Informatik III

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

Arne Vater
Wintersemester 2006/07
10. Vorlesung
24.11.2006

Turing-maschinen

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

Turingmaschinen

- **Eine (*deterministische 1-Band*) *Turingmaschine* (DTM) 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 und es gilt:**
  - $F \subseteq Q$, $\Sigma \subseteq \Gamma$, $q_0 \in Q$
  - $\_ \in \Gamma \cap \Sigma$ ist das *Blank*symbol.

- **Q ist die *Zustandsmenge***
- **Σ ist das *Eingabealphabet***
- **Γ das *Bandalphabet*.**

- **Zustände**
  - $q_0 \in Q$ ist der *Startzustand.*
  - $q_{accept} \in Q$ ist der akzeptierende Endzustand
  - $q_{reject} \in Q$ ist der ablehnende Endzustand

- **$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ ist die (partielle) *Übergangsfunktion***
  - ist nicht definiert für $q \in \{q_{accept}, q_{reject}\} \subseteq \Gamma$ definiert

Arbeitsweise einer Turingmaschine

- **Initial:**
  - Eingabe steht links auf dem Band
  - Der Rest des Bands ist leer
  - Kopf befindet sich ganz links

- **Berechnungen finden entsprechend der Übergangsfunktion statt**

- **Wenn der Kopf sich am linken Ende befindet und nach links bewegen soll, bleibt er an seiner Position**

- **Wenn q_{accept} oder q_{reject} erreicht wird, ist die Bearbeitung beendet**

Konfiguration

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

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

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

Aufeinanderfolgende Konfigurationen

- **Gegeben: Konfigurationen C_1, C_2**

- **Wir sagen:**
  - **Konfiguration C_1 führt zu C_2**, falls die TM 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 $\delta(q_i,b) = (q_j,c,L)$ und
  - **u a q_i b v** führt zu **u a c q_j v**,
    - falls $\delta(q_i,b) = (q_j,c,R)$

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

Konfigurationen

- **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

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

Akzeptanz von Turingmaschinen

- **Eine Turingmaschine M akzeptiert eine Eingabe w, falls es eine Folge von Konfigurationen C_1, C_2, …, C_k gibt, so dass**
  - C_1 ist die Startkonfiguration von M bei Eingabe w
  - C_i führt zu C_{i+1}
  - C_k ist eine akzeptierende Konfiguration

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

- **Eine Turingmaschine *entscheidet* eine Sprache, wenn jede Eingabe in einer haltende Konfiguration C_k resultiert**

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

Rekursive und rekursiv aufzählbare Sprachen

- **Eine Sprache L heißt *rekursiv aufzählbar*, falls es eine Turingmaschine M gibt, die L akzeptiert**

- **Eine Sprache L heißt *rekursiv* oder *entscheidbar*, falls es eine Turingmaschine M gibt, die L entscheidet**

Mehrband Turingmaschinen

- **Eine *Mehrband* oder *k-Band Turingmaschine* (k-Band DTM) hat *k* Bänder mit je einem Kopf.**

- **Die Übergangsfunktion ist dann von der Form**
  – $\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k$

- **Die Arbeitsweise ist analog zu 1-Band-DTMs definiert.**
  – Zu Beginn steht die Eingabe auf Band 1,
  – sonst stehen überall Blanks.

Äquivalenz von 1-Band und Mehrband Turingmaschinen

- **Satz:**
  - Zu jeder Mehrband Turingmaschine gibt es eine äquivalente 1-Band Turingmaschine

- **Korollar:**
  - Eine Sprache L ist genau dann **rekursiv aufzählbar**, wenn es eine **Mehrband-Turingmaschine** gibt, die L akzeptiert

Warum rekursiv aufzählbar rekursiv aufzählbar heißt

- **Definition**
  - Eine aufzählende Turing-Maschine ist eine Turingmaschine, die mit einem zusätzlichen speziellen Ausgabe-Band ausgestattet ist.
    - Die Turing-Maschine muss nicht unbedingt halten.
    - Auf dem Ausgabeband kann die Turingmaschine nur nach rechts gehen.
    - Wörter sind durch das Sondersymbol "_" von einander getrennt und können damit weder gelöscht noch überschrieben werden.
    - Die Vereinigung aller jemals erzeugten Wörter, beschreibt die Sprache der aufzählenden Turing-Maschine.
- **Theorem**
  - Eine Sprache ist rekursiv aufzählbar genau dann wenn eine aufzählende Turing-Maschine sie beschreibt.

