Informatik III

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

Christian Schindelhauer
Wintersemester 2006/07
13. Vorlesung
07.12.2006

Überblick: Die Church-Turing-These

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

- **Turing-Maschinen**
  - 1-Band Turing-Maschine
  - Mehrband-Turing-Maschinen
- **Nichtdeterministische Turing-Maschinen**
  - Formale Definition
  - Beispiel
  - Äquivalenz zu deterministischen Turing-Maschinen
- **Aufzählbar und Abzählbar**
  - Die Aufzähler-TM
  - Abzählbar
- **Die Church-Turing-These**
  - Hilberts Problem
  - Der Begriff des Algorithmus
  - Die Church-Turing-These

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

Turing-Maschinen

Wiederholung: Turing-Maschine

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

- **Was ist eine Turing-Maschine:**
  - DFA + 1 Band

- **Zum ersten Mal:**
  - Turing-Maschinen können endlos rechnen (d.h. abstürzen)

- **Turing-Maschinen akzeptieren Sprachen,**
  - wenn sie jedes Wort der Sprache "erkennen"
  - wenn sie ansonsten irgendwas machen (außer akzeptieren)
    - d.h. z.B. verwerfen, unendlich lange rechnen, etc.

- **Turing-Maschinen entscheiden Sprachen,**
  - wenn sie auf jeder Eingabe halten und
  - akzeptieren, wenn das Wort in der Sprache ist und
  - verwerfen, wenn das Wort nicht in der Sprache ist.

Entscheidbarkeit und rekursive Aufzählbarkeit

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

- Eine Sprache L heißt **rekursiv aufzählbar**, falls es eine Turingmaschine M gibt, die L **akzeptiert**
  - Ausschließlich Ja-Antworten
  - Programmabsturz = wird als Nein interpretiert

- Eine Sprache L heißt **rekursiv oder entscheidbar**, falls es eine Turingmaschine M gibt, die L entscheidet
  - Ja oder Nein! (in einer endlichen Zeit)

Der Maschinenpark der Turingmaschinen

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

- **Keller-Automaten (PDA)**
  - NFA + Keller
- **1-Band-Turing-Maschinen (TM, DTM)**
  - DFA + Band
- **Mehr-Band-Turing-Maschinen (k-Tape-TM)**
  - DFA + Band + Band + Band
- **Nichtdeterministische Turing-Maschine (NTM)**
  - **NFA** + Band

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

Nicht-deterministische Turing-Maschinen

Die Nichtdeterministische Turing-Maschine (NTM)

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

➢ **Definition**

- Eine (**nicht-**deterministische 1-Band) Turingmaschine (NTM) wird beschrieben durch ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$.
- Dabei sind Q, Σ, Γ endliche, nichtleere Mengen
  - Q ist die Zustandsmenge,
  - Σ ist das Eingabealphabet,
  - Γ das Bandalphabet.
  - $q_0 \in Q$ ist der Startzustand.
  - $q_{accept} \in Q$ ist der akzeptierende Endzustand
  - $q_{reject} \in Q$ ist der ablehnende Endzustand
- δ **: Q** $\times \Gamma \rightarrow$ **P(Q** $\times \Gamma \times$ **{L,R}) ist die Übergangsfunktion**.
- Sie ist für kein Argument aus $\{q_{accept}, q_{reject}\} \times \Gamma$ definiert.

Konfiguration

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

- **Momentaufnahme einer NTM**
  - Bei Bandinschrift uv
    - dabei beginnt u am linken des Bandes und hinter v stehen nur Blanks
  - Zustand q,
  - Kopf auf erstem Zeichen von v

- **Konfiguration C = u*qv***

Aufeinanderfolgende Konfigurationen

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

- **Gegeben: Konfigurationen C_1, C_2**

- **Wir sagen:**
  - **Konfiguration C_1 führt zu C_2**, falls die NTM von C_1 in einem Schritt zu C_2 übergehen kann.

- **Formal:**
  - Seien a, b, c $\in \Gamma$, u, v $\in \Gamma^*$ und Zustände q_i , q_j gegeben
- **Wir sagen**
  - **u a q_i b v** führt zu **u q_j a c v**,
    - falls $(q_j,c,L) \in \delta(q_i,b)$ und
  - **u a q_i b v** führt zu **u a c q_j v**,
    - falls $(q_j,c,R) \in \delta(q_i,b)$

Akzeptanz einer NTM

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

Eine Nichtdeterministische Turingmaschine M **akzeptiert** eine Eingabe w, falls es eine Folge von Konfigurationen C_1, C_2, …, C_k gibt, so dass

1. C_1 ist die Startkonfiguration von M bei Eingabe w
2. C_i kann zu C_{i+1} überführt werden
3. C_k ist eine akzeptierende Konfiguration

- Die von M akzeptierten Worte bilden die **von M akzeptierte Sprache** L(M)

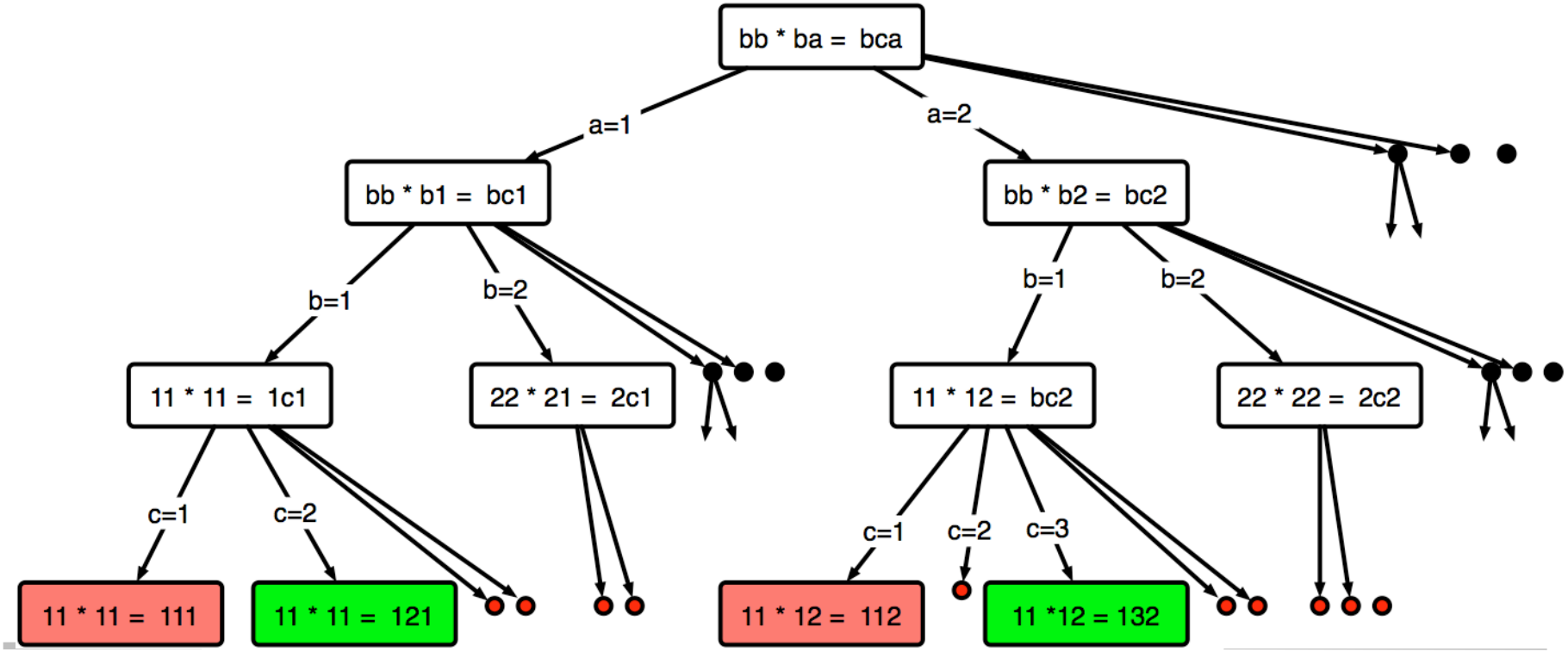
- Eine NTM **entscheidet** eine Sprache, wenn jede Eingabe zu einem endlichen Berechnungsbaum der Konfigurationen führt.

Die Nützlichkeit einer NTM

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

- **NTMs können "raten"**
- **Beispielproblem:**
  - Gibt es eine Lösung für das Zahlenrätsel:
    - bb * ba = bca
    - wobei jeder Buchstabe für eine Ziffer steht

- **NTM:**
  - Für alle Buchstabe
    - "rate" eine Möglichkeit
    - setze Buchstaben ein
  - Verifiziere Gleichung



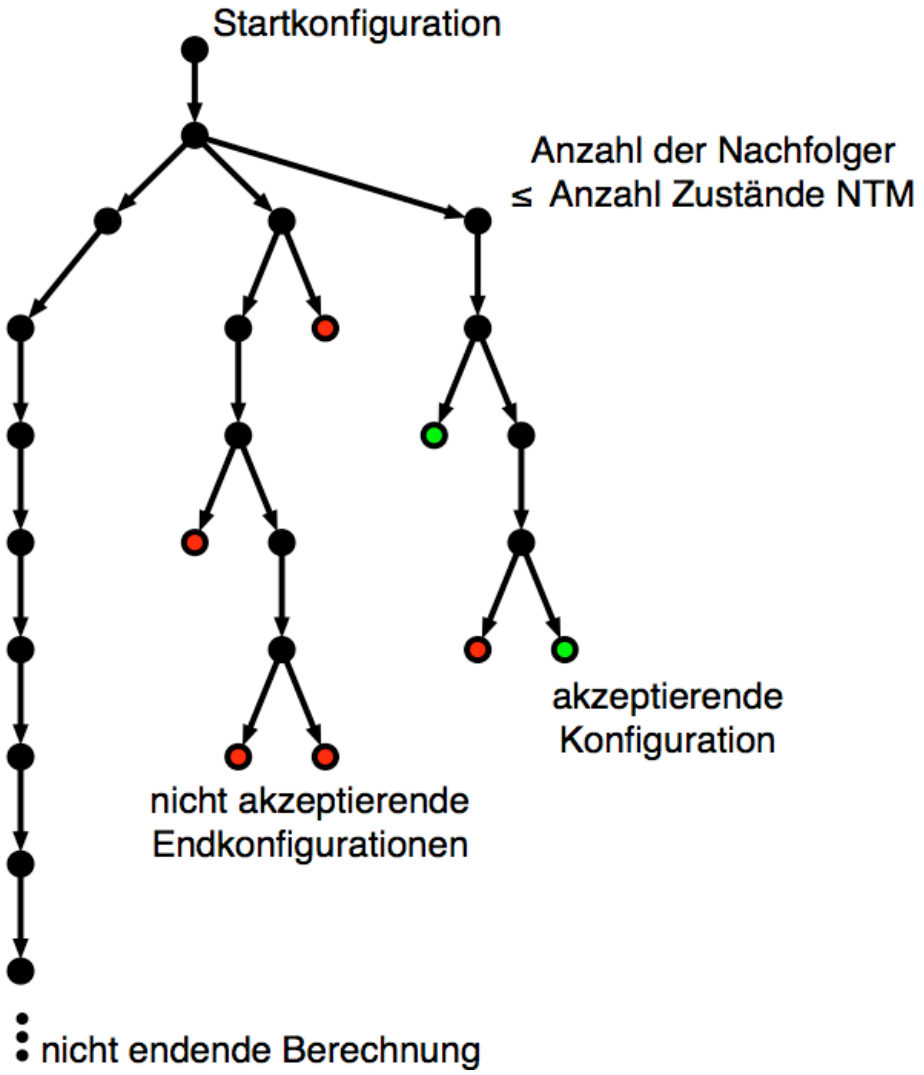
Konfigurationen einer NTM

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

- **Startkonfiguration: q_0w, wobei w die Eingabe ist**
- **Akzeptierende Konfiguration: Konfigurationen mit Zustand q_{accept}**
- **Ablehnende Konfiguration: Konfigurationen mit Zustand q_{reject}**
- **Haltende Konfiguration: akzeptierende oder ablehnende Konfigurationen**

