Informatik III

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

Christian Schindelhauer
Wintersemester 2006/07
5. Vorlesung
09.11.2006
schindel@informatik.uni-freiburg.de

Äquivalenzklassen Definition und Beispiel

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

- **Definition**
  - Für eine Sprache $L \subseteq \Sigma^*$ bezeichnen wir zwei Worte $x,y \in \Sigma^*$ als **L-äquivalent**, geschrieben als
    $$x \equiv_L y,$$
    wenn für alle Worte $z \in \Sigma^*$ gilt
    $$x\,z \in L \;\Leftrightarrow\; y\,z \in L\ .$$
  - Die Menge aller zu x äquivalenten Worte wird als Äquivalenzklasse von x bezeichent:
    $$[x]_L = \{\, w \in \Sigma^* \mid w \equiv_L x \,\}$$
  - Die Anzahl der Äquivalenzklassen wird Index bezeichnet

- **1. Beispiel:**
  - Betrachte $B = \{0^n1^n \mid n \geq 0\}$
  - Worte in B:
    - $\{\varepsilon,01,0011,000111,...\}$
  - Äquivalente Worte:
  - $1 \equiv_B 10 \equiv_B 11 \equiv_B 100 \equiv_B 101 \equiv_B ...$
    - kein angehängtes Wort kann das Wort zu einem Wort der Sprache ergänzen
  - $01 \equiv_B 0011 \equiv_B ...$
    - weil nur ε angehängt werden darf damit das Wort in B ist
  - $0 \equiv_B 001 \equiv_B 00011 \equiv_B 0000111$
    - nur durch Anhängen von 1 kann ein Wort in B erzeugt werden

Äquivalenzklassen: Beispiel

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

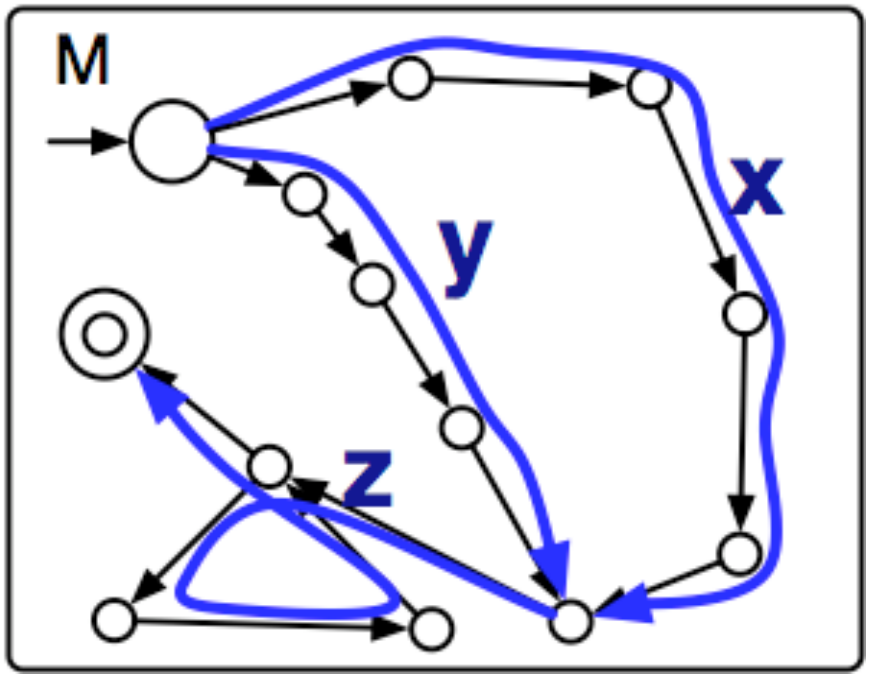
- **1. Beispiel:**
  - Betrachte $B = \{0^n1^n \mid n \geq 0\}$
  - Worte in B:
    - $\{\varepsilon,01,0011,000111,...\}$
  - Es gibt folgende Äquivalenzklassen:
  - $B_\varepsilon = \{\varepsilon\}$
  - $B\_ = \{0^n1^m \mid m>n\geq 0\} \cup L(\Sigma^*1\Sigma^*0\Sigma^*)$
  - $B_0 = \{0^n1^n \mid n\geq 1\}$
  - $B_1 = \{0^{n+1}1^n \mid n\geq 0\}$
  - $B_2 = \{0^{n+2}1^n \mid n\geq 0\}$
  - $B_3 = \{0^{n+3}1^n \mid n\geq 0\}$
  - ...
  - Der Index von B ist unendlich.

