Informatik III

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

Christian Schindelhauer

Wintersemester 2006/07

14. Vorlesung

08.12.2006

Reduktionen

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

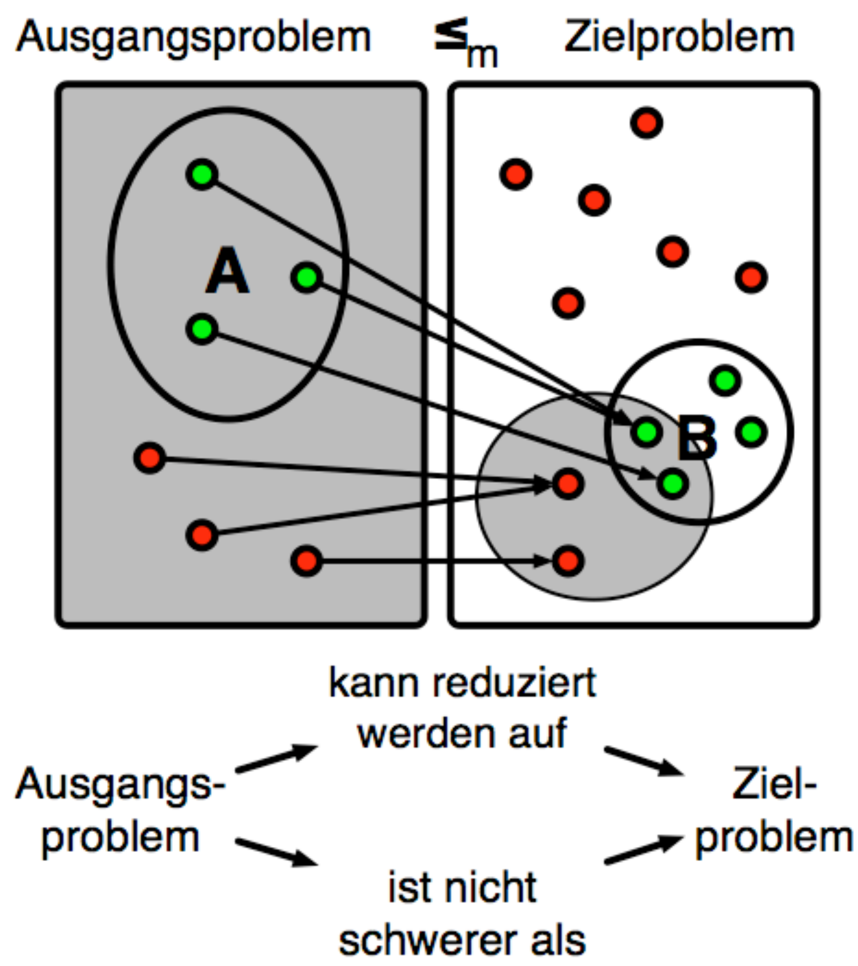
- **Unentscheidbare Probleme**
  - Das Halteproblem
  - Das Leerheitsproblem einer Turingmaschine
- **Ein einfaches nicht berechenbares Problem**
  - Das Postsche Korrespondenzproblem
- **Abbildungsreduktionen**
  - Definition
  - Anwendungen
  - Äquivalenzproblem zweier Turingmaschen
  - Der Satz von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

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

Wiederholung: Abbildungsreduktion

- **Definition**
  - Eine Funktion f: $\Sigma^* \rightarrow \Sigma^*$ 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: **A $\leq_m$ B**,
    - falls es eine 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.