Beweis des Theorems

- **Theorem**
  - Eine Sprache ist rekursiv aufzählbar genau dann wenn eine aufzählende Turing-Maschine sie beschreibt.
- **Beweis (⇐)**
  - Sei U eine Aufzähler-TM für die Sprache A
  - Wir konstruieren eine Akzeptor-TM K wie folgt
  - K = "Auf Eingabe w:
    1. Simuliere U.
    2. Jedes Mal, wenn U eine Ausgabe macht, vergleiche sie mit w
    3. Falls w erscheint, akzeptiere"
- **Beweis (⇒)**
  - Sei K eine Akzeptor-TM
  - Sei $s_1, s_2, ..$ eine einfach erzeugbare Folge aller Zeichenketten
    - z.B. längenlexikographisch: ε,0,1,00,01,10,11,000,001,...
  - Wir konstruieren eine Aufzähler-TM U wie folgt:
  - U = "
    1. Für i =1,2, ..
    2. Für jedes w aus $\{s_1, s_2, .., s_i\}$
    3. Falls K auf Eingabe w in i Schritten hält und akzeptiert, gib w aus."
  - Falls K eine Eingabe w in endlicher Zeit akzeptiert, wird sie ausgegeben
    - sogar beliebig häufig
  - Andere Ausgaben werden von U nicht erzeugt.

Rekursiv aufzählbar beinhaltet rekursiv entscheidbar

- **Korollar**
  - Jede rekursiv entscheidbare Menge ist rekursiv aufzählbar
- **Beweis:**
  - Betrachte die DTM, die eine rekursiv entscheidbare Menge M entscheidet
  - Diese DTM ist bereits ein Maschine, die M akzeptiert
    - Da die aufzählbare Mengen über die Akzeptor-TM definiert ist

Der Maschinenpark der Turingmaschinen

- **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

Church-Turing-These

Was heißt entscheidbar?

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

- **Genau dann wenn eine**
  - 1-Band-TM
  - 2-Band-TM
  - 3-Band-TM
  - ..
  - k-Band-TM
  - NTM
- **für jedes Wort einer Sprache in endlicher Zeit ausgibt, ob es in L ist oder nicht, dann ist die Sprache entscheidbar.**
- **Gilt auch für ein Programm in**
  - Java
  - C
  - C++
  - Pascal
  - Fortran

Hilberts 10. Problem

- **Rede auf dem internationalen Mathematiker-Kongreß, Paris 1900**
  - mit 23 (seiner Zeit aktuellen Problemen)
- **das 10. Problem**
  - Eine **Diophantische** Gleichung mit irgend welchen Unbekannten und mit ganzen rationalen Zahlencoefficienten sei vorgelegt: *man soll ein Verfahren angeben, nach welchem sich mittelst einer endlichen Anzahl von Operationen entscheiden löst, ob die Gleichung in ganzen rationalen Zahlen lösbar ist.*
- ***Frage beantwortet durch***
  - *Yuri Matiyasevich 1970*
  - *nach Vorarbeiten von Martin Davis und Julia Robinson*

Diophantisches Polynom

- **Beispielprobleme**
  - Gibt es x,y aus den ganzen Zahlen **Z**, so dass
    - x + y - 3 = 0 gilt?
  - Gibt es x,y,z aus **Z** so dass gilt:

$$6x^3yz^2 + 3y^2 - x^3 - 10 = 0$$

- **Lösung des 10. Hilbertschen Problem:**
  - Geht nicht!
  - Soll heißen: es gibt kein algorithmisches Verfahren, dass dieses Problem lösen kann.
- **Liegt das aber vielleicht an unserem eingeschränkten Begriff der Verfahren?**
  - Gibt es mächtigere Programmiersprachen als die der Turing-Maschine, Java, C++, ...
- **Antwort:**
  - ?