Berechnungsbaum einer NTM

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



Äquivalenz von DTM und NTM

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

Theorem

Zu jeder nichtdeterminischen Turingmaschine M gibt es eine äquivalente deterministische Turingmaschine D, d.h.

- Falls M auf Eingabe x akzeptiert, dann akzeptiert D auf x und hält
- Falls M auf x nicht akzeptiert, dann akzeptiert D nicht
- Falls M auf allen Berechnungspfaden hält, dann hält auch D

Äquivalenz von DTM und NTM

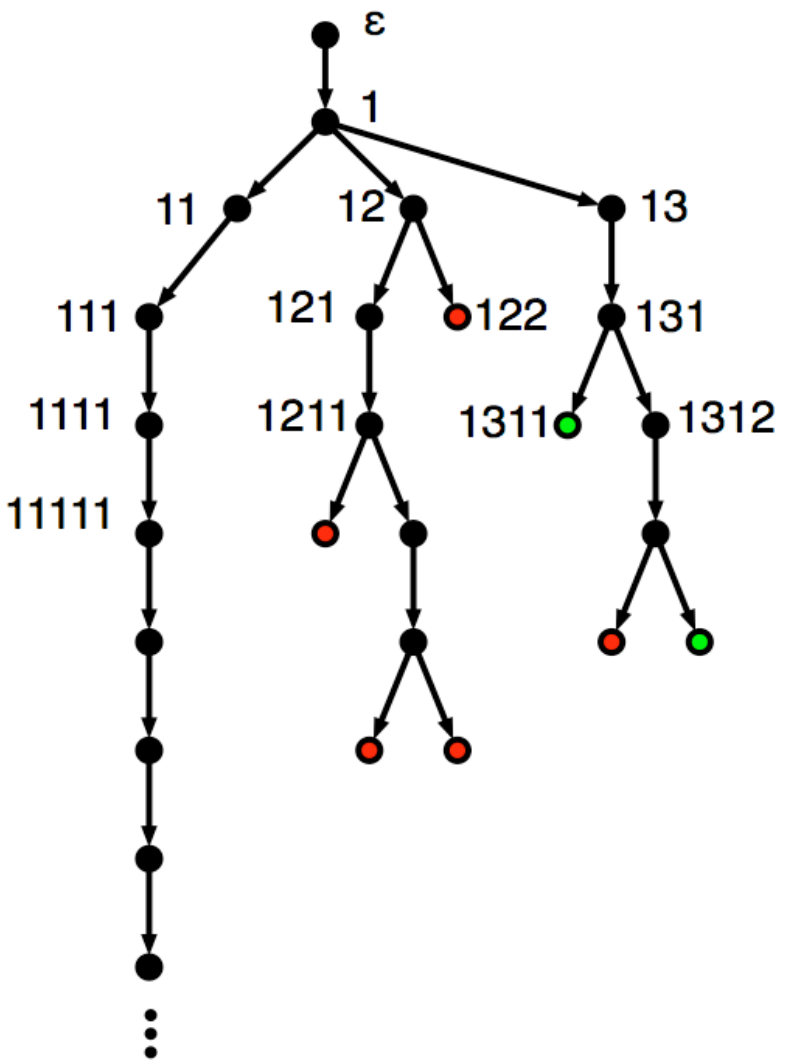
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Theorem

Zu jeder nichtdeterminischen Turingmaschine M gibt es eine äquivalente deterministische Turingmaschine D.

Beweisidee:

- Simuliere Berechnungsbaum
  - ausgehend von Startkonfiguration
- Verwende BFS (breadth first search)
  - DFS nicht möglich
- DTM D:
  - Drei Bänder:
    - Eingabeband
    - Simulationsband
    - Adressband

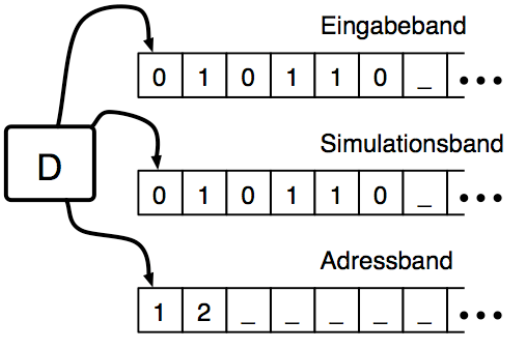


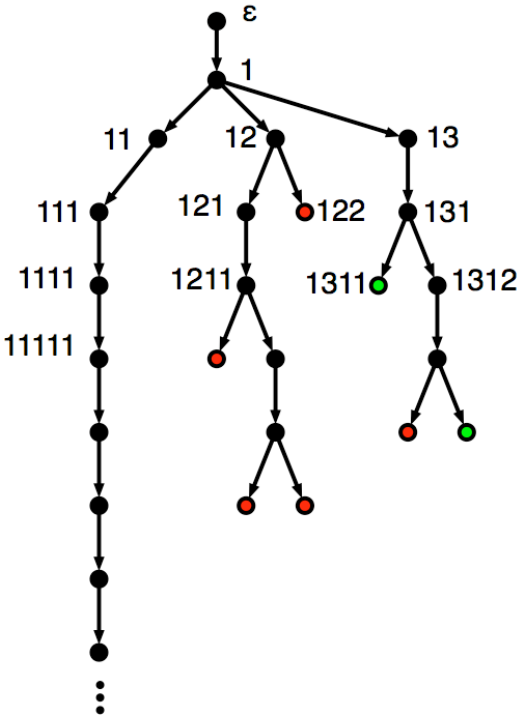
Äquivalenz von DTM und NTM

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

Beweisidee:

- Simulator DTM D:
  - Drei Bänder:
    - Eingabeband
    - Simulationsband
    - Adressband
- Zähle alle Knoten des Berechnungsbaums im Adressband auf. Sortierung:
  - zuerst Länge
  - dann lexikographisch
    - ε,
    - 1,2,3,
    - 11,12,13, 21,22,23, 31,32,33,
    - 111,112,...
- Führe die (nun deterministische) entsprechenden Berechnung auf dem entsprechenden Berechnungspfad auf dem Simulationsband durch





NTMs und DTMs sind äquivalent Beweis (1.)

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

➢ **Beweis:**

- Die simulierende DTM hat drei Bänder
  - Nach Theorem 10. Vorlesung sind 3-Band DTMs und DTMs äquivalent
- Auf Band 1 wird das Eingabewort geschrieben.
  - Dieses Band bleibt während der ganzen Berechnung unverändert
- Auf Band 2 wird eine Kopie des Bandes der NTM M in einer Konfiguration im Berechnungsbaums gespeichert
- Auf Band 3 wird die Position der NTM M im Berechnungsbaum nachverfolgt

➢ **Band 3:**

- Jeder Knoten im Berechnungsbaum kann höchstens b=|Q| Nachfolger haben,
  - wobei Q die Zustandsmenge der NTM M ist
- Wir betrachten daher das Alphabet {1,2,..,b}
- Die Adresse 231 steht dann für die 2. Möglichkeit des Berechnungspfades ab der Startkonfiguration
  - dessen 3. Kindes
    - und dessen 1. Kindes
- Die Kopfposition auf dem Band beschreibt die Tiefe im Berechnungsbaumes während der Simulation der NTM
- Das leere Wort beschreibt die Startkonfiguration

NTMs und DTMs sind äquivalent Beweis (2.)

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

- **Beweis (Fortsetzung):**
  - Die DTM D arbeitet wie folgt:

1. Band 1 wird mit der Eingabe w beschrieben Band 2 und Band 3 sind leer
2. Kopiere Band 1 auf Band 2
3. Benutze Band 2 um einen Schritt der nichtdeterministischen Berechnung der NTM M zu berechnen.
   Hierzu wird die x. Möglichkeit durchgeführt,
   - wobei x das aktuelle Zeichen auf Band 3 ist.

   Ist x = "_", wird keine Berechnung durchgeführt und überprüft, ob M im akzeptierenden Zustand ist.
   - Falls ja, akzeptiert D und hält

   Gibt es keine x. Möglichkeit oder ist der Zustand verwerfend gehe zu Schritt 4

   Ansonsten bewege den Kopf auf Band 3 nach rechts und wiederhole Schritt 3.

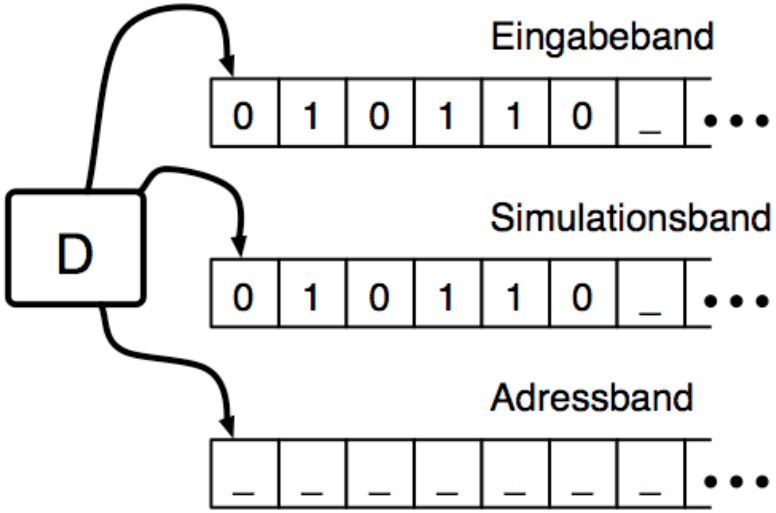
4. Ersetze die Zeichenkette auf Band 3 durch die lexikographisch nächste Zeichenkette gleicher Länge.