- **2. Beispiel:**
  - $A = L(\Sigma^* 0 \Sigma)$
  - $\Sigma = \{0,1\}$
  - $L(A) = \{\mathbf{0}0,\mathbf{0}1,0\mathbf{0}0,0\mathbf{0}1,1\mathbf{0}0,1\mathbf{0}1,...\}$
  - Äquivalente Worte:
    - $\varepsilon \equiv_A 1 \equiv_A 11 \equiv_A 011 \equiv_A 111 \equiv_A 0011 \equiv_A ...$
    - $01 \equiv_A 001 \equiv_A 101 \equiv_A 0001 \equiv_A ...$
    - $0 \equiv_A 10 \equiv_A 010 \equiv_A 110 \equiv_A 0010 \equiv_A ...$
    - $00 \equiv_A 000 \equiv_A 100 \equiv_A 0000 \equiv_A ...$
  - Äquivalenzklasen:
    - $[00]_A,[01]_A,[10]_A,[11]_A$
  - Der Index von A ist 4

Die Äquivalenzklassen beschreiben das Gedächtnis eines Automaten

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

- **Lemma:**
  - Falls $x \equiv_L y$ und $x \in L$, dann gilt $y \in L$
  - Falls $x \equiv_L y$, dann gilt
    $xa \equiv_L ya$ für alle $a \in \Sigma$
- **Beweis:**
  - folgt direkt aus der Definition
- **Hat ein DFA M den Zustand q mit zwei verschiedenen Worten x und y erreicht, dann gilt:**
  - $x \equiv_{L(M)} y$
- **Denn im folgenden wird M sich völlig identisch verhalten.**
- **Gibt es zwei Zustände im DFA, ab der für jedes folgende Teilwort das gleiche Ergebnis herauskommt, so können sie zu einem vereinigt werden.**
- **Idee: der Index beschreibt die Anzahl der Zustände des minimalen Automaten**