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

Die Church-Turing-These

- **Nachdem man eine Reihe von verschiedenen diskreten Berechnungsmodellen als gleichwertig bewiesen hat, hat die Fachwelt die folgende These als allgemeingültig angesehen:**
  - Lambda-Kalkül von Alonzo Church (1936)
  - Turing-Maschine von Alan Turing (1936)

- **Church-Turing-These**
  - Der intuitive Begriff eines Algorithmus wird vollständig beschrieben durch die Algorithmen, welche Turing-Maschinen beschreiben können.

- **Tatsächlich sind alle Maschinen, die bisher von Menschenhand gebaut wurden, durch eine Turing-Maschine beschreibbar.**
- **Hoffnung für Gegenbeispiele:**
  - Analog-Computer
  - Quanten-Computer
  - Computer, die man in schwarze Löcher wirft.

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

Hilberts 10. Problem ist nicht entscheidbar, aber rekursiv aufzählbar

- **Theorem**
  - Die Menge der diophantischen Gleichungen mit ganzzahligen Lösungen ist rekursiv aufzählbar.
- **Beweis: folgt gleich**

- **Theorem**
  - Hilberts 10. Problem ist nicht rekursiv entscheidbar.
- **Beweis**
  - sprengt den Rahmen dieser Vorlesung

Hilberts 10. Problem ist rekursiv aufzählbar

- **Theorem**
  - Die Menge der diophantische Gleichungen mit ganzzahligen Lösungen ist rekursiv aufzählbar.
- **Beweis**
  - Wir konstruieren eine Akzeptor-TM M
  - Gegeben eine diophantische Gleichung $G(x_1,x_2,..,x_m)$ mit den Variablen $x_1,x_2,..,x_m$
  - M = “Für b = 1,2,3,...

    Für alle $x_1,x_2,..,x_m \in \{-b,-b+1,...,-1,0,1,2,..., b\}$

    Falls $G(x_1,x_2,..,x_m) = 0$ akzeptiere”
  - Beweis der Korrektheit:
    - Falls für $y_1,y_2,..,y_m$: $G(y_1,y_2,..,y_m) = 0$
      - Dann wird für $b= \max\{| y_1 |, | y_2 |,.., | y_m |\}$ die Kombination $y_1,y_2,..,y_m$ in $x_1,x_2,..,x_m$ eingesetzt
    - Falls für alle $x_1,x_2,..,x_m$: $G(x_1,x_2,..,x_m) \neq 0$,
      - akzeptiert M niemals.

Entscheidbare Sprachen

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

- **Reguläre Sprachen**
  - Wortproblem
  - Leerheitsproblem
  - Äquivalenzproblem
- **Kontextfreie Sprachen**
  - Wortproblem
  - Leerheitsproblem
  - Äquivalenzproblem
- **Das Halteproblem**
  - Diagonalisierung
  - Das Halteproblem ist nicht entscheidbar
  - Eine nicht rekursiv aufzählbare Sprache

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

Entscheidbare reguläre Sprachprobleme

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

Entscheidbare Reguläre Sprachprobleme

- **Das Wortproblem regulärer Sprachen**
  - Gegeben:
    - Ein DFA B
    - ein Wort w
  - Gesucht:
    - Akzeptiert B das Wort w?
- **Beschrieben durch die Sprache:**

$$\mathsf{A}_{\mathrm{DFA}} = \{\langle \mathsf{B}, \mathsf{w}\rangle \mid B \text{ ist ein DFA der eine Eingabe } x \text{ akzeptiert}\}$$

  - DFA ist geeignet kodiert
  - Klammern stehen für eine geeignete Tupelkodierung
- **Das Problem einer Sprache L wird beschrieben durch die charakteristische Funktion der Sprache L**

$$\chi_{\mathsf{L}}(\mathsf{w}) = \begin{cases} 1\,, & w \in L \\ 0\,, & w \notin L \end{cases}$$