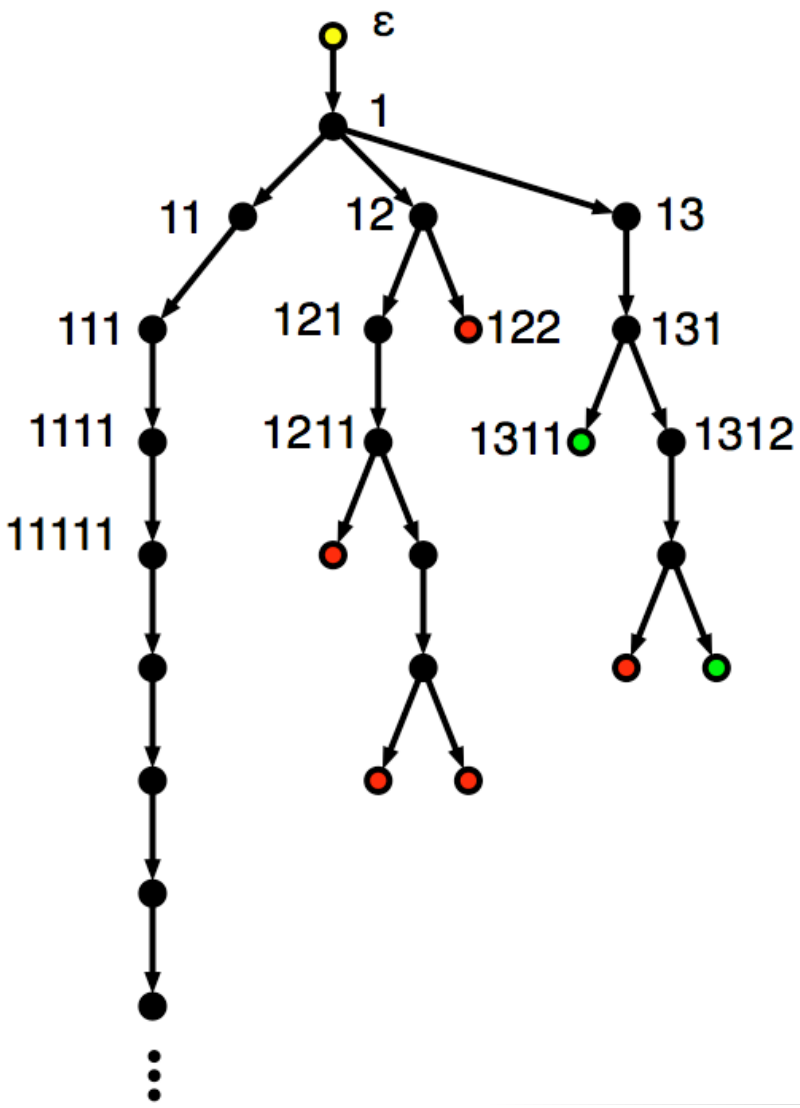
Gibt es keine dieser Länge mehr, ersetze sie durch die lexikographisch erste Zeichenkette der nächstgrößeren Länge.

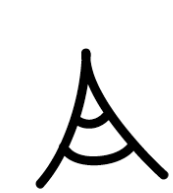
- Die hier beschriebene DTM erfüllt die gewünschte Eigenschaft.

Simulation einer NTM: Beispiel

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

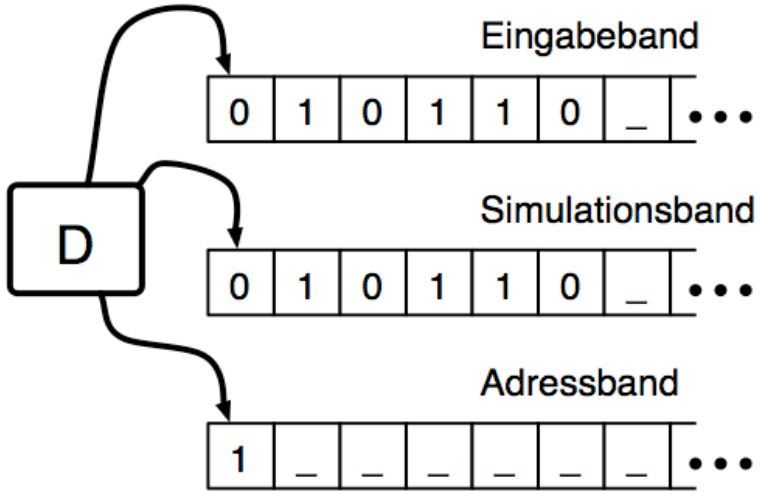


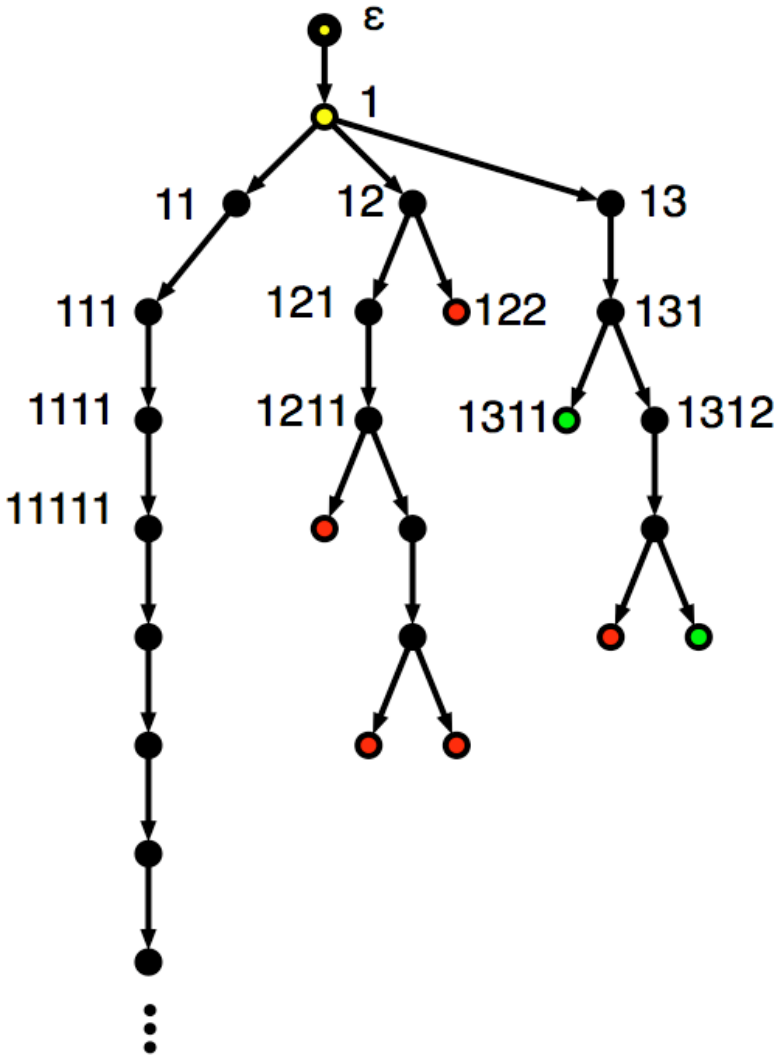




Simulation einer NTM: Beispiel

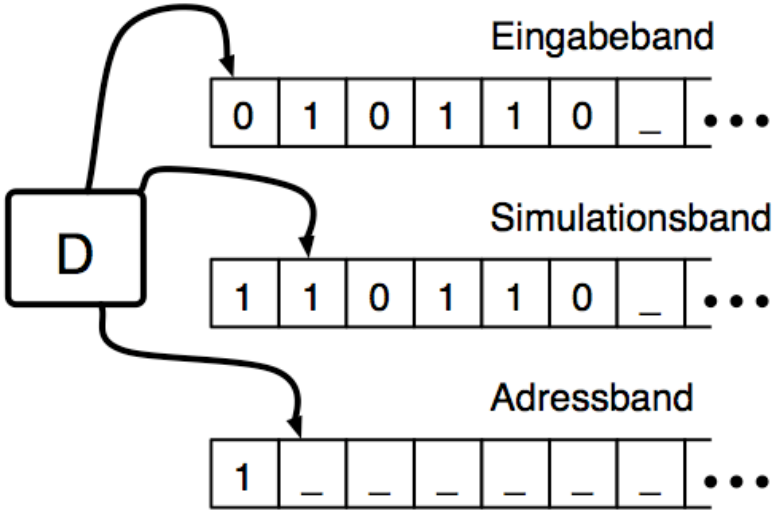
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