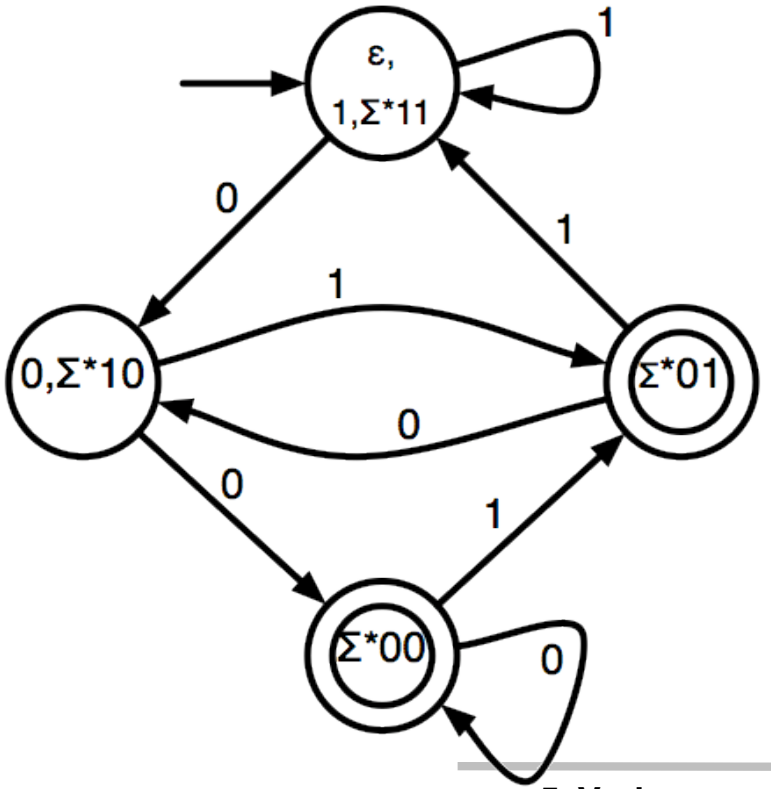


Der Satz von Myhill-Nerode

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

- **Theorem (Teil 1)**
  - **Ist der Index einer Sprache A gleich k, dann gibt es einen DFA mit k Zuständen der A akzeptiert.**
- Beweis
  - Konstruiere DFA $M= (Q, \Sigma, \delta, q_0, F)$ mit
    - $Q = \{[x]_A \mid x \in \Sigma^*\}$
    - $\delta([x]_A, a) = [xa]_A$
    - $q_0 = [\varepsilon]_A$
    - $F = \{[w]_A \mid w \in A\}$
  - Die Übergangsfunktion ist nach letztem Lemma wohl definiert
  - Nach dem letztem Lemma akzeptiert M das Wort w gdw. $w \in A$

- **2. Beispiel:**
  - $A = \Sigma^* 0 \Sigma$
  - Äquivalenzklasen:
    - $[00]_A, [01]_A, [10]_A, [11]_A$



Der Satz von Myhill-Nerode

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

- **Theorem (Teil 2)**
  - **Jeder DFA M mit k Zuständen akzeptiert eine Sprache mit Index $\leq$ k.**
- Betrachte einen Zustand q des DFA
  - Definiere: $L_q = \{w \in \Sigma^* \mid \delta(q_0,w) = q\}$
    - wobei $\delta(q,wa) := \delta(\delta(q,w),a)$ für $a \in \Sigma$

- **Behauptung: L_q beinhaltet nur äquivalente Worte bezüglich L = L(M)**
  - Beweis:
    - Für alle $x,y \in L_q$ und $z \in \Sigma^*$ gilt:
      - $\delta(q,z) = \delta(q_0,xz) = \delta(q_0,yz)$
    - Also $x \equiv_{L(M)} y$
- **Wenn jeder der k Zustände nur äquivalente Worte hat, dann kann es höchstens k Äquivalenzklassen geben**

Der minimale endliche deterministische Automat

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

- **Korollar**
  - Die Anzahl der Zustände eines minimalen endlichen Automats entspricht der Anzahl der Äquivalenzklassen der zugehörigen Sprache
- **Beweis:**
  - Es gibt einen DFA mit k=Index Zuständen
  - Jeder DFA hat mindestens k Zustände
  - Daher ist der durch die Äquivalenzklassen definierte Automat minimal.

- **Mit der Kenntniss der Äquivalenzklassen lässt sich ein minimaler DFA konstruieren**

Äquivalenzklassen zum Beweis der Nichtregularität

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

- **Theorem (Teil 2)**
  - **Jeder DFA M mit k Zuständen akzeptiert eine Sprache mit Index $\leq$ k.**

- Ist der Index einer Sprache unendlich, dann kann es keinen endlichen Automaten geben, der die Sprache akzeptiert.