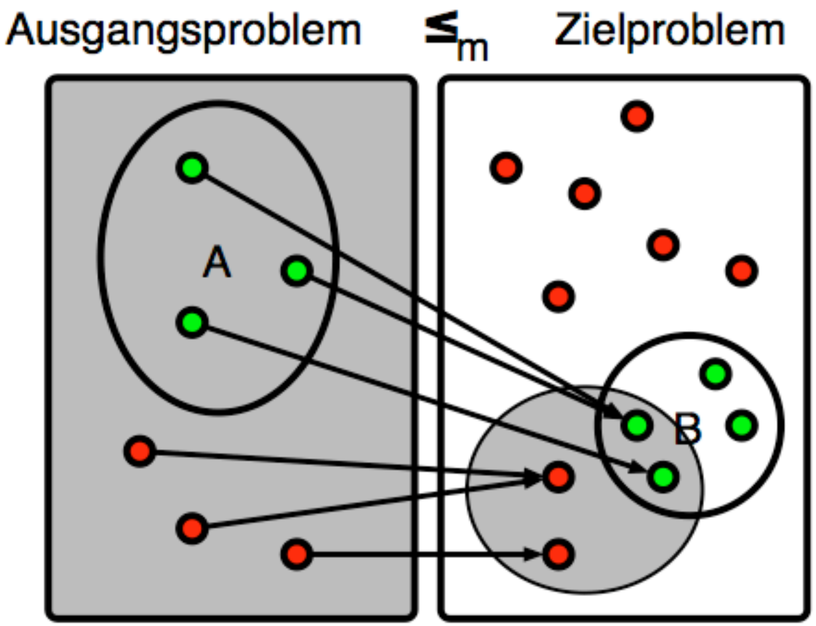
Reduktionen und Rekursive Aufzählbarkeit

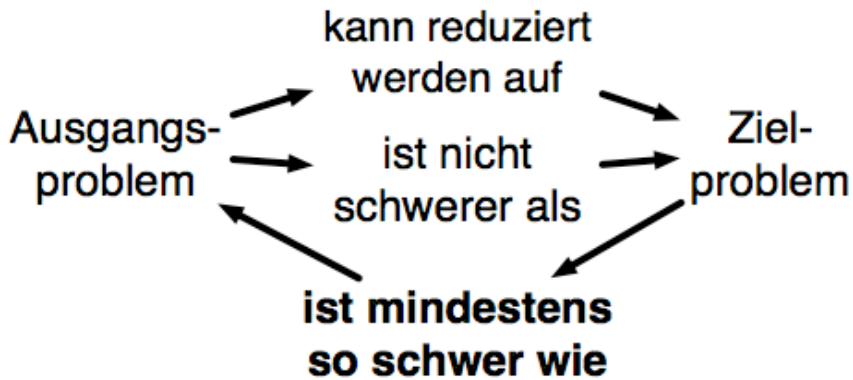
- **Theorem**
  - Falls $A \leq_m B$ und B ist rekursiv aufzählbar, dann ist A rekursiv aufzählbar.
- **Beweis**
  - Sei M, eine Turing-Maschine, die B akzeptiert.
  - Betrachte die Akzeptor-TM N:
  - N = “Auf Eingabe w:
    - Berechne f(w)
    - Führe die Berechnung von M auf Eingabe f(w) durch
   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

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

Nicht-Rekursive Aufzählbarkeit und Reduktionen

- **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.