Das Wortproblem regulärer Sprachen

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

- **Theorem**
  - Das Wortproblem der regulären Sprachen ist entscheidbar.

$$\mathrm{A}_{\mathrm{DFA}} = \{\langle \mathrm{B}, \mathrm{w}\rangle \mid B \text{ ist ein DFA der eine Eingabe } x \text{ akzeptiert}\}$$

- **Beweis**
  - Konstruiere eine Turing-Masche M, die folgendes berechnet:
  - M = “Für Eingabe <B,w> mit DFA B und Wort w:
    1. Simuliere B auf Eingabe w
    2. Falls die Simulation akzeptiert, dann akzeptiere
       Sonst verwerfe”
  - Implementationsdetails:
    - Übergangsfunktion ist geeignet kodiert auf einem Band
    - Zustand ist auf einem separaten Band
    - Eingabe auf einem dritten Band
    - Suche nächsten Übergang in der Kodierung des DFAs

Das Leerheitsproblem regulärer Sprachen

- **Das Leerheitsproblem regulärer Sprachen:**
  - Gegeben:
    - Ein DFA A
  - Entscheide:
    - Ist L(A) = ∅
- **Die zugehörige Sprache wird beschreiben durch:**

$$E_{\mathrm{DFA}} = \{ \langle \mathrm{A} \rangle \mid A \text{ ist ein DFA und } L(A) = \emptyset \}$$

- **Theorem**
  - Das Leerheitsproblem regulärer Sprachen ist entscheidbar

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

Das Leerheitsproblem regulärer Sprachen

- **Theorem**
  - Das Leerheitsproblem regulärer Sprachen ist entscheidbar.

$$\mathrm{E}_{\mathrm{DFA}} = \{\langle \mathrm{A} \rangle \mid A \text{ ist ein DFA und } L(A) = \emptyset \}$$

- **Beweis**
  - Ein DFA akzeptiert mindestens ein Wort, wenn der DFA mindestens einen vom Startzustand erreichbaren Zustand besitzt
  - T = "Auf Eingabe A, wobei A ein DFA ist
    1. Markiere Startzustand A
    2. Wiederhole
    3. Markiere jeden Folgezustand eines markierten Zustands
    4. bis kein neuer Folgezustand in A markiert wurde
    5. Falls kein akzeptierender Zustand markiert wurde: Akzeptiere sonst: verwerfe"

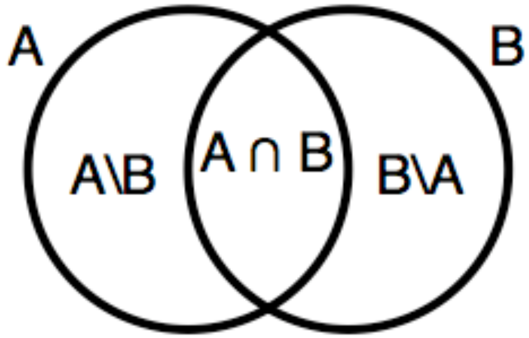
Das Äquivalenzproblem regulärer Sprachen

- **Das Äquivalenzproblem regulärer Sprachen:**
  - Gegeben:
    - Zwei DFAs A, B
  - Entscheide:
    - Ist L(A) = L(B)
- **Die zugehörige Sprache wird beschrieben durch:**

$$\mathrm{EQ}_{\mathrm{DFA}} = \{\langle \mathrm{A}, \mathrm{B} \rangle \mid A, B \text{ sind DFAs und } L(A) = L(B) \ \}$$

- **Theorem**
  - Das Äquivalenzproblem regulärer Sprachen ist entscheidbar
- **Beobachtung für zwei Mengen A,B:**

