CoNe Freiburg

Informatik III

V Chomsky-Sprachklassen

Christian Schindelhauer

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik

Wintersemester 2007/08

Zusammenfassung

Chomsky-Sprachklassen

2

Die Chomsky-Klassifizierung

- **Chomsky-Hierachien**
  - 3: Reguläre Grammatiken
  - 2: Kontextfreie Grammatiken
  - 1: Wachsende kontextsensitive Grammatiken
  - 0: Allgemeine Grammatiken
- **Alternative Beschreibung**
  - Reguläre Sprachen und konstanter Platz
  - Kontextfreie Sprachen und Kellerautomaten
  - Wachsende kontextsensitive Sprachen und linearer Platz
  - Chomsky-0-Sprachen und Rekursiv aufzählbare

Übersicht Chomsky-Charakterisierung

Stufe	Sprache	Regeln	Maschinenmodell
Typ-0	**Rekursiv Aufzählbar**	**keine Einschränkung**	**Turing-Maschine**
Typ-1	**(Wachsend) Kontextsensitive Sprachen**	$\alpha A \beta \rightarrow \alpha\gamma\beta$ $A \in V$, $\lvert\gamma\rvert \geq 1$ $\alpha,\beta,\gamma \in (\Sigma \cup V)^*$	**Linear-Platz-NTM**
Typ-2	**Kontextfreie Sprachen**	$A \rightarrow \gamma$ $A \in V$ $\gamma \in (\Sigma \cup V)^*$	**Nichtdet. Kellerautomat (PDA)**
Typ-3	**Reguläre Sprachen**	$A \rightarrow aB$ $A \rightarrow a$ $A,B \in V$ $a \in \Sigma$	**Endlicher Automat (NFA/DFA)**

Zusammenfassung

Chomsky-3

5

Chomsky-3: Reguläre Sprachen und endliche Automaten

- **Definition**
  - Eine (rechts)-reguläre Grammatik ist ein Vierer-Tupel $G = (V,\Sigma,R,S)$, wobei
    - V ist die endliche Menge der Variablen (Nichtterminale)
    - Σ ist die endliche Menge der Terminale (Terminalsymbole)
    - R ist eine endliche Menge an Ersetzungsregeln (Regeln/ Produktionen)
      - Jede Regel bildet eine Variable auf ein Terminal
        - $A \to a$, mit $A \in V$ und $a \in \Sigma$
      - oder auf ein Wort aus einem Terminal und einer Variable ab
        - $A \to aB$ mit $A,B \in V$ und $a \in \Sigma$
      - oder die Startvariable wird auf das leere Wort abgebildet
        - $S \to \varepsilon$
    - $S \in V$ ist die Startvariable
- **Beispiel**
  - $(\{S,T\},\{0,1\}, \{S \to 0S, S \to 0, S \to 1T, T \to 1S, , T \to 0T\}, S)$
- **Theorem**
  - Die Menge der regulären Sprachen werden durch die rechts-regulären Grammatiken beschrieben

Komplexitätstheorie

Chomsky-2

7

Chomsky-2: Kontextfreie Sprachen und PDA

- **Definition**
  - Eine kontextfreie Grammatik ist ein Vierer-Tupel $G = (V,\Sigma,R,S)$, wobei
    - V ist die endliche Menge der Variablen (Nichtterminale)
    - Σ ist die endliche Menge der Terminale (Terminalsymbole)
      - V und Σ sind disjunkte Mengen
    - R ist eine endliche Menge an Ersetzungsregeln (Regeln/ Produktionen)
      - Jede Regel besteht aus einer Variable und einer Zeichenkette aus Variablen und Terminalen,
        - $A \rightarrow w$, mit $A \in V$, $w \in (V \cup \Sigma)^*$
    - $S \in V$ ist die Startvariable
- **Ableitung**
  - Falls die Regel $A \rightarrow w$ in R ist, dann ist $uAv \Rightarrow uwv$, d.h.
    - uAv kann zu uwv in einem Schritt abgeleitet werden
  - Wir sagen das u zu v abgeleitet werden kann oder
    $u \Rightarrow^* v$, wenn es ein $k \geq 0$ gibt mit
      - $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow ... \Rightarrow u_k \Rightarrow v$
- **Sprache der Grammatik G**
  - $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* v\}$
- **Die kontextfreien Sprachen werden durch kontextfreie Grammatiken erzeugt.**

