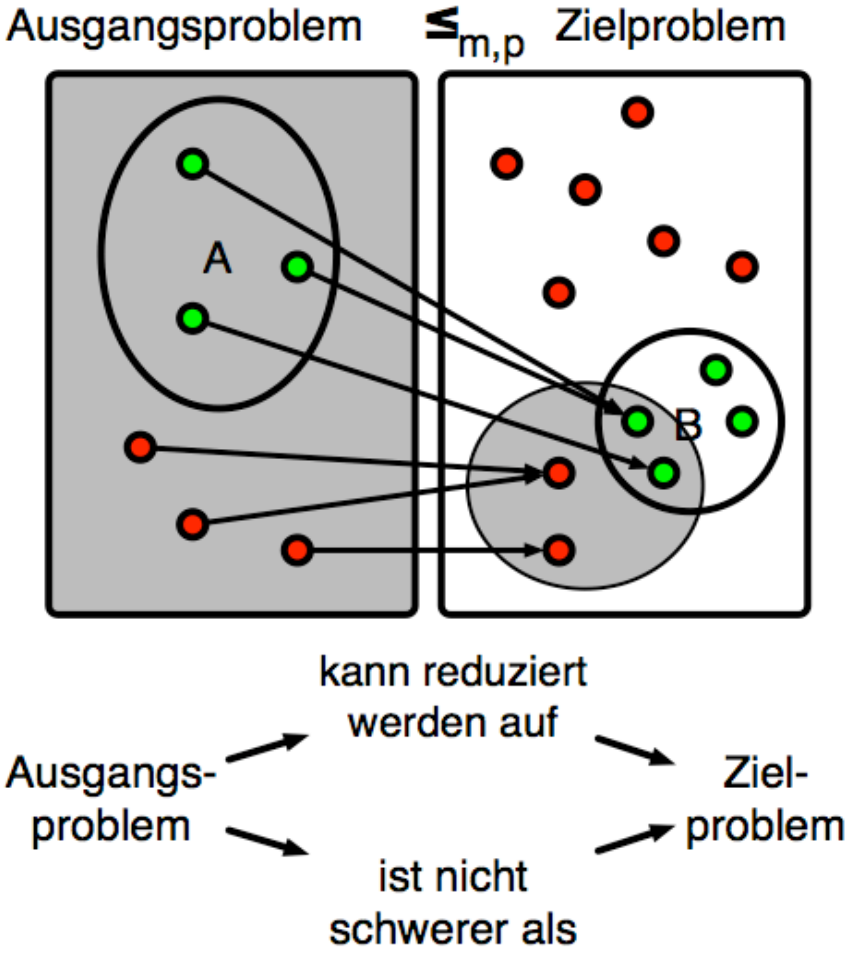
- **Allgemein wird aber vermutet, dass alle Inklusionen echt sind, d.h.**



Die Polynom-Zeit-Abbildungsreduktion

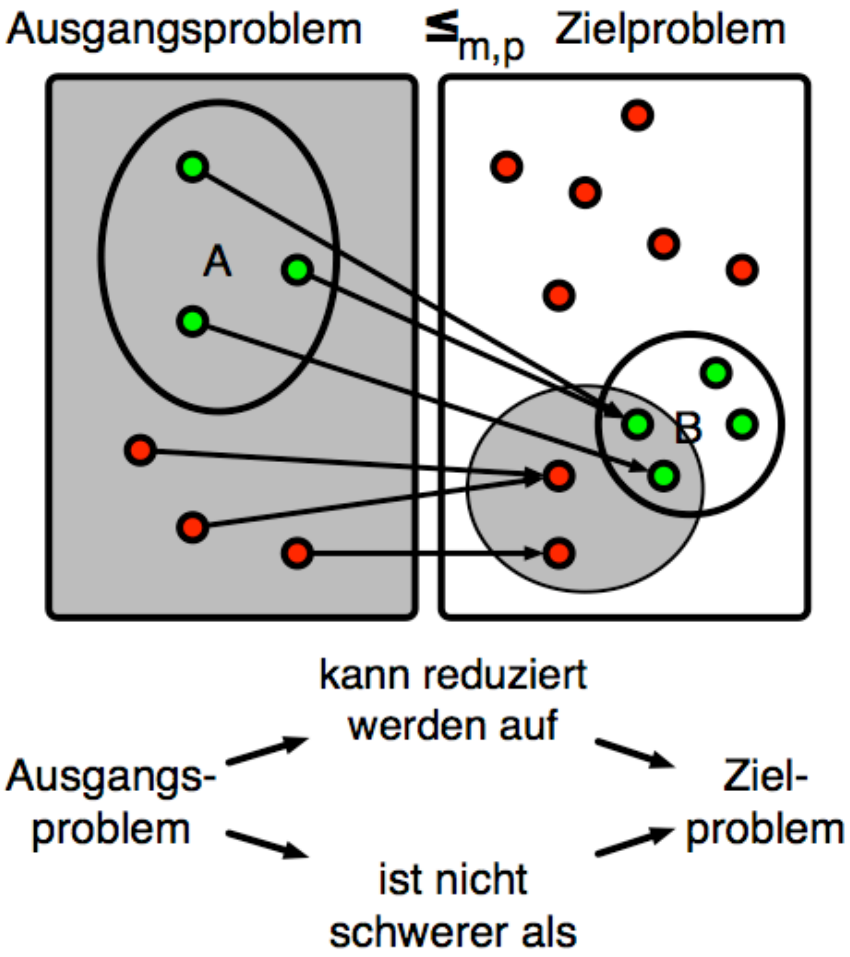
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **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, PSPACE