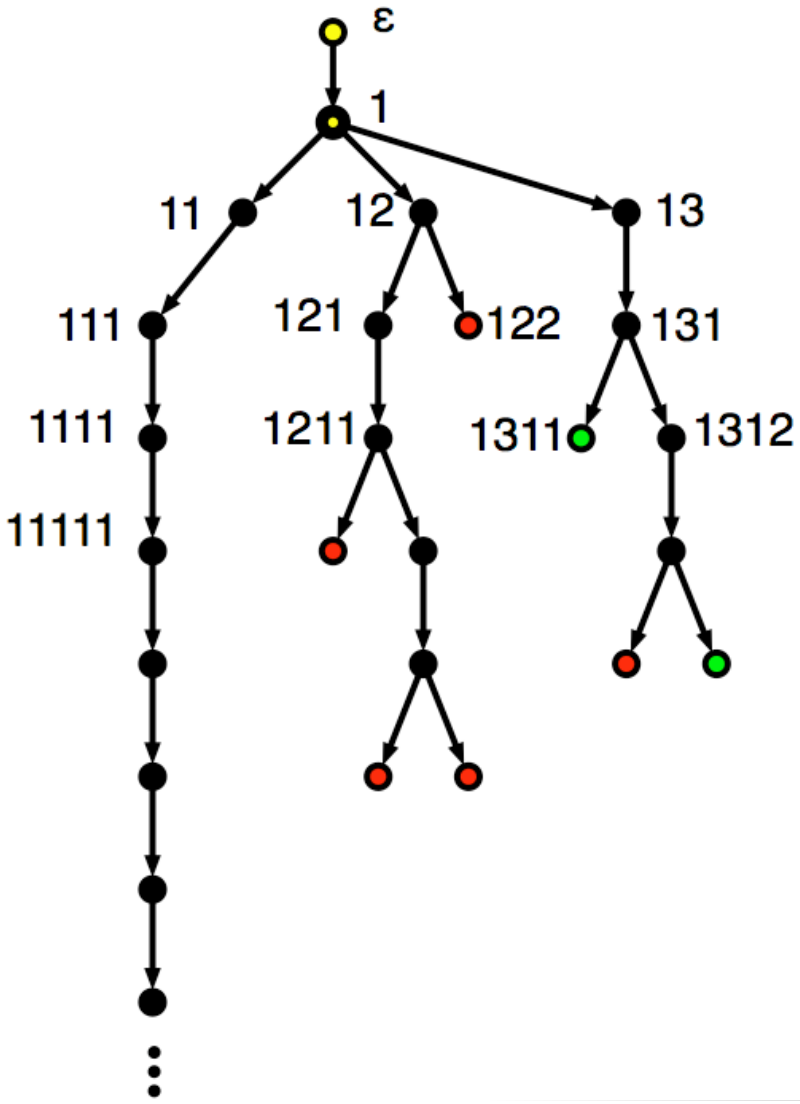


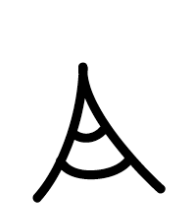


Simulation einer NTM: Beispiel

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

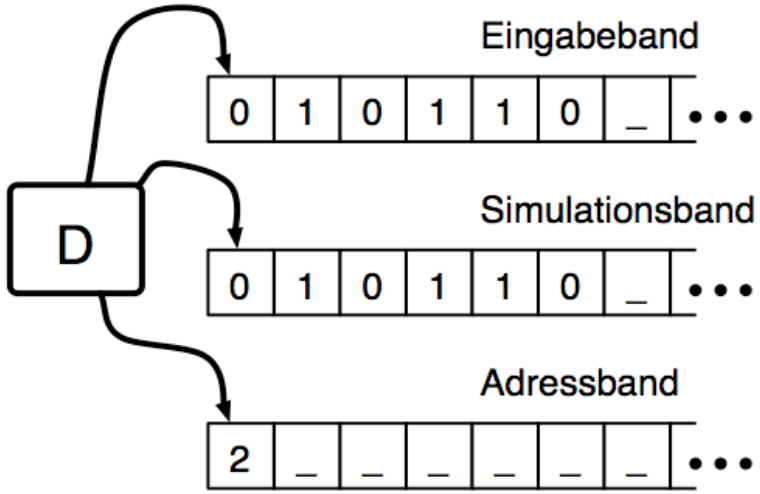


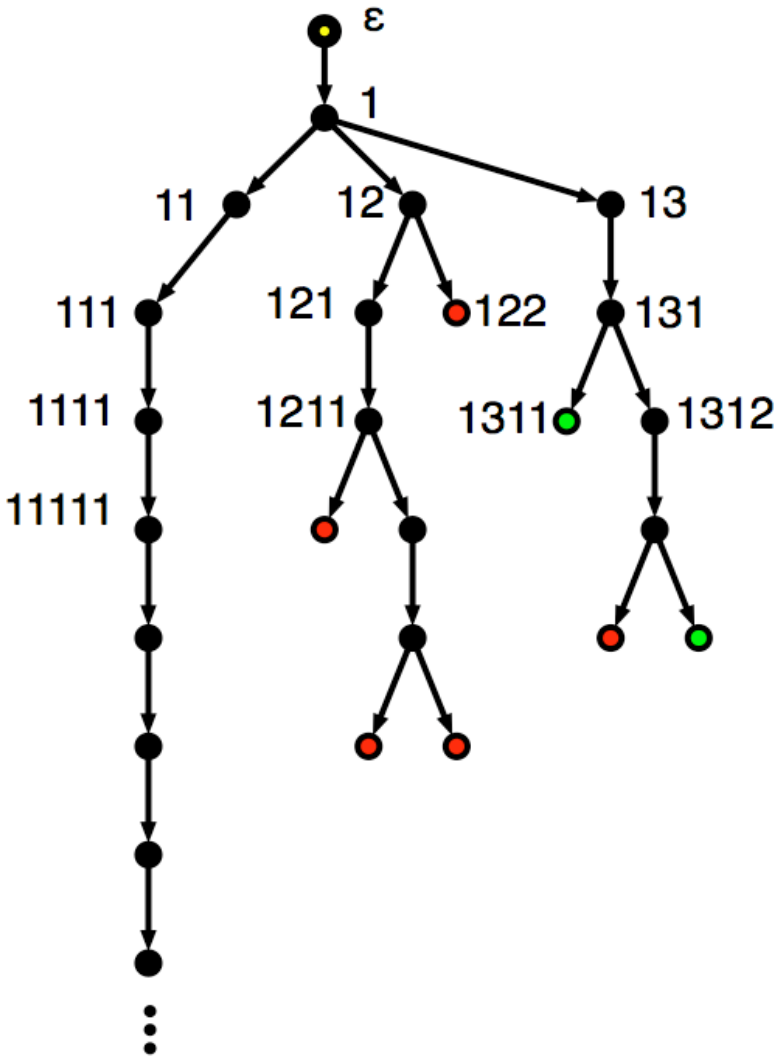




Simulation einer NTM: Beispiel

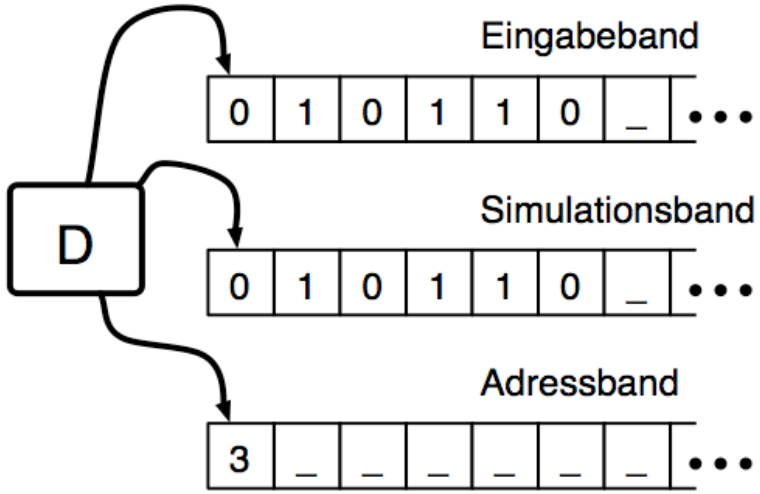
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