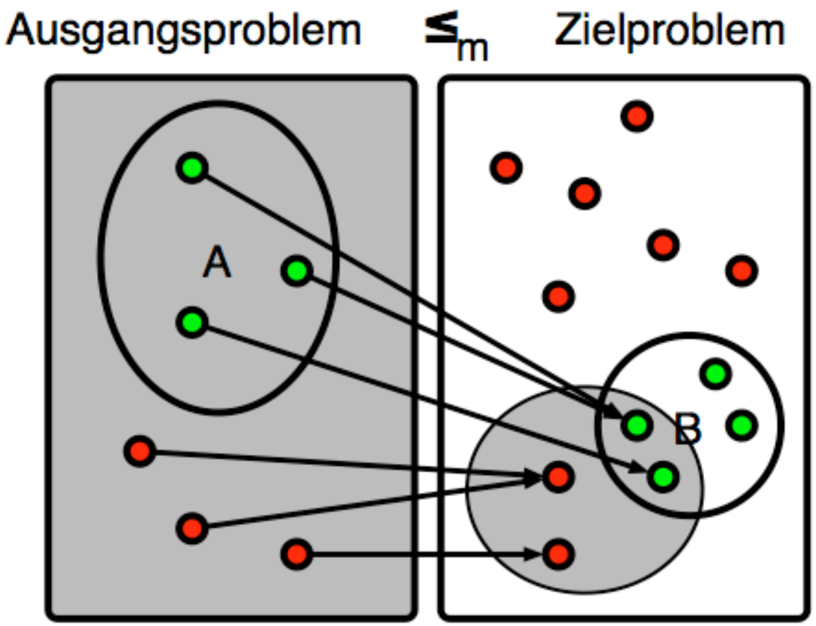


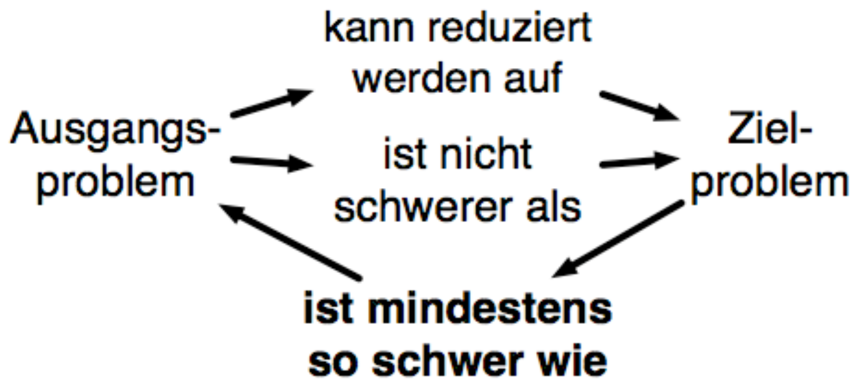


Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Zusammenfassung: Abbildungsreduktionen

- **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

- **Definition**
  - Das TM-Äquivalenzproblem
    - Gegeben: TM M_1 und TM M_2
    - Gesucht: Ist $L(M_1) = L(M_2)$?
  - Definition als Sprache:
    - $\mathrm{EQ}_{\mathrm{TM}} = \{\langle \mathsf{M}_1, \mathsf{M}_2 \rangle \mid \mathsf{M}_1, \mathsf{M}_2 \text{ sind TMs und } L(M_1) = L(M_2)\}$
- **Theorem**
  - EQ_{TM} ist weder rekursiv aufzählbar noch rekursiv ko-aufzählbar.
- **Beweisidee:**
  - Reduktion: $A_{TM} \le_m \overline{EQ_{TM}}$
    - äquivalent zu $\overline{A_{TM}} \le_m EQ_{TM}$
    - beweist, dass $\overline{EQ_{TM}}$ nicht rekursiv aufzählbar ist.
  - Reduktion: $\overline{A_{TM}} \le_m EQ_{TM}$
    - beweist, dass EQ_{TM} nicht rekursiv aufzählbar ist.