Keller-Automaten und kontextfreien Sprachen

- **Theorem**
  - Eine Sprache ist genau dann kontextfrei, wenn ein Kellerautomat sie erkennt

Komplexitätstheorie

Chomsky-1

10

Chomsky-1: (Wachsend) Kontext-sensitive Sprachen und NSPACE(n)

- **Definition (Wachsend) Kontextsensitive Grammatik**
  - Eine wachsend kontext-sensitive Grammatik ist ein Vierer-Tupel $G = (V,\Sigma,R,S)$, mit
    - Variablen V (Nichtterminale)
    - Terminalen Σ und
    - einer Regelmenge R
      - Jede Regel besteht aus einer Variable und einem Kontext α, β, wobei die Variable auf ein Wort mindestens der Länge 1 abgebildet wird:
      - $\alpha A \beta \rightarrow \alpha \gamma \beta$ mit $A \in V$, $|\gamma| \geq 1$, $\alpha,\beta,\gamma \in (\Sigma \cup V)^*$
      - oder $S \rightarrow \varepsilon$
    - $S \in V$ ist die Startvariable
  - Die kontextsensitive Grammatiken produzieren die kontextsensitiven Sprachen CSL
- **Theorem**
  - CSL = NSPACE(n)

Beispiel einer kontext-sensitiven Grammatik

- **G=(V,Σ,R,S) mit**
  - Variablen V = {S,B,C,D}
  - Terminalen Σ = {a,b,c}
  - Regeln R:
    - S → aSBC
    - S → aBC
    - CB → DB
    - DB → DC
    - DC →BC
    - aB → ab
    - bB → bb
    - bC → bc
    - cC → cc
  - Startsymbol: S

- **Beispiel:**
- **S** ⇒ **aSBC**
  - ⇒ a**aSBC**BC
  - ⇒ aa**aBC**BCBC
  - ⇒ aaaB**D**BCBC
  - ⇒ aaaBD**C**CBC
  - ⇒ aaaB**B**CCBC
  - ...
  - ⇒ aaaBBC**BC**C
  - ...
  - ⇒ aaaBB**BC**CC
  - ⇒ aaa**b**BBCCC
  - ⇒ aaab**b**BCCC
  - ⇒ aaabb**b**CCC
  - ⇒ aaabbb**c**CC
  - ⇒ aaabbbc**c**C
  - ⇒ aaabbbcc**c**
- **L(G) = { $a^nb^nc^n$ | n ≥ 1}**

Chomsky-1: (Wachsend) Kontext-sensitive Sprachen und NSPACE(n)

- **Theorem**
  - CSL = NSPACE(n)
- **Beweisidee**
  - CSL $\subseteq$ NSPACE(n)
    - NTM M rät die Ableitungen bis die Eingabewortläge erreicht wird
    - M akzeptiert, falls die Eingabe abgeleitet worden ist
    - Problem: nichthaltende wiederholende Berechnungspfade
    - Lösung:
      - M führt einen Zähler, der mit jeder geratenen Produktion um eins erhöht wird.
      - Überschreitet der Zähler $2^{c\,n}$, für geeignete Konstante c, dann verwirft M
    - Diese Lösung funktioniert, da höchstens 2^{cn} mögliche Zwischenschritte in der minimalen Ableitung einer kontextsensitiven Sprache vorkommen

Chomsky-1: (Wachsend) Kontext-sensitive Sprachen und NSPACE(n)

- **Theorem**
  - CSL = NSPACE(n)
- **Beweisidee: CSL $\supseteq$ NSPACE(n)**
  - Betrachte 1-Band-NTM mit linearem Platz, die genau auf der Eingabe rechnet und keine weiteren Symbole des Bandes bearbeitet
  - Die Berechnung der 1-Band-NTM wird rückwärts dargestellt
  - Für jedes Eingabesymbol a wird durch eine Variable A und ein Terminal a dargestellt
  - Am Anfang wird eine akzeptierende Konfiguration erzeugt durch entsprechende Regeln:
    Von Startsymbol S → “Q_{akz} _ _ _ _ _ .... _ “
  - Für jeden Übergang (q,a) → (q’,r,b) der NTM führe zwei Variablen
    - $X_{[(q,a) \to (q',r,b)]}$ und $Y_{[(q,a) \to (q',r,b)]}$ ein