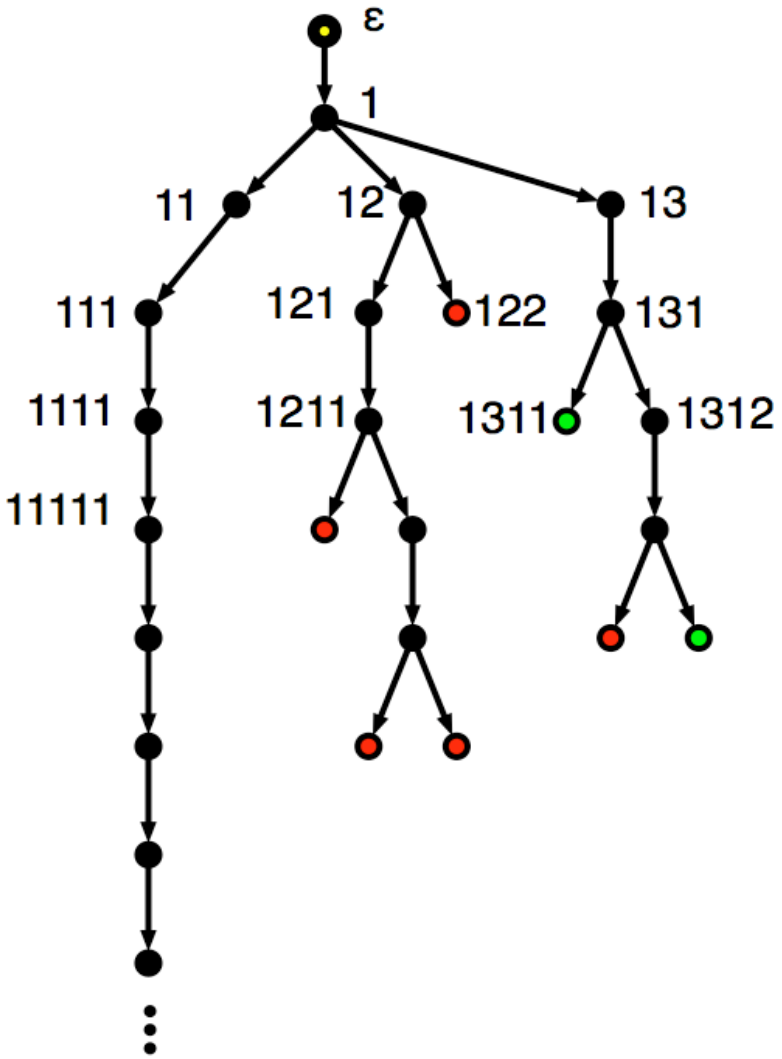


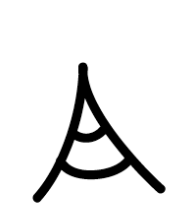


Simulation einer NTM: Beispiel

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

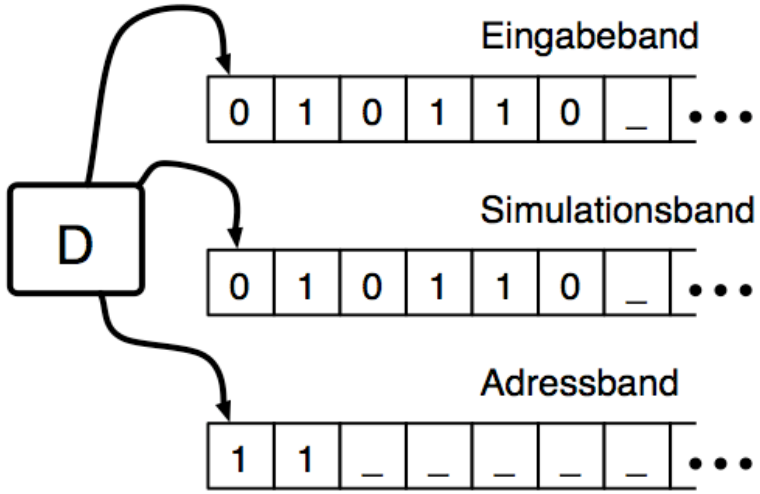


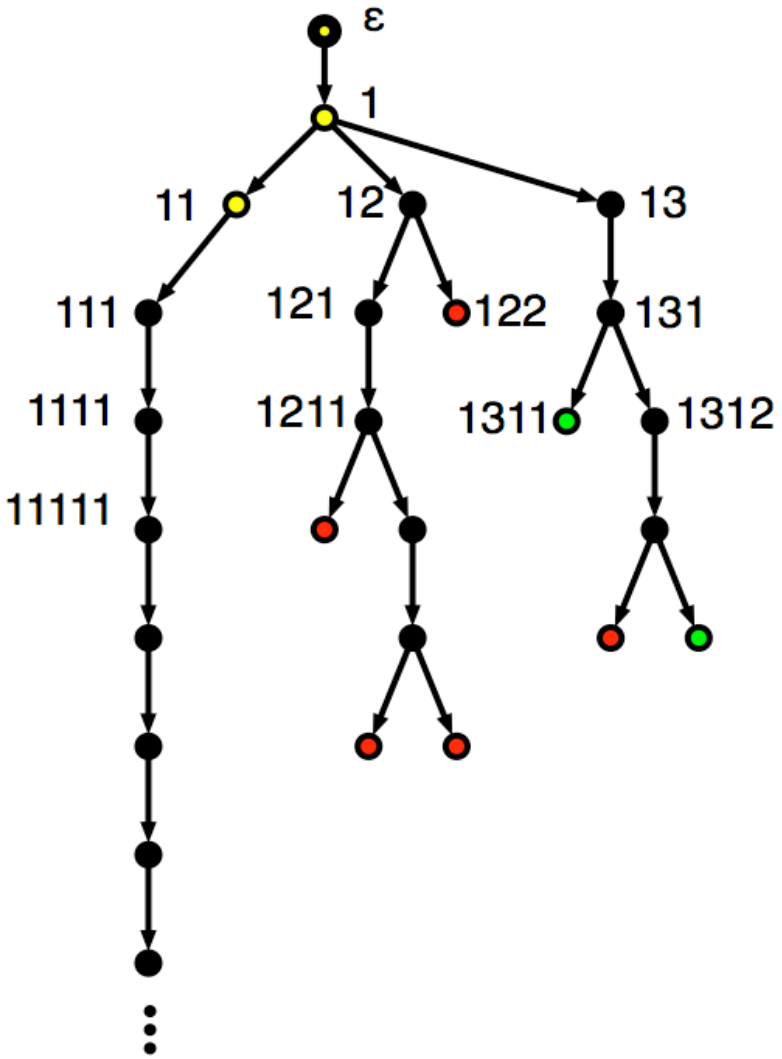




Simulation einer NTM: Beispiel

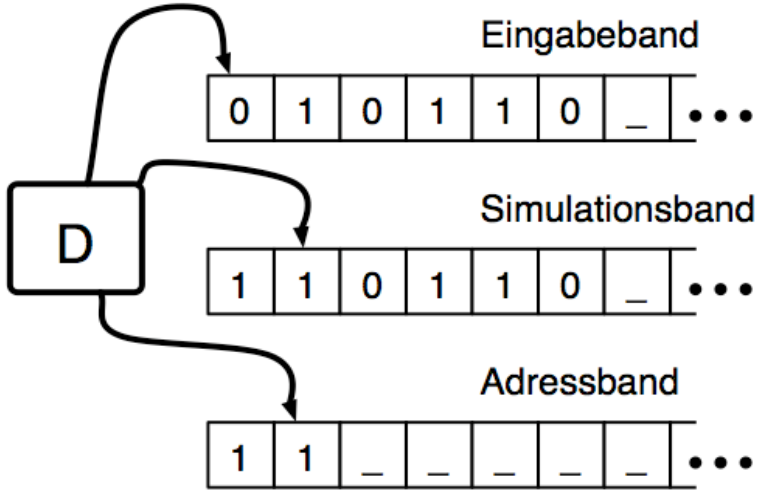
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

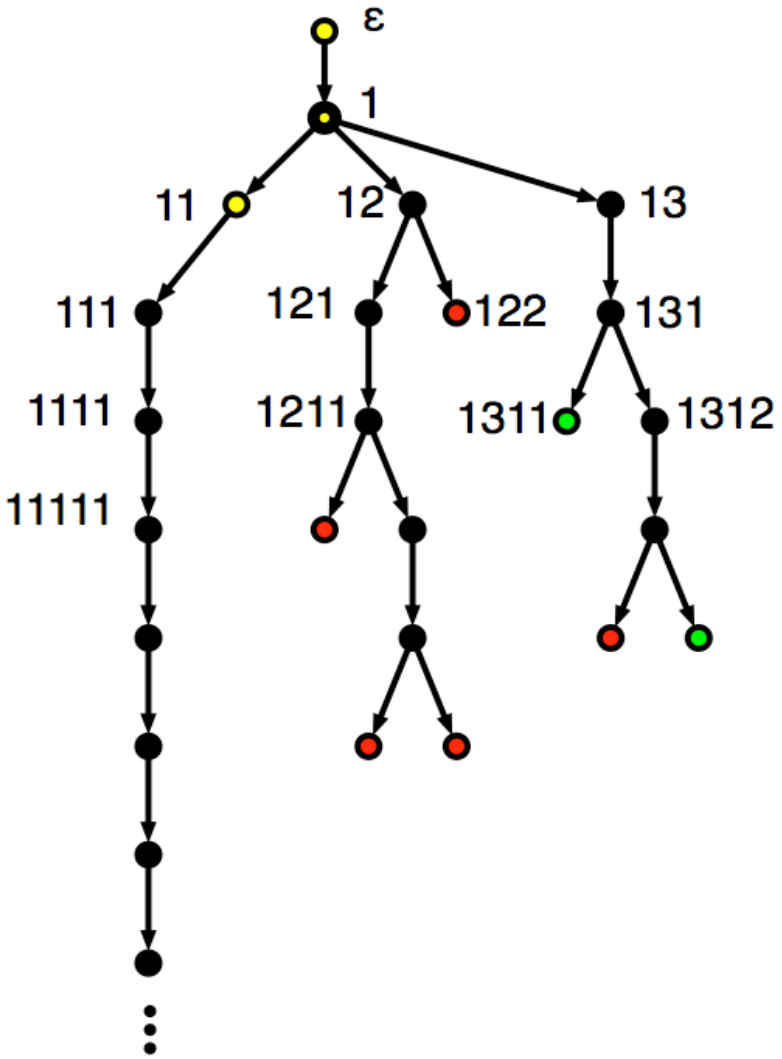




Simulation einer NTM: Beispiel

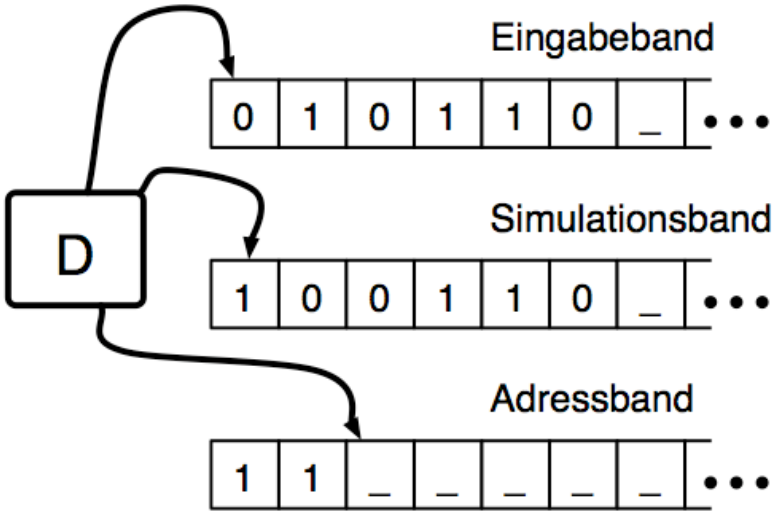
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