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

- **Reductionsfunktion: 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

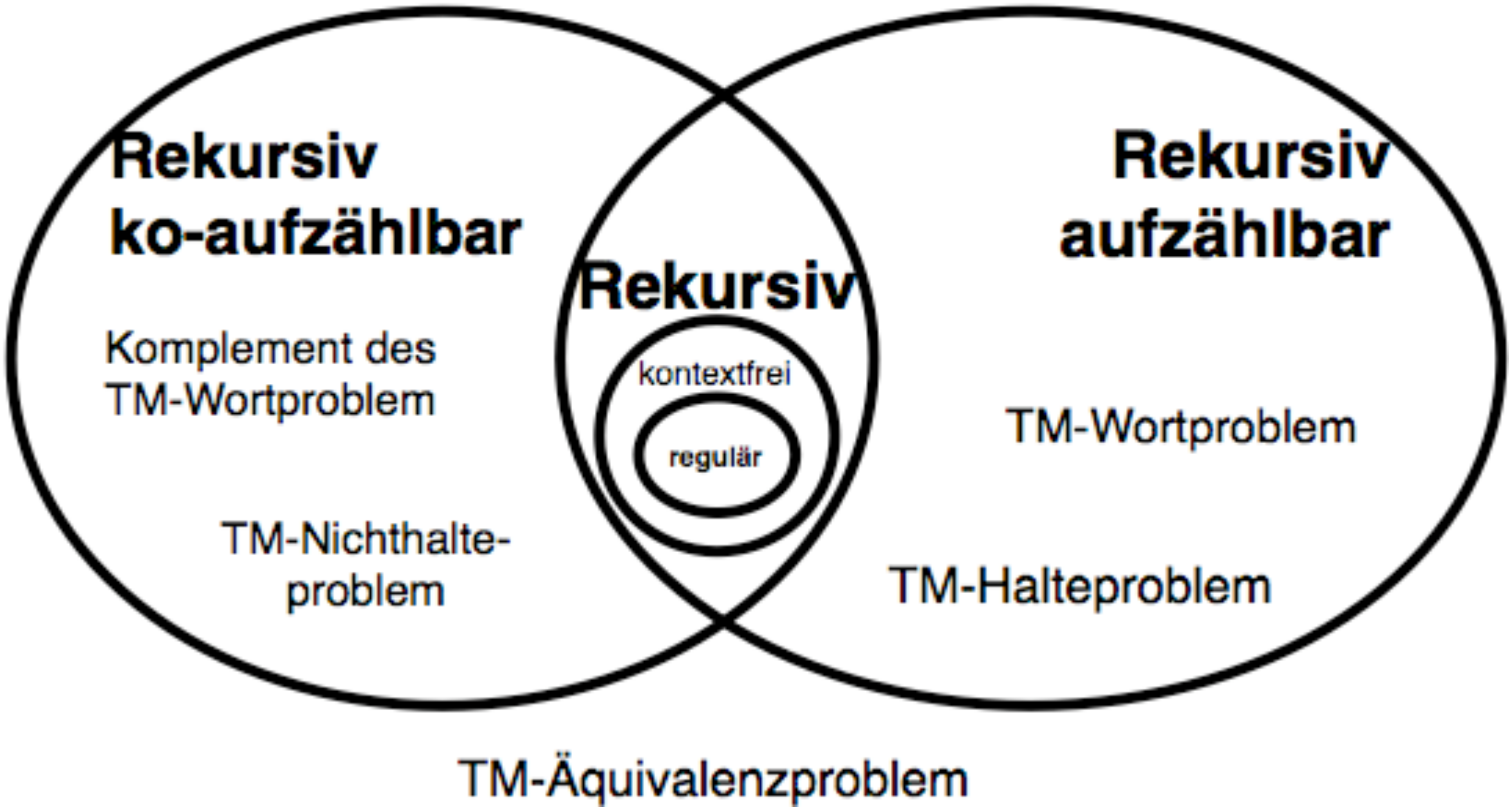
- **Daraus folgt:**
  - $<M,w> \in A_{TM} \Leftrightarrow F(<M,w>) \notin EQ_{TM}$

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

- **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 $\langle M_1, M_2 \rangle$ aus"
- **Zu beweisen:**
  - **F ist berechenbar**
    - die Kodierung der TM kann automatisch erfolgen
  - $\langle M,w\rangle \in A_{TM} \Leftrightarrow F(\langle M,w\rangle) \in EQ_{TM}$
- **M ist TM und $F(\langle M,w\rangle) = \langle M_1, M_2\rangle$**
  - wobei $L(M_1) = \Sigma^*$ und
  - $L(M_2) = \Sigma^*$, falls M(w) akzeptiert
  - $L(M_2) = \emptyset$, falls M(w) nicht akzeptiert
- **Daraus folgt:**
  - $\langle M,w\rangle \in A_{TM} \Leftrightarrow F(\langle M,w\rangle) \in EQ_{TM}$

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

Überblick



Der Satz von Rice

- **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

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

- **Beispiele**

$$\mathsf{L}_1 = \{\langle \mathsf{M} \rangle \mid M \text{ ist eine TM, die eine reguläre Sprache akzeptiert}\}$$

$$\mathsf{L}_2 = \{\langle \mathsf{M} \rangle \mid M \text{ ist eine TM, welche keine Eingabe akzeptiert}\}$$

$$\mathsf{L}_3 = \{\langle \mathsf{M} \rangle \mid M \text{ ist eine TM, so dass } M(10) = 1\}$$