- Aus der Kenntnis unendlicher Äquivalenzklassen lässt sich also die Nichtregularität beweisen

Beispiel

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

- **1. Beispiel:**
  - Betrachte $B = \{0^n1^n \mid n \geq 0\}$
  - Worte in B:
    - $\{\varepsilon, 01, 0011, 000111, ...\}$
  - Es gibt folgende Äquivalenzklassen:
  - $B_\varepsilon = \{\varepsilon\}$
  - $B_{\_} = \{0^n1^m \mid m>n\geq 0\} \cup L(\Sigma^*1\Sigma^*0\Sigma^*)$
  - $B_0 = \{0^n1^n \mid n\geq 1\}$
  - $B_1 = \{0^{n+1}1^n \mid n\geq 0\}$
  - $B_2 = \{0^{n+3}1^n \mid n\geq 0\}$
  - $B_3 = \{0^{n+3}1^n \mid n\geq 0\}$
  - ...
  - Der Index von B ist unendlich.

- **Also ist B nach dem Satz von Myhill-Nerode nicht regulär.**

Überblick: Kontextfreie Sprachen

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

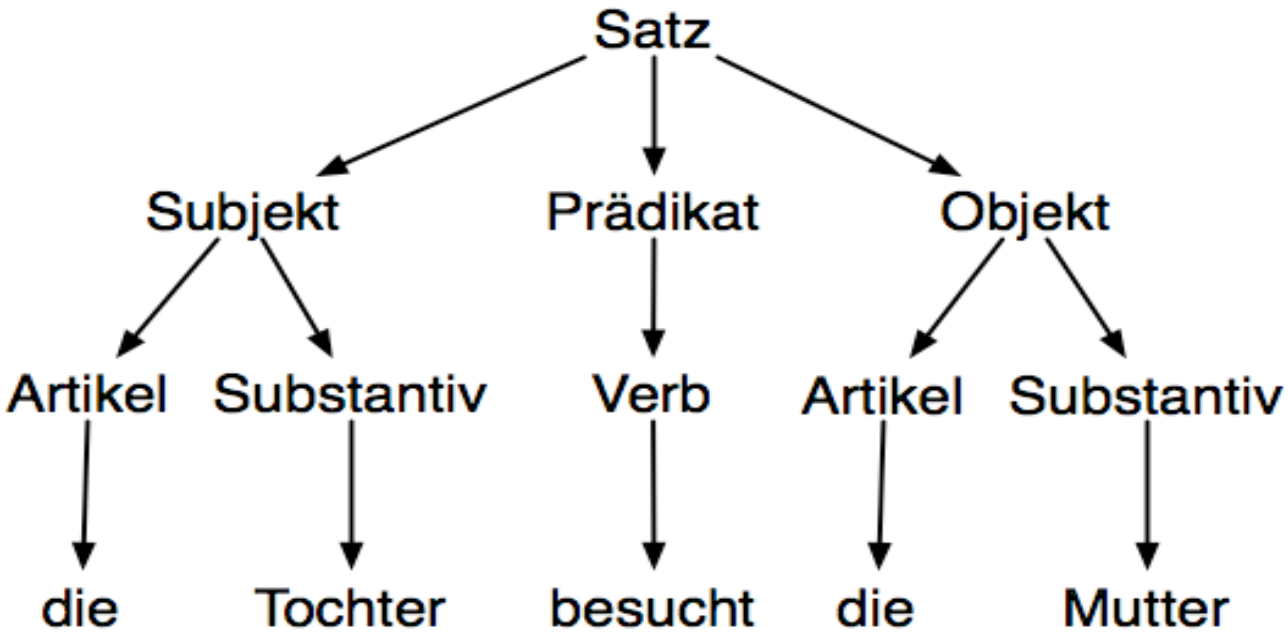
- **Formale Grammatik**
  - Einführung, Beispiele
  - Formale Definition
  - Das Wort-Problem
  - Chomsky Normalform
  - Cocke-Younger-Kasami-Algorithmus
- **Kellerautomaten**
  - Formale Definition
  - Beispiele
  - Äquivalenz zu kontextfreien Sprachen
- **Nichtkontextfreie Sprachen**
  - Pumping-Lemma für kontextfreie Sprachen