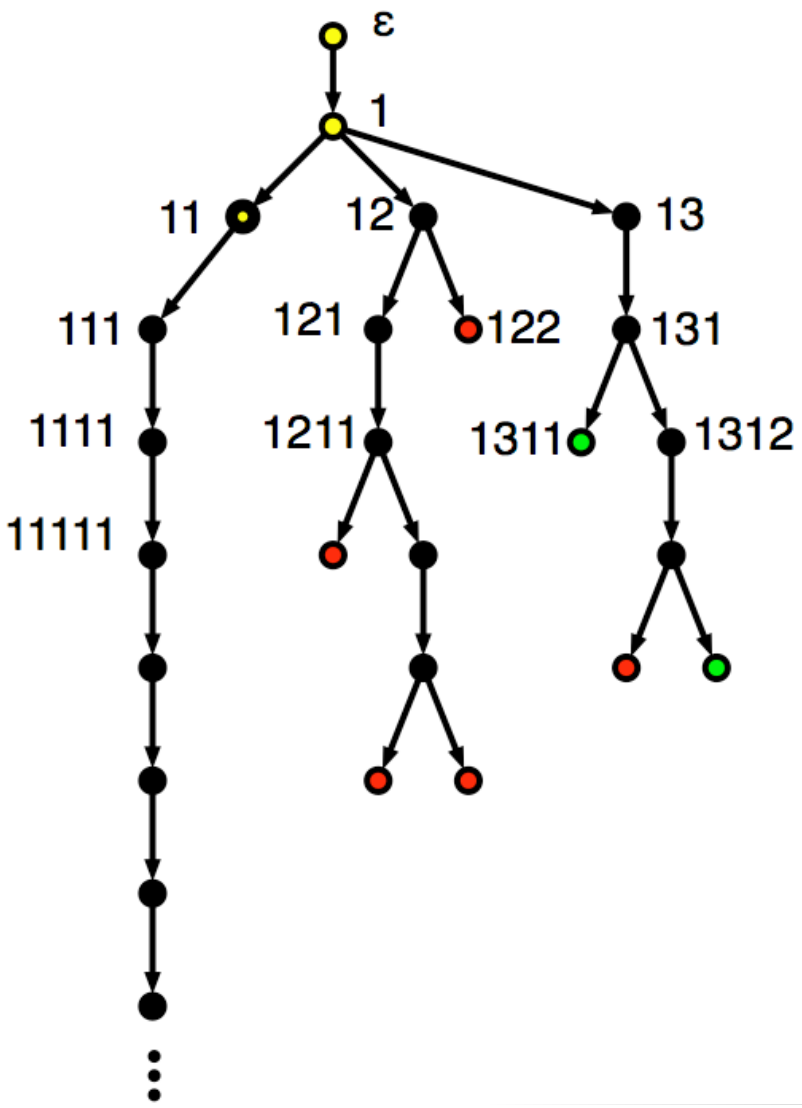


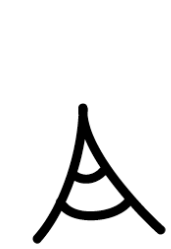


Simulation einer NTM: Beispiel

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

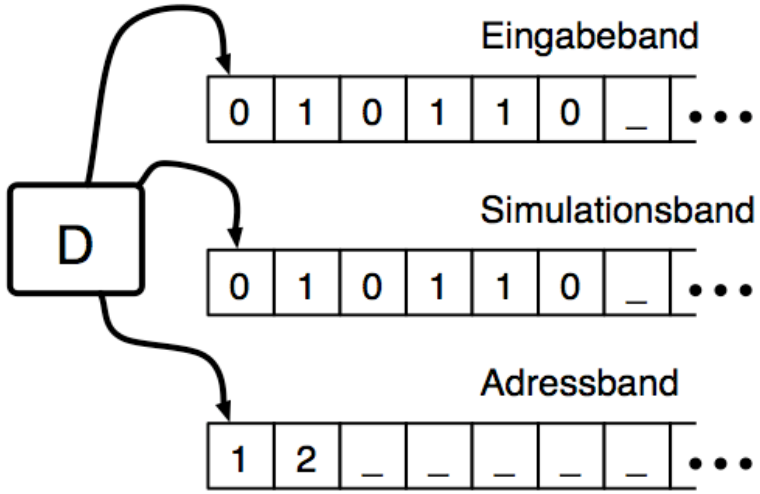


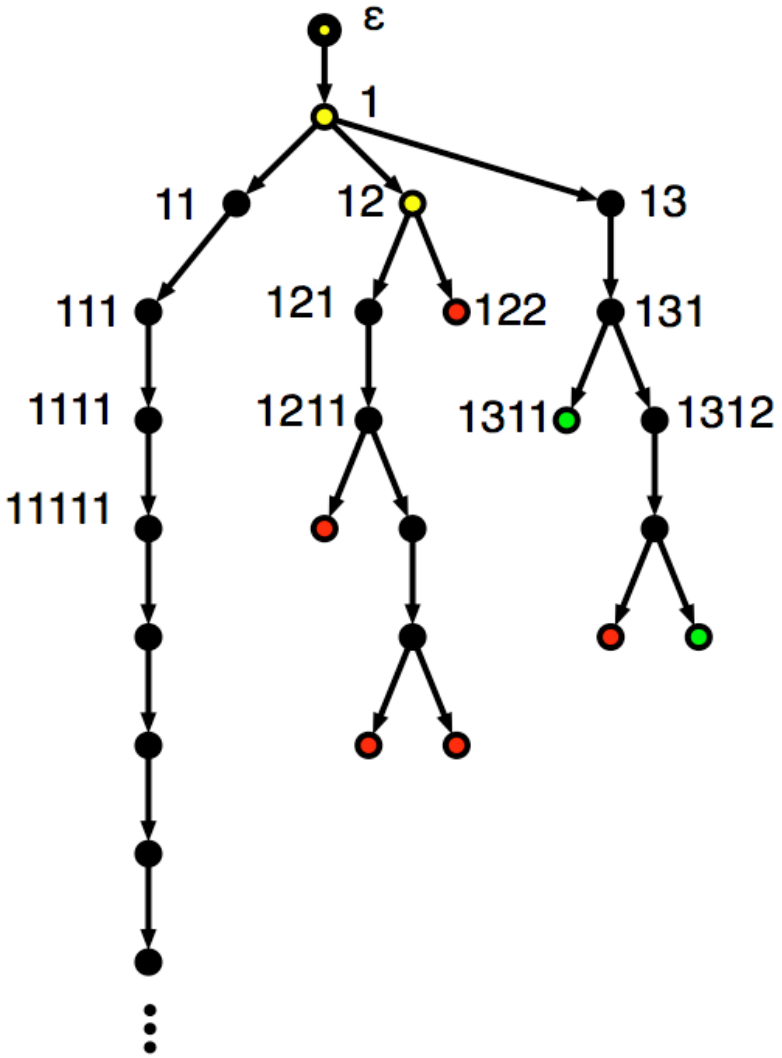




Simulation einer NTM: Beispiel

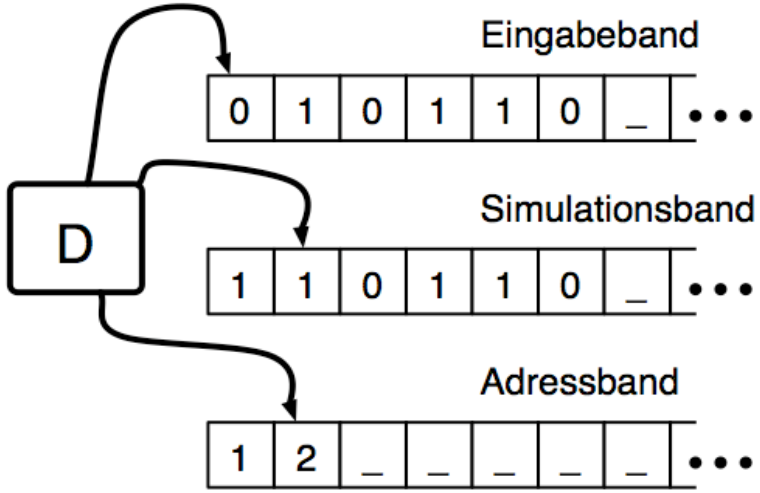
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

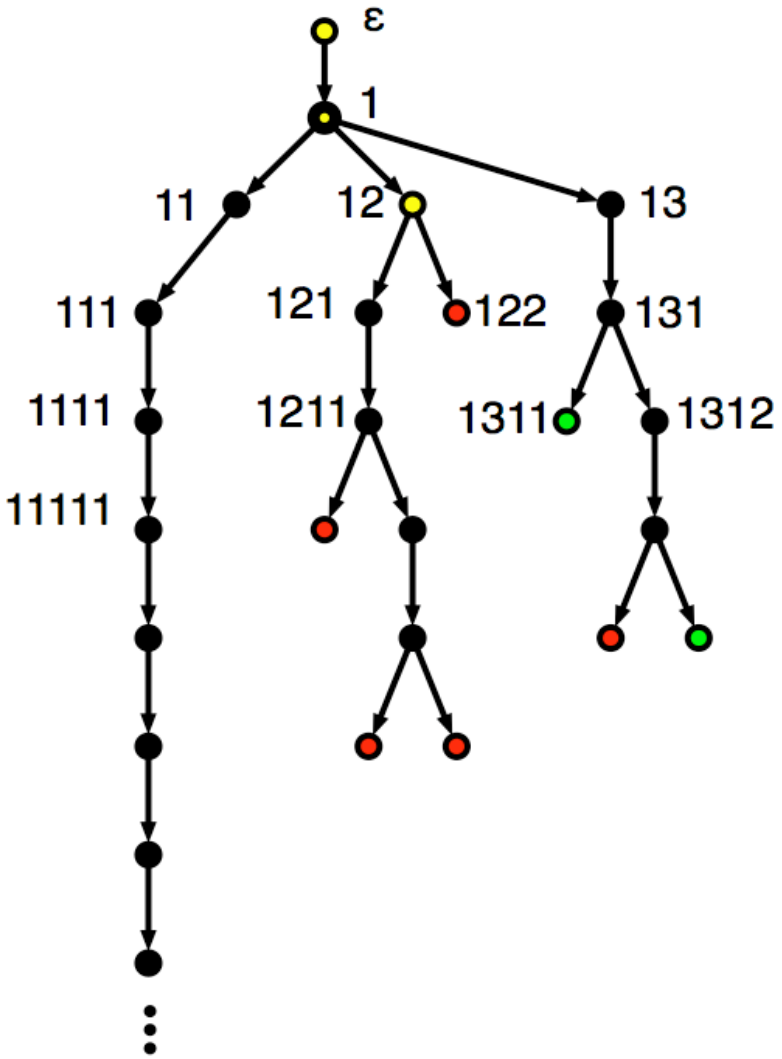


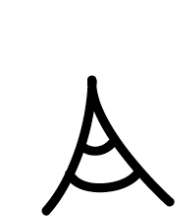


Simulation einer NTM: Beispiel

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

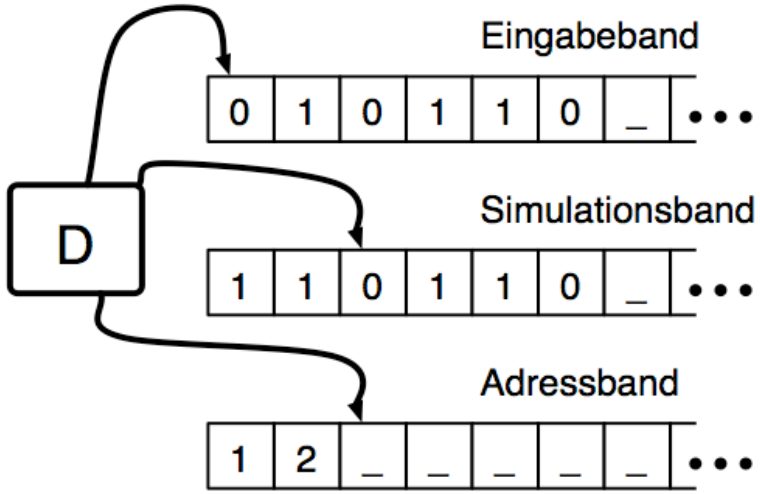


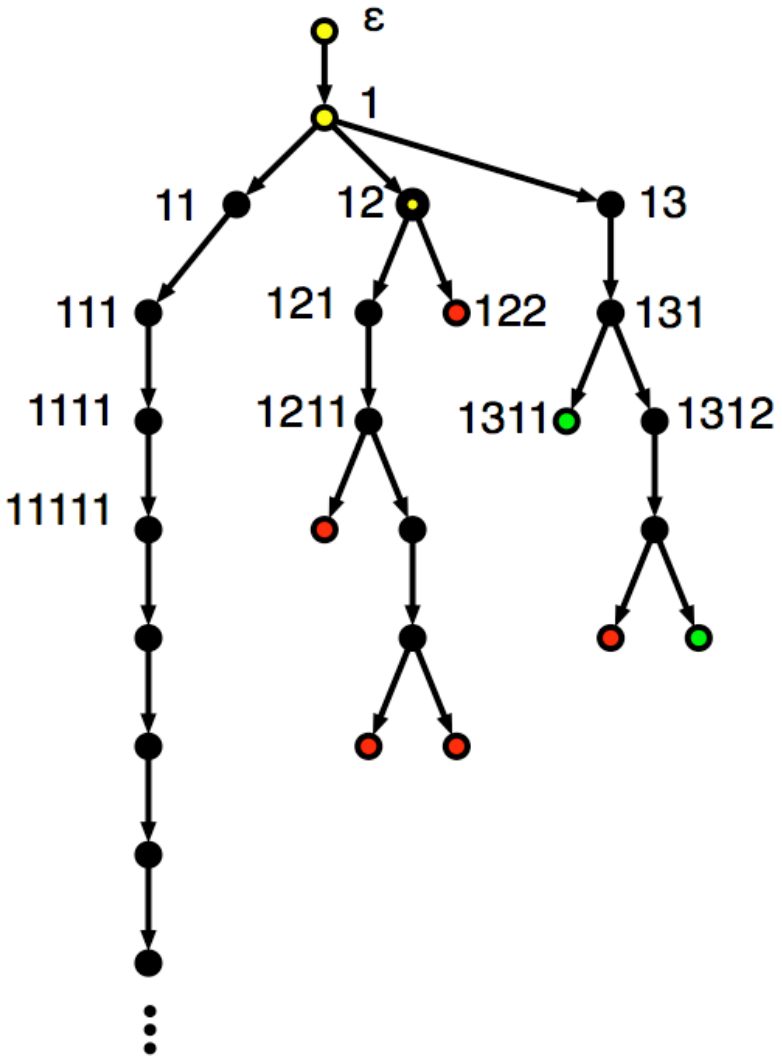




Simulation einer NTM: Beispiel

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





Schlussfolgerungen

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

- **Korollar:**
  - Eine Sprache ist genau dann rekursiv aufzählbar, wenn sie es eine nichtdeterministische Turing-Maschine gibt, die jedes Wort der Sprache akzeptiert.
  - Eine Sprache ist genau dann entscheidbar, wenn eine nichtdeterministische Turingmaschine sie entscheidet.