Beweis des Satzes von Rice - 1. Teil

- **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

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

- **Beweis: 1. Fall: $\emptyset \notin K$**
  - **Reduktion: $A_{TM} \leq_m L_K$:**
  - Da die Klasse K nicht trivial ist,
  - existiert eine Sprache $A \in K$
  - Sei M_A eine TM mit $L(M_A) = A$

- **Betrachte Reduktionsfunktion F:**
  - F = "Auf Eingabe <M, w>:
    - Konstruiere TM M':
      M' = "*Für Eingabe x:*
      - *Führe M auf Eingabe w aus*
      - *Falls M das Wort w akzeptiert,*
        - *führe TM* ***M_A auf x*** *aus,*
        - *gib Ergebnis $M_A(x)$ aus*
      - *Ansonsten verwerfe* "
    - F gibt <M'> aus"

- **Korrektheit der Reduktion:**
  - Falls $\langle M, w\rangle \in A_{TM}$
    - F(<M, w>) = TM, die A akzeptiert
    - daraus folgt: $F(\langle M, w\rangle) \in L_K$
  - Falls $\langle M, w\rangle \notin A_{TM}$
    - F(<M, w>) = TM, die nichts akzeptiert
    - daraus folgt: $F(\langle M, w\rangle) \notin L_K$

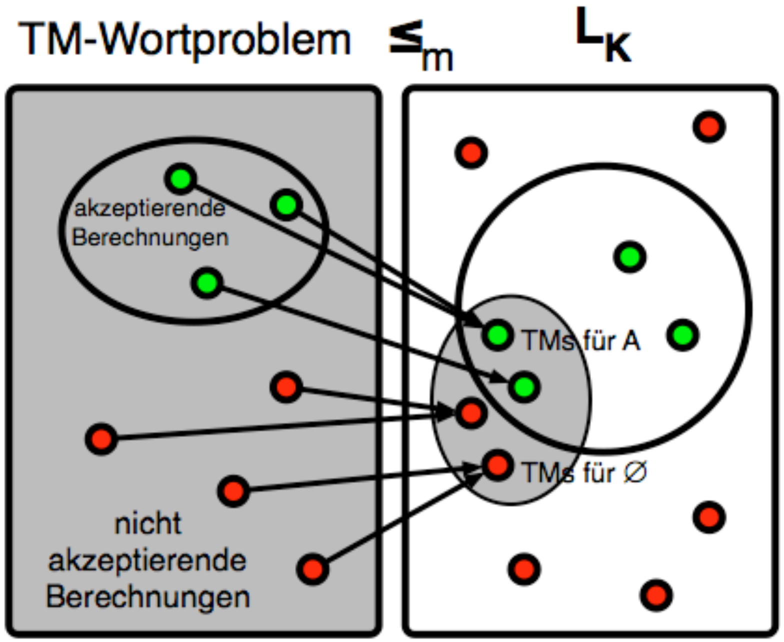
Beweis des Satzes von Rice - 1. Teil

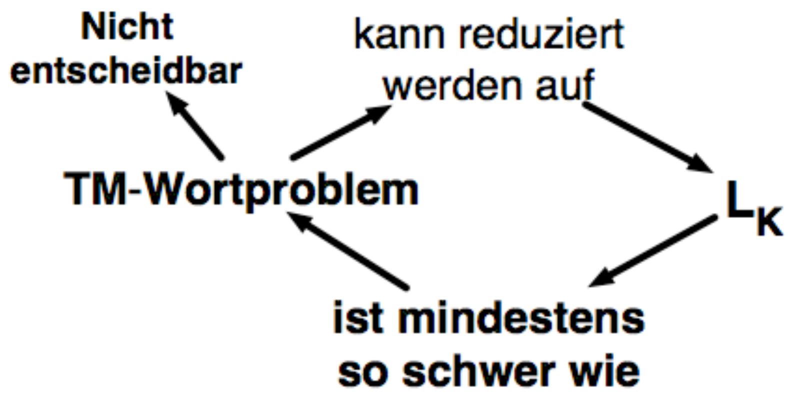
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **1. Fall: $\emptyset \notin K$**
- **Reduktion: $A_{TM} \leq_m L_K$:**