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

Kontextfreie Grammatik

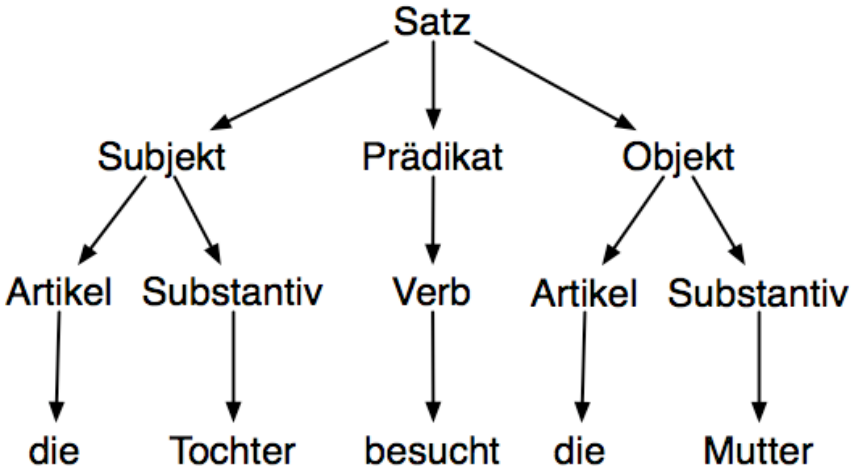
Grammatik - Was ist das?

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



Grammatik - Was ist das?

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



<SATZ> → **<SUBJEKT> <PRÄDIKAT> <OBJEKT>**

<SUBJEKT> → **<ARTIKEL> <SUBSTANTIV>**
<OBJEKT> → **<ARTIKEL> <SUBSTANTIV>**
<PRÄDIKAT> → **<VERB>**

<ARTIKEL> → `die`

<SUBSTANTIV> → `Mutter`
<SUBSTANTIV> → `Tochter`
<SUBSTANTIV> → `Professorin`
<SUBSTANTIV> → `Studentin`

<VERB> → `langweilt`
<VERB> → `prüft`
<VERB> → `belehrt`
<VERB> → `besucht`
<VERB> → `beleidigt`
<VERB> → `tötet`

Syntax und Semantik

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

<SATZ> →
<SUBJEKT> <PRÄDIKAT> <OBJEKT>

<SUBJEKT> → <ARTIKEL> <SUBSTANTIV>
<OBJEKT> → <ARTIKEL> <SUBSTANTIV>
<PRÄDIKAT> → <VERB>

<ARTIKEL> → die

<SUBSTANTIV> → Mutter
<SUBSTANTIV> → Tochter
<SUBSTANTIV> → Professorin
<SUBSTANTIV> → Studentin

<VERB> → langweilt
<VERB> → prüft
<VERB> → belehrt
<VERB> → besucht
<VERB> → beleidigt
<VERB> → tötet