Reduktionen

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

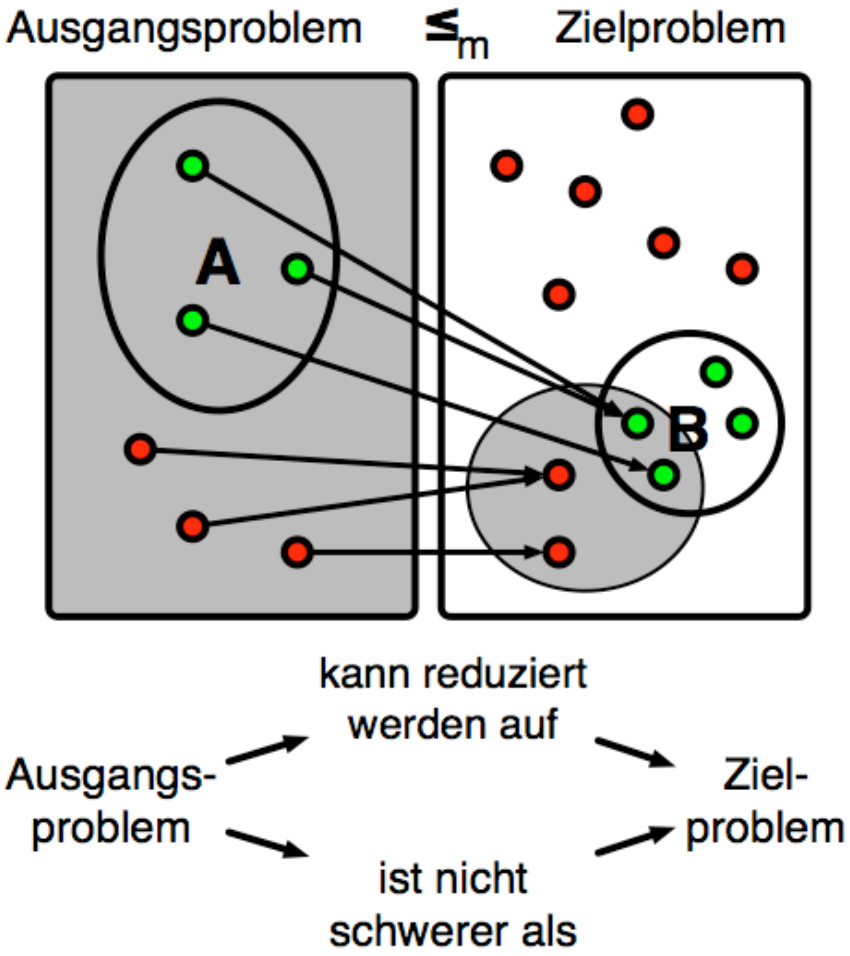
- **Unentscheidbare Probleme**
  - Das Halteproblem
  - Das Leerheitsproblem einer Turingmaschine
- **Ein einfaches nicht berechenbares Problem**
  - Das Postsche Korrespondenzproblem
- **Abbildungsreduktionen**
  - Definition
  - Anwendungen
  - Äquivalenzproblem zweier Turingmaschen
  - Der Satz von Rice
- **Turing-Reduktionen**

Wiederholung: Abbildungsreduktion

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

- **Definition**
  - Eine Funktion f: $\Sigma^* \to \Sigma^*$ ist **berechenbar,** falls eine Turing-Maschine für jede Eingabe w mit dem Ergebnis f(w) auf dem Band hält

- **Definition (Abbildungsreduktion, Mapping Reduction, Many-one)**
  - Eine Sprache A ist kann durch Abbildung auf eine Sprache B reduziert werden: $\mathbf{A \leq_m B}$,
    - falls es eine 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.



Reduktionen und Rekursive Aufzählbarkeit

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

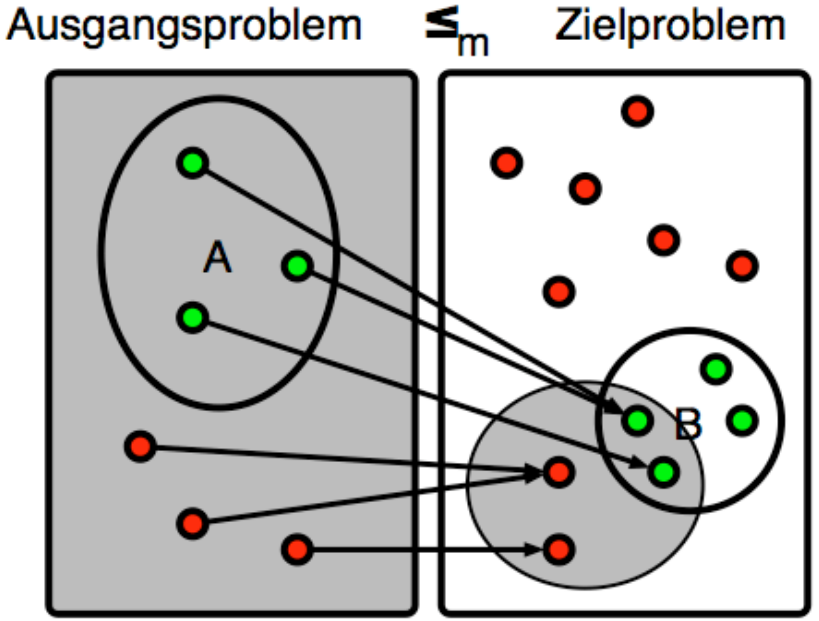
- **Theorem**
  - Falls $A \leq_m B$ und B ist rekursiv aufzählbar, dann ist A rekursiv aufzählbar.
- **Beweis**
  - Sei M, eine Turing-Maschine, die B akzeptiert.
  - Betrachte die Akzeptor-TM N:
  - N = “Auf Eingabe w:
    - Berechne f(w)
    - Führe die Berechnung von M auf Eingabe f(w) durch
    - N gibt aus, was M ausgibt”
  - Falls $f(w) \in B$,
    - dann akzeptiert M
    - dann ist auch $w \in A$
  - Falls $f(w) \notin B$,
    - dann akzeptiert M nicht
    - dann ist auch $w \notin A$

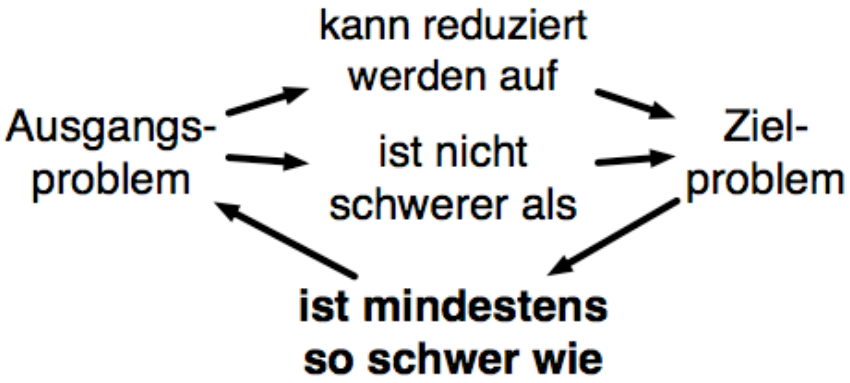
Nicht-Rekursive Aufzählbarkeit und Reduktionen

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

- **Theorem**
  - Falls $A \leq_m B$ und B ist rekursiv aufzählbar, dann ist A rekursiv aufzählbar.

- **Korollar**
  - Falls $A \leq_m B$ und A ist nicht rekursiv aufzählbar, dann ist B nicht rekursiv aufzählbar.