- **Betrachte Reduktionsfunktion F:**
  - F = "Auf Eingabe <M, w>:
    - Konstruiere TM M':
      M' = "*Für Eingabe x:*
      - *Führe M auf Eingabe w aus*
      - *Falls M das Wort w akzeptiert,*
        - *führe TM **M_A auf x** aus,*
        - *gib Ergebnis $M_A(x)$ aus*
      - *Ansonsten verwerfe* "
    - F gibt <M'> aus"

- **Korrektheit der Reduktion:**
  - Falls <M, w> $\in A_{TM}$
    - F(<M, w>) = TM, die A akzeptiert
    - daraus folgt: F(<M, w>) $\in L_K$
  - Falls <M, w> $\notin A_{TM}$
    - F(<M, w>) = TM, die nichts akzeptiert
    - daraus folgt: F(<M, w>) $\notin L_K$





Beweis des Satzes von Rice - 2. Teil

- **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

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

- **Beweis: 2. Fall: $\varnothing \in K$**
  - **Reduktion: $\overline{A_{TM}} \leq_m L_K$:**
  - Da die Klasse K nicht trivial ist,
  - existiert eine Sprache $B \notin K$
  - Sei M_B die TM mit $L(M_B) = B$

- **Betrachte Reduktionsfunktion F:**
  - F = “Auf Eingabe <M, w>:
    - Konstruiere TM M’:
      M’ = “Für jede Eingabe x:
      - Führe M auf Eingabe w aus
      - Falls M das Wort w akzeptiert,
        - führe TM **M_B auf x** aus,
        - gib Ergebnis $M_B(x)$ aus
      - Ansonsten verwerfe ”
    - F gibt <M’> aus”

- **Korrektheit der Reduktion:**
  - Falls $\langle M, w\rangle \in A_{TM}$
    - F(<M, w>) = TM, die B akzeptiert
    - daraus folgt: $F(\langle M, w\rangle) \notin L_K$
  - Falls $\langle M, w\rangle \notin A_{TM}$
    - F(<M, w>) = TM, die nichts akzeptiert
    - daraus folgt: $F(\langle M, w\rangle) \in L_K$