- **Theorem**
  - Falls $A \leq_{m,p} B$ und B ist in PSPACE, dann ist A auch in PSPACE.
- **Beweis:**
  - Sei M eine Polynom-Platz-DTM für B
  - Betrachte folgende DTM M' für A:
    - Auf Eingabe x
    - Berechne f(x)
    - Berechne M(f(x))
    - Gib Ergebnis aus
  - Platzbedarf:
    - Berechnung f(x):
      - Polynom-Zeit und damit Polynom-Platz
    - Berechnung M(f(x)):
      - Platz: polynomiell in $|f(x)| \leq |x|^k$
      - Damit wiederum polynomiell

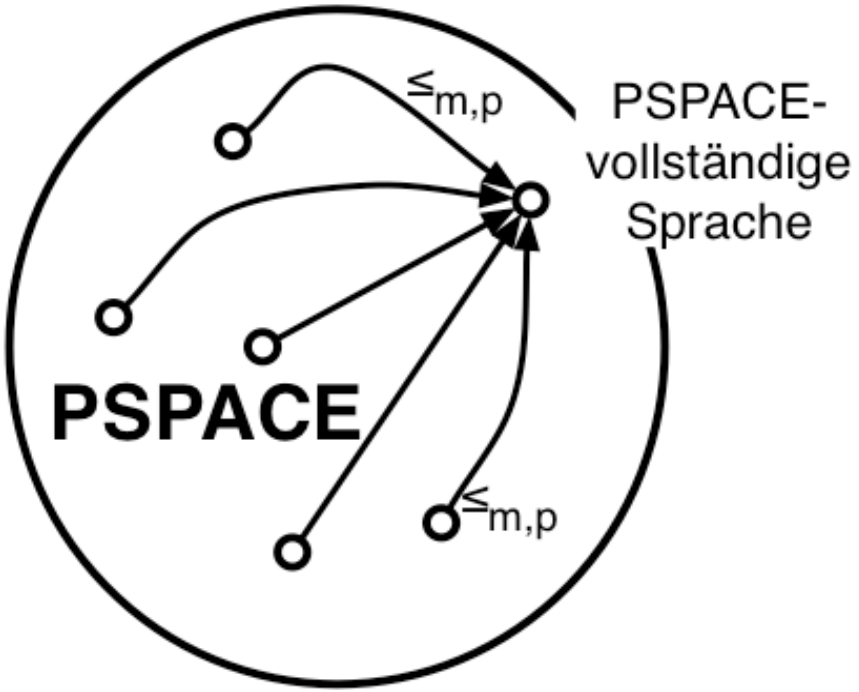


PSPACE-Vollständigkeit

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

➢ **Definition:**

- Eine Sprache S ist **PSPACE-schwierig**, falls
  - für alle L ∈ PSPACE: $L \leq_{m,p} S$
- Eine Sprache S ist **PSPACE-vollständig** wenn:
  - S ∈ PSPACE
  - S ist PSPACE-schwierig



Quantifizierte Boolesche Formeln

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

- **Eine Boolesche Funktion ist definiert durch**
  - Eine Konstante 0 oder 1
  - Eine Variable, z.B. x, y, z
  - Die Negation einer Booleschen Funktion, z.B. $\neg$ F(x,y,z)
  - Die Disjunktion zweier Booleschen Funktionen, z.B. F(x,y,z) $\vee$ G(x,y,z)
  - Die Konjunktion zweier Booleschen Funktionen, z.B. F(x,y,z) $\wedge$ G(x,y,z)
- **Eine quantifizierte Boolesche Formel (QBF) besteht aus**
  - Einer Folge von Quantoren $\exists$x, $\forall$y mit daran gebundenen Variablen
  - Einer Booelschen Funktion $F(x_1, x_2, ..., x_m)$
  - Jede Variable der Funktion ist genau einmal an einem Quantor gebunden
- **Die quantifizierte Boolesche Formel ist erfüllbar falls**
  - Im Falle eines Existenzquantors: $\exists x\ F(x) \Leftrightarrow F(0) \vee F(1)$
    - wobei F eine weitere QBF sein kann
  - Im Falle eines Allquantors: $\forall x\ F(x) \Leftrightarrow F(0) \wedge F(1)$