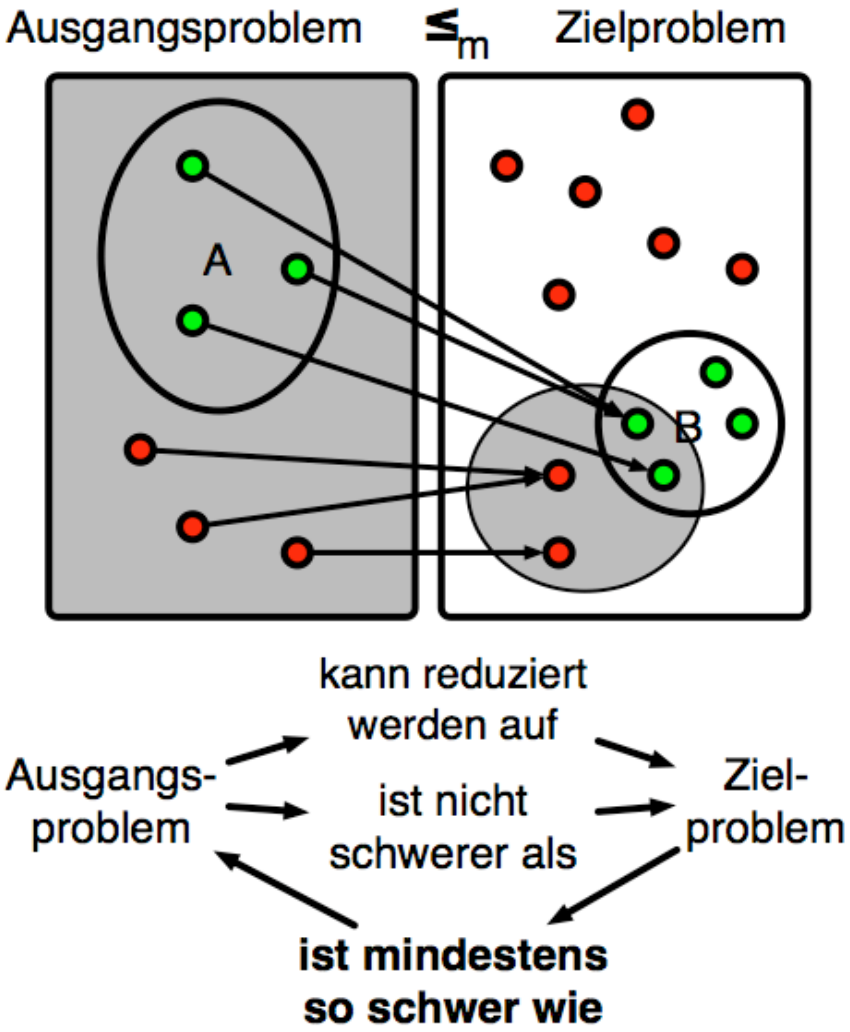




Zusammenfassung: Abbildungsreduktionen

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

- **Eine Sprache A ist kann durch Abbildung auf eine Sprache B reduziert werden:** $A \leq_m B$**,**
  - falls es eine 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.
- **Theorem**
  - Falls $A \leq_m B$ und B ist entscheidbar, dann ist A entscheidbar.
- **Korollar**
  - Falls $A \leq_m B$ und A ist nicht entscheidbar, dann ist B auch nicht entscheidbar.
- **Theorem**
  - Falls $A \leq_m B$ und B ist rekursiv aufzählbar, dann ist A rekursiv aufzählbar.
- **Korollar**
  - Falls $A \leq_m B$ und A ist nicht rekursiv aufzählbar, dann ist B nicht rekursiv aufzählbar.



Ein nicht rekursiv aufzählbares und nicht rekursiv ko-aufzählbares Problem

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

- **Definition**
  - Das TM-Äquivalenzproblem
    - Gegeben: TM M_1 und TM M_2
    - Gesucht: Ist $L(M_1) = L(M_2)$?
  - Definition als Sprache:
    - $\mathrm{EQ}_{\mathrm{TM}} = \{\langle M_1, M_2\rangle \mid M_1, M_2 \text{ sind TMs und } L(M_1) = L(M_2)\}$
- **Theorem**
  - $\mathrm{EQ}_{\mathrm{TM}}$ ist weder rekursiv aufzählbar noch rekursiv ko-aufzählbar.
- **Beweisidee:**
  - Reduktion: $A_{TM} \leq_m EQ_{TM}$
    - äquivalent zu $\overline{A_{TM}} \leq_m \overline{EQ_{TM}}$
    - beweist, dass $\overline{EQ_{TM}}$ nicht rekursiv aufzählbar ist.
  - Reduktion: $A_{TM} \leq_m EQ_{TM}$
    - äquivalent zu $A_{TM} \leq_m \overline{EQ_{TM}}$
    - beweist, dass $\overline{EQ_{TM}}$ nicht rekursiv ko-aufzählbar ist.

$\overline{A_{TM}} \leq_m EQ_{TM}$

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

- **Reduktionsfunktion: F**
- **F** = "Auf Eingabe <M,w>, wobei M eine TM ist und w ein Wort
  - Konstruiere Maschinen M_1 und M_2 wie folgt
    - M_1 = "Für jede Eingabe:
      - Verwerfe"
    - M_2 = "Für jede Eingabe:
      - Führe M auf w aus
      - Falls M akzeptiert, akzeptiert M_2"
  - F gibt $<M_1, M_2>$ aus"
- **Zu beweisen:**
  - **F ist berechenbar**
    - die Kodierung der TM kann automatisch erfolgen
  - **$<M,w> \in A_{TM} \Leftrightarrow F(<M,w>) \in \overline{EQ_{TM}}$**

- **M ist TM und $F(<M,w>) = <M_1, M_2>$**
  - wobei $L(M_1) = \emptyset$ und
  - $L(M_2) = \Sigma^*$, falls M(w) akzeptiert
  - $L(M_2) = \emptyset$, falls M(w) nicht akzeptiert

- **Daraus folgt:**
  - $<M,w> \in A_{TM} \Leftrightarrow F(<M,w>) \notin EQ_{TM}$

$A_{TM} \leq_m EQ_{TM}$

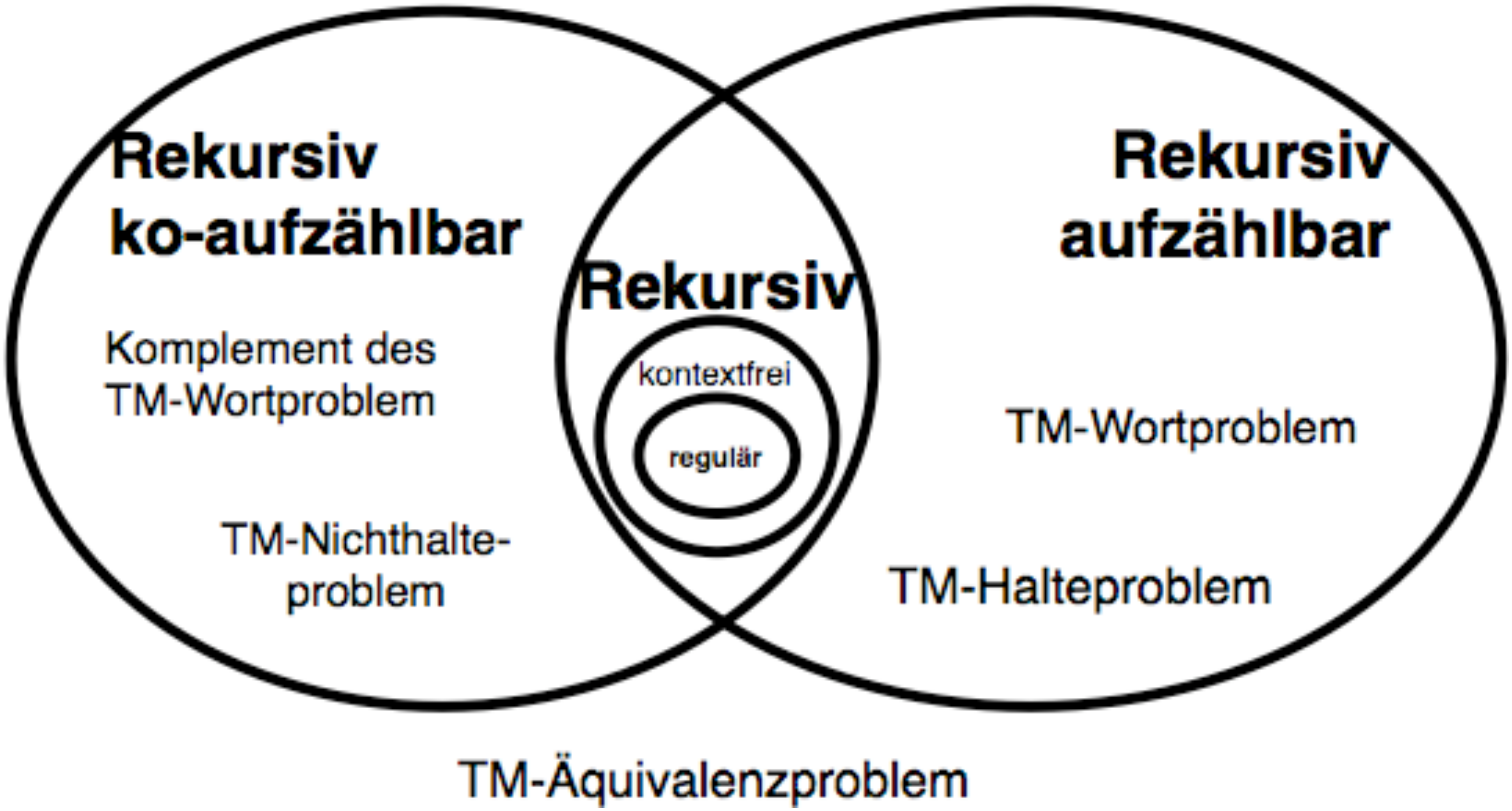
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Reduktionsfunktion: F**
- **F** = "Auf Eingabe <M,w>, wobei M eine TM ist und w ein Wort
  - Konstruiere Maschinen M_1 und M_2 wie folgt
    - M_1 = "Für jede Eingabe:
      - Akzeptiere"
    - M_2 = "Für jede Eingabe:
      - Führe M auf w aus
      - Falls M akzeptiert, akzeptiert M_2"
  - F gibt $<M_1, M_2>$ aus"
- **Zu beweisen:**
  - **F ist berechenbar**
    - die Kodierung der TM kann automatisch erfolgen
  - **$<M,w> \in A_{TM} \Leftrightarrow F(<M,w>) \in EQ_{TM}$**

- **M ist TM und $F(<M,w>) = <M_1, M_2>$**
  - wobei $L(M_1) = \Sigma^*$ und
  - $L(M_2) = \Sigma^*$, falls M(w) akzeptiert
  - $L(M_2) = \varnothing$, falls M(w) nicht akzeptiert

- **Daraus folgt:**
  - $<M,w> \in A_{TM} \Leftrightarrow F(<M,w>) \in EQ_{TM}$

Überblick


Rekursiv
ko-aufzählbar
Komplement des
TM-Wortproblem
TM-Nichthalte-
problem
Rekursiv
kontextfrei
regulär
Rekursiv
aufzählbar
TM-Wortproblem
TM-Halteproblem
TM-Äquivalenzproblem

Der Satz von Rice

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

- **Jede Menge von Turing-Maschinen, die über eine funktionale Eigenschaft definiert werden, ist nicht entscheidbar.**

- **Theorem**
  - Sei $K \subseteq \mathbf{P}(\Sigma^*)$ eine nicht triviale Klasse von rekursiv aufzählbaren Sprachen, d.h.
    - K ist nicht leer und
    - K beinhaltet nicht alle rekursiv aufzählbare Sprachen
  - Dann ist die folgende Sprache nicht entscheidbar

$$L_K = \{\langle M \rangle \mid M \text{ ist eine TM und } L(M) \in K\}$$

- **Beispiele**

$$L_1 = \{\langle M \rangle \mid M \text{ ist eine TM, die eine reguläre Sprache akzeptiert}\}$$

$$L_2 = \{\langle M \rangle \mid M \text{ ist eine TM, welche keine Eingabe akzeptiert}\}$$

$$L_3 = \{\langle M \rangle \mid M \text{ ist eine TM, so dass } M(10) = 1\}$$

Ende der 13. Vorlesung

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

Informatik III
Armer Vater
07.12.2006