$A = B \Leftrightarrow (A \setminus B) \cup (B \setminus A) = \varnothing$



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

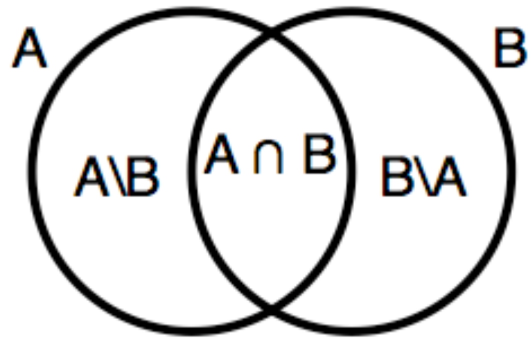
Das Äquivalenzproblem regulärer Sprachen

- **Theorem**
  - Das Äquivalenzproblem regulärer Sprachen ist entscheidbar

$$\mathrm{EQ}_{\mathrm{DFA}} = \{\langle \mathsf{A}, \mathsf{B} \rangle \mid A, B \text{ sind DFAs und } L(A) = L(B)\ \}$$

- **Beobachtung für zwei Mengen A,B:**

A = B ⇔ (A \ B) ∪ (B \ A) = ∅



- **Beweis:**
  - Konstruiere DFA X für die Sprache (A \ B) ∪ (B \ A)
  - Durch Mehrfachanwendung des kartesischen Produkts
  - Teste ob L(X) = ∅
    - mit Hilfe der Turing-Maschine für das Leerheitsproblem

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

(Un-) Entscheidbare kontextfreie Sprachprobleme

Das Wortproblem der kontextfreien Sprachen

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

- **Wortproblem der kontextfreien Sprachen:**
  - Gegeben:
    - Kontextfreie Grammatik G in geeigneter Kodierung
    - Wort w
  - Gesucht:
    - Ist $w \in L(G)$?
- **Theorem**
  - Das Wortproblem der kontextfreien Sprachen ist entscheidbar.
- **Beweis**
  - Wandle Grammatik in Chomsky-Normalform um
  - Führe Cocke-Younger-Kasami-Algorithmus für eine gegebene kontextfreie Grammatik und ein gegebenes Wort durch
  - Akzeptiere falls das Wort vom Startsymbol abgeleitet werden kann, ansonsten verwerfe.

Beispiel

- **Ist „baaba“ in L(G)?**

$G = (V, \Sigma, S, P)$, wobei $V = \{S, A, B, C\}$ und $\Sigma = \{a, b\}$.

$P = \{$ $S \rightarrow AB$, $S \rightarrow BC$,
$A \rightarrow BA$
$B \rightarrow CC$,
$C \rightarrow AB$,
$A \rightarrow a$, $B \rightarrow b$, $C \rightarrow a$ $\}$

- Dann akzeptiere
- Sonst lehne ab

l = 5	{S,A,C}				
l = 4	{}	{S,A,C}			
l = 3	{}	{B}	{B}		
l = 2	{S,A}	{B}	{S,C}	{S,A}	
l = 1	{B}	{A,C}	{A,C}	{B}	{A,C}
	b	a	a	b	a

Das Leerheitsproblem der kontextfreien Sprachen

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

- **Das Leerheitsproblem kontextfreier Sprachen:**
  - Gegeben:
    - Eine kontextfreie Grammatik G
  - Entscheide:
    - Ist L(G) = ∅
- **Theorem**
  - Das Leerheitsproblem kontextfreier Sprachen ist entscheidbar.
- **Beweis**
  - Wandle G in Chomsky-Normalform um
  - Markiere alle Terminalsymbole
  - Solange neue Markierungen erscheinen
    - Markiere alle Nichtterminale, die Regeln besitzen deren rechte Seite vollständig markiert ist
  - Falls das Startsymbol markiert ist, verwerfe
  - Ansonsten akzeptiere

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

Beispiel

$G = (V, \Sigma, S, P)$, wobei $V = \{S, A, B, C\}$ und $\Sigma = \{a, b\}$.