Beispiele:

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

- $\exists x\ F(x) \Leftrightarrow F(0) \vee F(1)$, $\forall x\ F(x) \Leftrightarrow F(0) \wedge F(1)$
- $\exists x\ \forall y\ (x \wedge y) \vee (\neg x \wedge \neg y)$
- $\forall y\ \exists x\ (x \wedge y) \vee (\neg x \wedge \neg y)$

Boolesche Funktionen

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

- **Definition QBF (Quantified Boolean Formula Problem)**
  - Das Quantifizierte Boolesche Erfüllbarkeitsproblem der Booleschen Funktion ist definiert als:
  - **QBF = { ϕ | ϕ ist eine wahre quantifizierte Boolesche Formel}**
  - **Gegeben**:
    - Boolesche quantifizierte Formel Q
  - **Gesucht**:
    - Ist Q wahr?

QBF ist NP-schwierig

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

- **Spezialfall: SAT**
  - Jedes Boolesche Erfüllbarkeitsproblem ist als QBF darstellbar
- **Theorem**
  - SAT $\leq_{m,p}$ QBF
- **Beweis:**
  - Reduktionsfunktion:
    - gegeben Funktion ϕ
    - Füge für alle Variablen der Funktion Existenzquantoren hinzu
    - Ausgabe: $\exists x_1 \exists x_2 ... \exists x_m \ \phi(x_1,..,x_m)$
  - Korrektheit
    - folgt aus der Äquivalenz der Formeln
  - Laufzeit: Linear
- **Korollar:**
  - QBF ist NP-schwierig

QBF ist in PSPACE

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

- **Theorem**
  - QBF ist in PSPACE
- **Beweis:**
  - Konstruiere TM
    - gegeben QBF: $Q_1x_1\ Q_2x_2 \ldots Q_mx_m\ \phi(x_1,..,x_m)$
      - für $Q_i \in \{\forall,\exists\}$
    - Setze $x_1 = 0$: Berechne $a = Q_2x_2 \ldots Q_mx_m\ \phi(x_1,..,x_m)$
    - Setze $x_1 = 1$: Berechne $b = Q_2x_2 \ldots Q_mx_m\ \phi(x_1,..,x_m)$
    - Falls $Q_1 = \forall$
      - Falls a und b gib 1 aus, sonst 0.
    - Falls $Q_1 = \exists$
      - Falls a oder b gib 1 aus, sonst 0."
  - Platzbedarf:
    - O(1) in jeder Rekursionstiefe
    - Anzahl Rekursionstiefen: $m \leq n$
    - Gesamtspeicher: O(n)

QBF ist PSPACE-schwierig

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

- **Theorem**
  - Für alle $L \in PSPACE$ gilt $L \leq_{m,p} QBF$
- **Beweis**
  - Betrachte 1-Band s(n)-Platz-TM mit $s(n) = O(n^k)$
  - Lemma

    Es gibt eine Boolesche Funktion polynomieller Größe, die wahr ist, falls **`Erreicht-Konf`**(C,C',s(n),1) für gegebene Eingabelänge gilt, und die in Polynom-Zeit beschreibbar ist.
    - Nun ist die QBF höchstens O(s(n) log T) groß
    - Wahl $T = 2^{c\ s(n)}$ reicht für die Berechnung einer s(n)-Platz-DTM
  - Konstruktion möglich mit polynomiellem Platz.

Wie man Erreicht-Konf nicht darstellen soll

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

- **Betrachte nun folgende QBF für** `Erreicht-Konf`**(C,C',s(n),2T)**
  - ∃ Z: **`Erreicht-Konf`**(C,Z,s(n),T) ∧ **`Erreicht-Konf`**(Z,C',s(n),T)
- **Problem: Rekursiv definierte Formel wächst linear in T**
- **T wächst exponentiell in s(n)**

Wie man Erreicht-Konf darstellt

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

– Lösung:
  - ∃ Z: ∀ A,B: ((A,B)=(C,Z) ∨ (A,B)=(Z,C')) ⇒ **`Erreicht-Konf`**(A,B,s(n),T))
  - Definiere damit rekursiv die QBF

Ende der 26. Vorlesung

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

Arne Vater

Wintersemester 2006/07

25. Vorlesung

02.02.2007