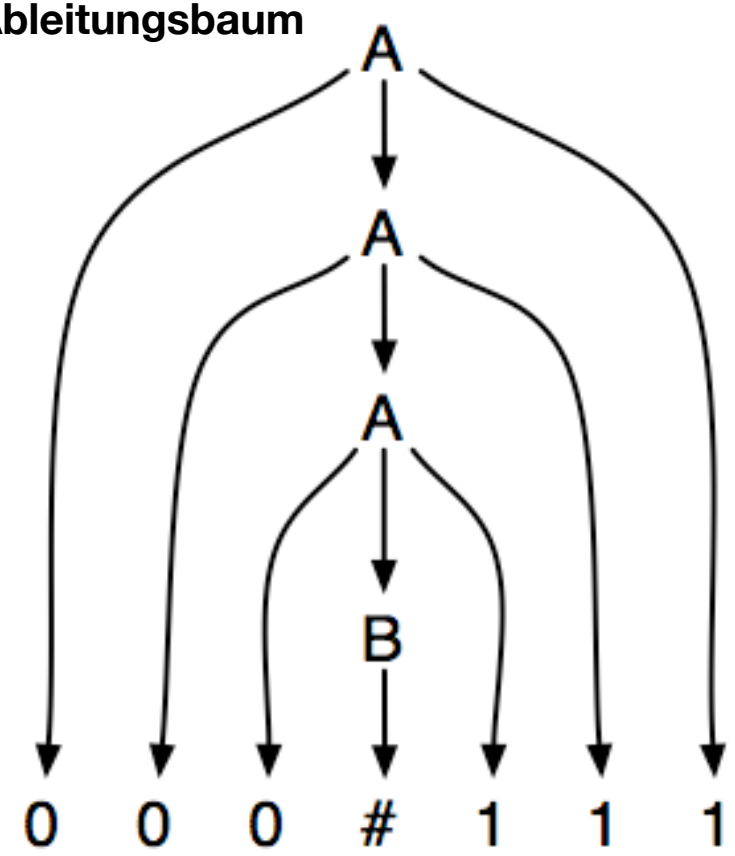
- **Mögliche Ableitungen:**
  - die Studentin besucht die Professorin
  - die Professorin belehrt die Studentin
  - die Professorin langweilt die Studentin
  - die Studentin beleidigt die Professorin
  - die Professorin prüft die Studentin
  - die Tochter besucht die Mutter
  - die Mutter besucht die Professorin
  - die Professorin belehrt die Mutter
  - die Mutter tötet die Professorin

Beispiele kontextfreier Grammatiken

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

- **Kontextfreie Grammatiken sind "Wortersetzer"**
  - A → 0A1
  - A → B
  - B → #
- **Ersetzungsregeln bestehen aus Produktionen**
- **Variablen, Terminalsymbole (Terminale)**
- **Startvariable: A**
  - A
  - 0A1
  - 00A11
  - 000A111
  - 000B111
  - 000#111 .... und Schluss
- **Das Wort 000#111 ist ein Wort der Grammatik**

Ableitungsbaum



Formale Definition einer kontextfreien Grammatik

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

- **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
    - $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_k \Rightarrow v$
- **Sprache der Grammatik G**
  - $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* v\}$
- **Die kontextfreien Sprachen werden durch kontextfreien Grammatiken erzeugt.**

Beispiel einer kontextfreien Grammatik

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

- **Definition**
  - Eine kontextfreie Grammatik ist ein Vierer-Tupel G=(V,Σ,R,S)
    - V: Variablen
    - Σ: Terminale
      - V und Σ sind disjunkt
    - R : Ersetzungsregeln
      - $A \rightarrow w$ mit $A \in V$, $w \in (V \cup \Sigma)^*$
    - $S \in V$: Startvariable
- **Ableitung**
  - Falls $A \rightarrow w$ in R, dann ist $uAv \Rightarrow uwv$
  - $u \Rightarrow^* v$, wenn es ein $k \geq 0$ gibt mit
    - $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_k \Rightarrow v$
- **Sprache der Grammatik G**
  - $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* v \}$

- **G = ({A,B}, {0,1,#}, R, A}**
- **R = {A → 0A1, A → B, B → #}**
- **Alternative Schreibweise:**
- **R = {A → 0A1 | B, B → #}**

- **A ⇒ 0A1**
  ⇒ 00A11
  ⇒ 000A111
  ⇒ 000B111
  ⇒ 000#111

Also:

A ⇒* 000#111

Damit ist das Wort 000#111 in der Sprache L(G)

L(G) = {#, 0#1, 00#11, 000#111, 0000#1111, ...}

Probleme mit kontextfreien Grammatiken

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

- **Mehrdeutigkeit:**
  - Beispiel: Grammatik ({S}, {+,x,3}, R, A}
  - R = {S → S + S,
    S → S x S,
    S → 3}
  - Betrachte das Wort: w = 3+3x3
    - S ⇒ S+S ⇒
      S + SxS ⇒* 3 + 3x3
    - S ⇒ SxS ⇒
      S+S x S ⇒* 3+3 x 3
  - Die Ableitung ist jeweils verschieden
    - damit ist auch die Interpretation verschieden