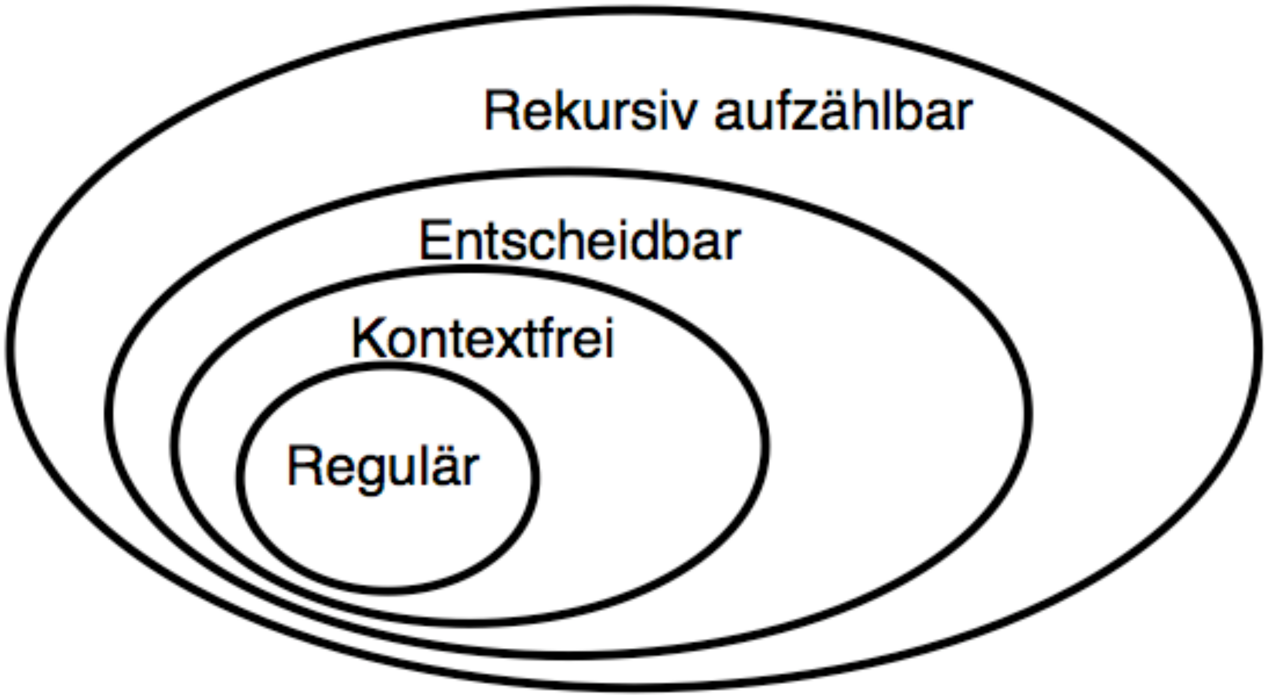
$P = \{$ $S \to AB$, $S \to BC$,
$A \to BA$
$B \to CC$,
$C \to AB$,
$A \to a$, $B \to b$, $C \to a$ $\}$

Das Äquivalenzproblem der kontextfreien Sprachen

- **Das Äquivalenzproblem kontextfreier Sprachen:**
  - Gegeben:
    - Kontextfreie Grammatiken G,G'
  - Entscheide:
    - Ist L(G) = L(G')
- **Problem:**
  - Kontextfreie Sprachen sind **nicht** abgeschlossen
    - unter Komplement
    - unter Schnittoperationen
  - Beweis des Äquivalenzproblem der kontextfreien Sprachen ist nicht übertragbar
- **Theorem**
  - Das Äquivalenzproblem kontextfreier Sprachen ist **nicht** entscheidbar.
  - (ohne Beweis)

Beziehungen zwischen den Sprachen

- **Jede reguläre Sprache ist eine kontextfreie Sprache.**
- **Jede kontextfreie Sprache ist eine entscheidbare Sprache.**
  - folgt aus der Entscheidbarkeit des Wortproblems der kontextfreien Sprachen.
- **Jede entscheidbare Sprache ist eine rekursiv aufzählbare Sprache.**



Ende der 10. Vorlesung

CoNe Freiburg

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

Informatik III
Arne Vater
24.11.2006