- Nun werden für (q,a) → (q’,r,b) die folgenden Produktionen eingfügt
  - $B\ Q' \to X_{[(q,a) \to (q',r,b)]}Q'$
  - $X_{[(q,a) \to (q',r,b)]}Q' \to X_{[(q,a) \to (q',r,b)]}\ Y_{[(q,a) \to (q',r,b)]}$
  - $X_{[(q,a) \to (q',r,b)]}\ Y_{[(q,a) \to (q',r,b)]} \to Q\ Y_{[(q,a) \to (q',r,b)]}$
  - $Q\ Y_{[(q,a) \to (q',r,b)]} \to Q\ A$
- Übergänge (q,a) → (q’,l,b) werden analog bearbeitet
- Damit wird die Berechnung rückwärts dargestellt
- Am Ende werden die Terminale A → a eingesetzt: Die “Rückwärtsrechnung hält”

Komplexitätstheorie

Chomsky-0

15

Chomsky-0: Allgemeine Grammatiken und Rekursiv Aufzählbar

- **Definition Allgemeine (Chomsky-0) Grammatik**
  - Eine allgemeine (Chomsky-0) Grammatik ist ein Vierer-Tupel $G = (V,\Sigma,R,S)$, mit
    - Variablen V (Nichtterminale)
    - Terminalen Σ und
    - einer Regelmenge R
      - Jede Regel besteht aus einem Wort $\alpha \in (\Sigma \cup V)^*$ auf der linken Seite und einem Wort $\beta \in (\Sigma \cup V)^*$ auf der rechten
      - Jede Regel $\alpha \rightarrow \beta$ mit $\alpha,\beta \in (\Sigma \cup V)^*$ ist erlaubt
    - $S \in V$ ist die Startvariable
- **Theorem**
  - Die allgemeinen Grammatiken beschreiben genau die Menge der rekursiv aufzählbaren Sprachen