- **Epsilon-Regeln**
  - Beispiel: Grammatik ({S,A,B,C}, {a,b,c}, R, S }
  - R = {S → ABC | ε,
    A → BaB | ε,
    B → ε | c,
    C → BAbACB | ε }
  - S ⇒ A*BC*
    ⇒ A**BBAbACB** C
    ⇒ ABBAbA **BAbACB** B
    ⇒ ABBAbA *B*AbA **BAbACB** B B
    ⇒ ABBAbA **c**AbA BAbACB B B
    ⇒* εεεεεbεcεbεεεbεεεεεε
    = bcbb
  - Zwischenergebnisse können gigantisch lang werden
    - und dann wieder verschwinden

Ist ein Wort in der Sprache?

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

- **Die Entscheidung, ob ein Wort in einer kontextfreien Grammatik abgeleitet werden kann ist nicht trivial:**
  - Mehrdeutige Ableitung
  - Riesige Worte, die aufgrund der Epsilon-Regeln wieder verschwinden
- **Wenig aussichtsreiche Ansätze:**
  1. Versuche Terminalsymbole von links nach rechts zu erzeugen
     - Führt für viele Sprachen in Sackgassen
  2. Probiere alle Ableitungsmöglichkeiten aus
     - Grammatik mit Epsilon-Regeln: Unendliche Laufzeit
     - Grammatik ohne Epsilon-Regeln: Unglaublich schlechte (= exponentielle) Laufzeit
- **Die Lösungsstrategie:**
  - Chomsky-Normalform
    - Jede Grammatik lässt sich (praktisch) ohne Epsilon-Regeln in einer einfachen Form darstellen
  - Cocke-Younger-Kasami-Algorithms
    - Durch dynamische Programmierung lässt sich die Laufzeit reduzieren

Chomsky-Normalform

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

- **Definition**
  - Eine kontextfreie Grammatik ist in Chomsky-Normalform, falls jede Regel die Form
    - $A \rightarrow BC$ oder
    - $A \rightarrow a$ oder
    - $S \rightarrow \varepsilon$ hat
  - wobei
    - A,B,C,S sind Variablen
    - a ist ein Terminal
    - B,C sind nicht das Startsymbol,
    - S ist das Startsymbol.
- **Beispiel**
    - $G = (\{A,B,C,N,E\}, \{0,1,\#\}, R, S\}$
    - $R = \{S \rightarrow NC, N \rightarrow 0, S \rightarrow \#,$
      $A \rightarrow NC, C \rightarrow AE, E \rightarrow 1, A \rightarrow \#\}$

Chomsky-Normalform

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

➢ **Definition**

–Eine kontextfreie Grammatik ist in Chomsky-Normalform, falls jede Regel die Form

- A → BC oder
- A → a oder
- S → ε hat

–wobei

- A,B,C,S sind Variablen
- a ist ein Terminal
- B,C sind nicht das Startsymbol,
- S ist das Startsymbol.

➢ **Theorem**

– Jede kontextfreie Sprache kann in Chomsky-Normalform dargestellt werden.

➢ **Beispiel**

– Kontextfreie Grammatik

- G = ({A,B}, {0,1,#}, R, A}
- R = {A → 0A1, A → B, B → #}

– Grammatik mit gleicher Sprache in Chomski-Normalform

- G'= ({A,B,C,N,E}, {0,1,#}, R, S}
- R = {S → NC, N→ 0,S → #,
  A → NC, C → AE, E →1, A → #}

Ende der 5. Vorlesung

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

Informatik III

Christian Schindelhauer
schindel@informatik.uni-freiburg.de