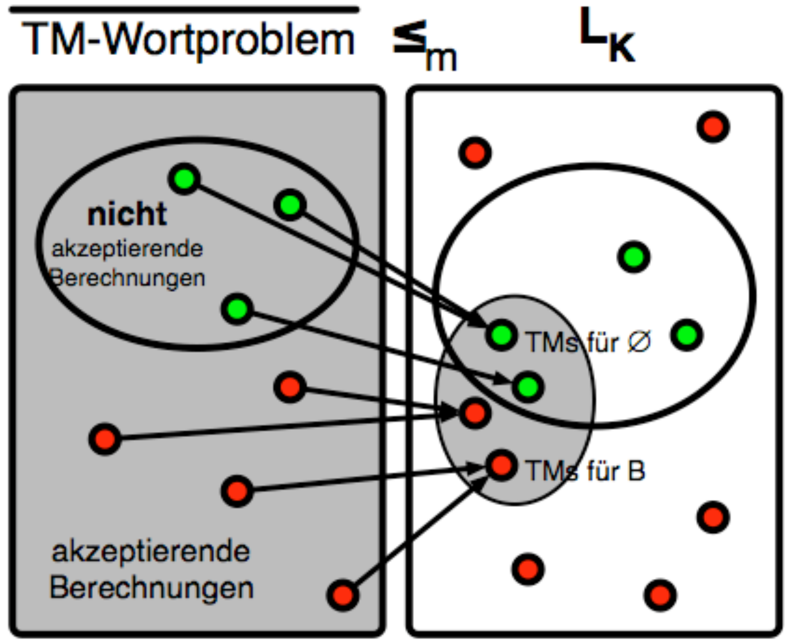
Beweis des Satzes von Rice - 2. Teil

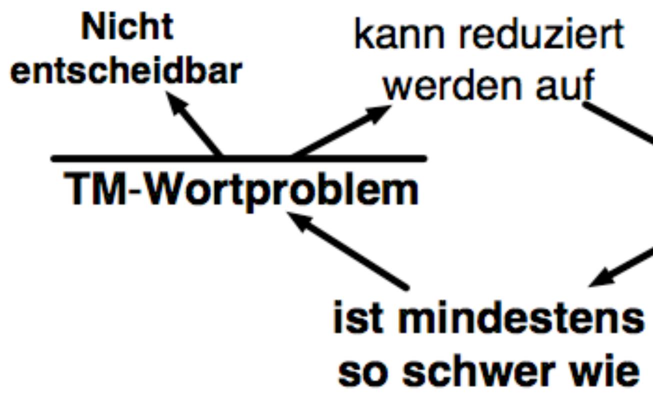
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Beweis: 2. Fall: $\emptyset \in K$**
- **Reduktion: $\overline{A_{TM}} \leq_m L_K$:**

- **Betrachte Reduktionsfunktion F:**
  - F = “Auf Eingabe <M, w>:
    - Konstruiere TM M’:
      M’ = “Für Eingabe x:
      - Führe M auf Eingabe w aus
      - Falls M das Wort w akzeptiert,
        - führe TM **M_A auf x** aus,
        - gib Ergebnis $M_A(x)$ aus
      - Ansonsten verwerfe ”
    - F gibt <M’> aus”

- **Korrektheit der Reduktion:**
  - Falls <M, w> $\in A_{TM}$
    - F(<M, w>) = TM, die A akzeptiert
    - daraus folgt: F(<M, w>) $\in L_K$
  - Falls <M, w> $\notin A_{TM}$
    - F(<M, w>) = TM, die nichts akzeptiert
    - daraus folgt: F(<M, w>) $\notin L_K$





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

Turing-Reduktionen: Vorüberlegung

- **Die Abbildungsreduktion ist ein “schwacher” Reduktionsbegriff**

- **Gedankenexperiment:**
  - Sei B eine entscheidbare Sprache
  - Dann gibt es ein haltendes Programm M, das B entscheidet.
  - Wir benutzen M nun als Unterprogramm
  - Wenn ein Programm M’ jetzt ein anderes Problem A entscheiden will,
    - kann es M beliebig häufig als Unterprogramm verwenden
      - weil M immer hält
  - Wenn das Programm M’ auf jeder Ausgabe von M immer hält und eine Entscheidung trifft,
    - dann löst es A mit Hilfe von B.
  - A ist entscheidbar,
    - weil man M und M’ zu einem Programm verschmelzen kann.

- **Diese Art Reduktion ist nicht durch den Begriff der Abbildungsreduktion abgedeckt.**

Orakel-Turing-Maschinen

- **Definition**
  - Ein **Orakel** für eine Sprache B
    - ist eine externe Einheit, welche für ein gegebenes Wort w entscheidet, ob w ein Element von B ist.
  - Eine **Orakel-Turing-Maschine (OTM)**, ist eine modifizierte Turing-Maschine,
    - welche beliebig häufig ein Orakel befragen kann.
      - Hierzu schreibt die **OTM** die Anfrage auf ein separates Orakelband
      - und findet nach dem Schreiben des Endsymbols
      - sofort die Antwort auf dem selben Band.
- **Beobachtung**:
  - Orakel müssen nicht notwendigerweise berechenbar sein.
  - Z.B. mit dem Halteproblem als Orakel lässt sich das TM-Wortproblem lösen:
    1. Frage Halteproblem-Orakel, ob gegebene TM M auf gegebener Eingabe w hält
    2. Falls nein, gib nein aus.
    3. Falls ja, führe M auf Eingabe w aus
    4. Gib Ergebnis von M(w) aus

Ende der 14. Vorlesung

CoNe Freiburg

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

Christian Schindelhauer
Wintersemester 2006/07
14. Vorlesung
08.12.2006