Chomsky-0: Allgemeine Grammatiken und Rekursiv Aufzählbar

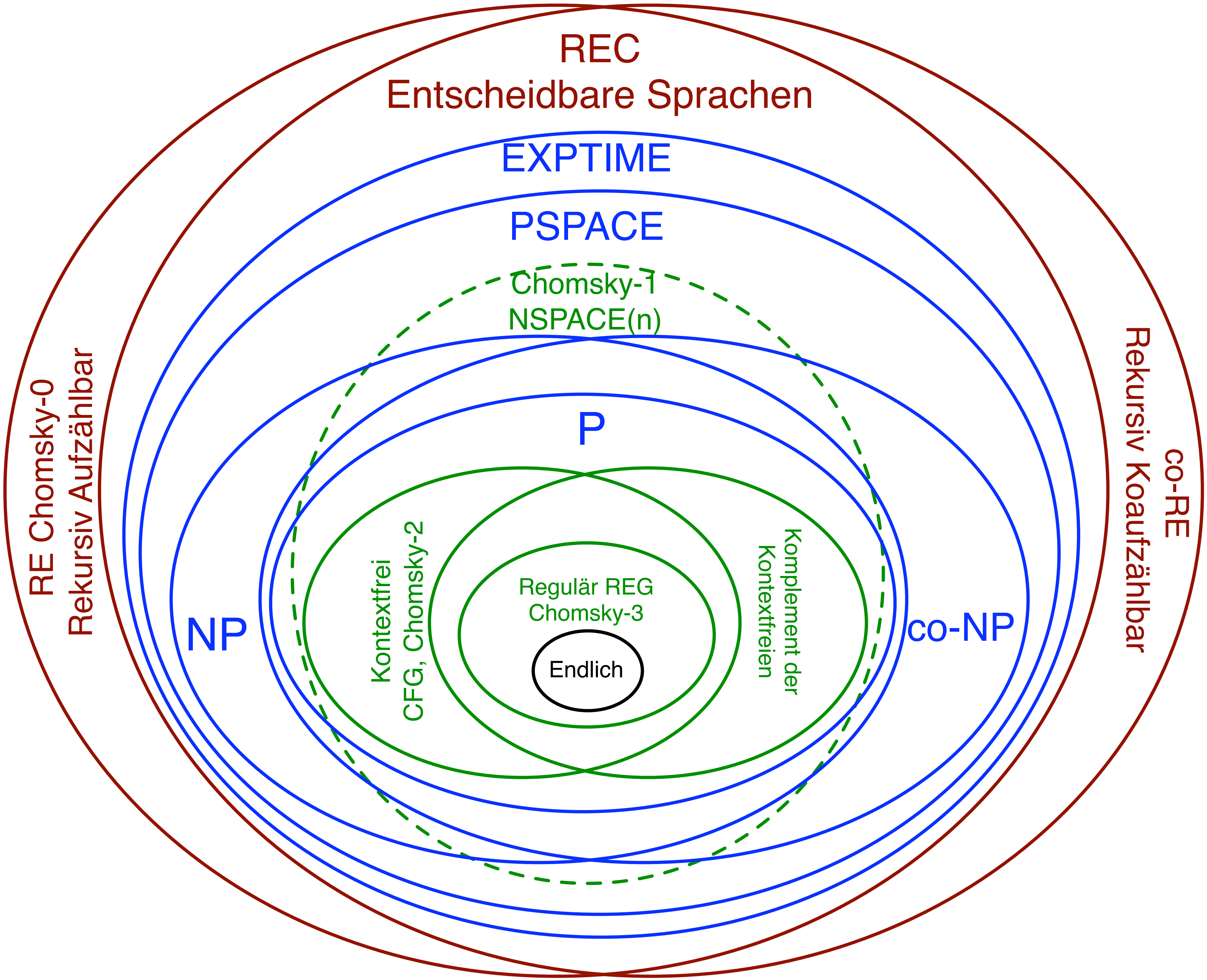
- **Theorem**
  - Die allgemeinen Grammatiken beschreiben genau die Menge der rekursiv aufzählbaren Sprachen
- **Beweisidee**
  - Hinrichtung:
    - NTM rät eine Ableitung und versucht dadurch aus dem Startsymbol das Eingabe-Wort abzuleiten
    - Falls keine Ableitung möglich, hält die NTM nie
  - Rückrichtung
    - Berechnung einer NTM wird umgekehrt
    - Ausgehend von der (eindeutigen) Endkonfiguration mit leerem Band (als Startsymbol) wird jeder Konfigurationsübergang durch lokale kontextsensitive Ableitungsregeln beschrieben
      - wie bei der Konstruktion für kontextsensitive Sprachen
    - Blank-Symbole am rechten Speicherrand können durch verkürzende kontextsensitive Regeln erzeugt und wieder freigegeben werden

Zusammenfassung

Rückblick

Zusammenfassung der Vorlesung

- **Formale Sprachen**
  - Chomsky-0 = Rekursiv Aufzählbar
  - Chomsky-1 = Kontextsensitiv = NSpace(n)
  - Chomsky-2 = Kontextfrei = PDA
    - Pumping-Lemma
  - Chomsky-3 = Regulär = NFA = DFA = Space(0)
    - Pumping-Lemma, Myhill-Nerode
- **Berechenbarkeitstheorie**
  - Abzählbar
    - Diagonalisierung
  - Rekursiv Aufzählbar = Akzeptor
    - Rekursiv Ko-Aufzählbar
    - Diagonalisierung, Abbildungsreduktion
  - Rekursiv = Entscheidbar
    - Diagonalisierung, Abbildungsreduktion, Turing-Reduktion
  - Beschreibungskomplexität
    - Unkomprimierbarkeit von Worten
- **Komplexitätstheorie**
  - DTM, NTM
    - Bänder, Konfigurationen
  - TIME(T(n)), NTIME(T(n))
  - P, NP
    - Abbildungsreduktion
    - NP-schwierig
    - NP-vollständig
    - SAT, 3-SAT, Clique, Subset-Sum, Hamiltonian
  - Approximationsalgorithmen
  - NSPACE(S(n))
    - $\subseteq$ SPACE($S(n)^2$) Satz von Savitch
  - PSPACE
    - PSPACE-schwierig
    - PSPACE-vollständig
    - QBF


REC
Entscheidbare Sprachen
EXPTIME
PSPACE
Chomsky-1
NSPACE(n)
P
RE Chomsky-0
Rekursiv Aufzählbar
co-RE
Rekursiv Koaufzählbar
NP
co-NP
Kontextfrei
CFG, Chomsky-2
Regulär REG
Chomsky-3
Endlich
Komplement der
Kontextfreien

20

5 Chomsky

Ende