Übungen zu Informatik I
Wintersemester 03/04

Übungsleiter: Dipl.-Inform. Tom Gelhausen

Organisatorisches

- Web-Login

Matrikelnummer: 1234567

Erstes Paßwort: Nachname
(Ändern!)

- eMails:
  - Absender, Betreff
  - Matrikelnummer
  - Rechtschreibung

é â

Grammatik: $G = (\Sigma, \mathcal{N}, \mathcal{P}, \mathcal{S})$ wobei

Σ	Menge der Terminale
$\mathcal{N}$	Menge der Nichtterminale
$\mathcal{S} \in \mathcal{N}$	Startsymbol
$(\Sigma \cup \mathcal{N}, \mathcal{P})$	Semi-Thue-System

Beachte:

$\Sigma \cap \mathcal{N} = \varnothing$

$\mathcal{V}^*$ = Menge *aller* Zeichenreihen über dem Zeichenvorrat $\mathcal{V}_G$

$\mathcal{V}^+ = \mathcal{V}^* \setminus \{\varepsilon\}$

Vokabular von G: $\mathcal{V}_G = \Sigma \cup \mathcal{N}$

Sprachschatz von G: $\mathcal{L}(G) = \{ w \in \Sigma^* : \mathcal{S} \Rightarrow^* w \}$

Phrase von G: $x \in \mathcal{V}^*$ mit $Y \Rightarrow^+ x$, $Y \in \mathcal{N}$

Charakterisierung des Sprachschatzes : alle terminalen Phrasen für $\mathcal{S}$

Zerteilung von $x \in \mathcal{V}^*$: findet heraus ob x Phrase von G ist

Beobachtung :

Ableitungsbaum beschreibt sowohl Zerteilung als auch Erzeugung von Phrasen

1.6 Äquivalenz von Grammatiken

Beispiel: Betrachte die Grammatiken G und G‘ mit Produktionenmengen

$\mathcal{P} = \{ \mathcal{S} \mapsto z \,;\, \mathcal{S} \mapsto z\mathcal{S}z \}$

$\mathcal{P}' = \{ \mathcal{S} \mapsto z \,;\, \mathcal{S} \mapsto \mathcal{S}zz \}$

Sprachschatz: $\mathcal{L}(G) = \mathcal{L}(G') = \{ z^{2n+1} : n \in \mathbb{N} \}$

schwach äquivalente Grammatiken G und G‘: L(G) = L(G‘)

andere Phrasenstruktur möglich, Darstellung über Klammern: ersetze Produktion $l \mapsto r$ durch $l \mapsto \langle r \rangle$

im **Beispiel**: $\langle z \langle z \langle z \rangle z \rangle z \rangle \leftrightarrow \langle \langle \langle z \rangle zz \rangle zz \rangle$

strukturäquivalente Grammatiken G und G‘:

auch nach expliziter Darstellung der Phrasenstruktur gilt $\mathcal{L}(G) = \mathcal{L}(G')$.

Beobachtung:

Ableitungsbäume beschreiben Phrasenstruktur

1.6 Chomsky-Hierarchie

CH-0 Grammatik: (lies: Chomsky 0) alle Produktionen haben allgemeine Form $\ell \mapsto r$; $\ell, r \in \mathcal{V}^*$ beliebig

- ☛ berechenbar, ob $\mathcal{S} \Rightarrow^* y$
- ☛ aber nicht immer, ob $\mathcal{S} \not\Rightarrow^* y$

$\varepsilon \rightarrow A$

CH-1 Grammatik: (auch kontextsensitive Grammatik)
alle Produktionen sind kontextsensitiv
(evtl. Ausnahme: $\mathcal{S} \mapsto \varepsilon$, dann aber nicht … $\mapsto$ …$\mathcal{S}$…!)

Kontextsensitive Produktion: $\underline{uAv} \mapsto urv$; $A \in \mathcal{N}$; $r \in \mathcal{V}^+$; $u,v \in \mathcal{V}^*$

äquivalente Definition: alle Produktionen sind beschränkt

Beschränkte Produktion: $\ell \mapsto r$; $\ell, r \in \mathcal{V}^*$; $1 \le |\ell| \le |r|$

$|abcde| = 5$

- ☛ In Ableitung $\mathcal{S} \Rightarrow^* x \Rightarrow y$ gilt $|y| \ge |x|$
- ☛ Entscheidbar, ob $\mathcal{S} \Rightarrow^* y$

CH-2 Grammatik: (auch **kontextfreie Grammatik**)
alle Produktionen sind kontextfrei

kontextfreie Produktion: $A \mapsto r \;;\; A \in \mathcal{N} \;;\; r \in \mathcal{V}^*$

String i;
i:=5

☛ Struktur korrekter Programme in Programmiersprachen

☛ Inhaltliche Konsistenzprüfung von Programmiersprachen ist kontextsensitiv!

CH-3 Grammatik: (auch **reguläre Grammatik**)

alle Produktionen sind linkslinear, terminierend oder ε-Produktionen

äquivalente Definition: alle Produktionen rechtslinear, terminierend oder ε-Produktionen

☛ nicht feststellbar, ob Klammerstrukturen stimmen

linkslineare Produktion: $A \mapsto Bx \;;\; A,B \in \mathcal{N} \;;\; x \in \Sigma$

rechtslineare Produktion: $A \mapsto xB \;;\; A,B \in \mathcal{N} \;;\; x \in \Sigma$

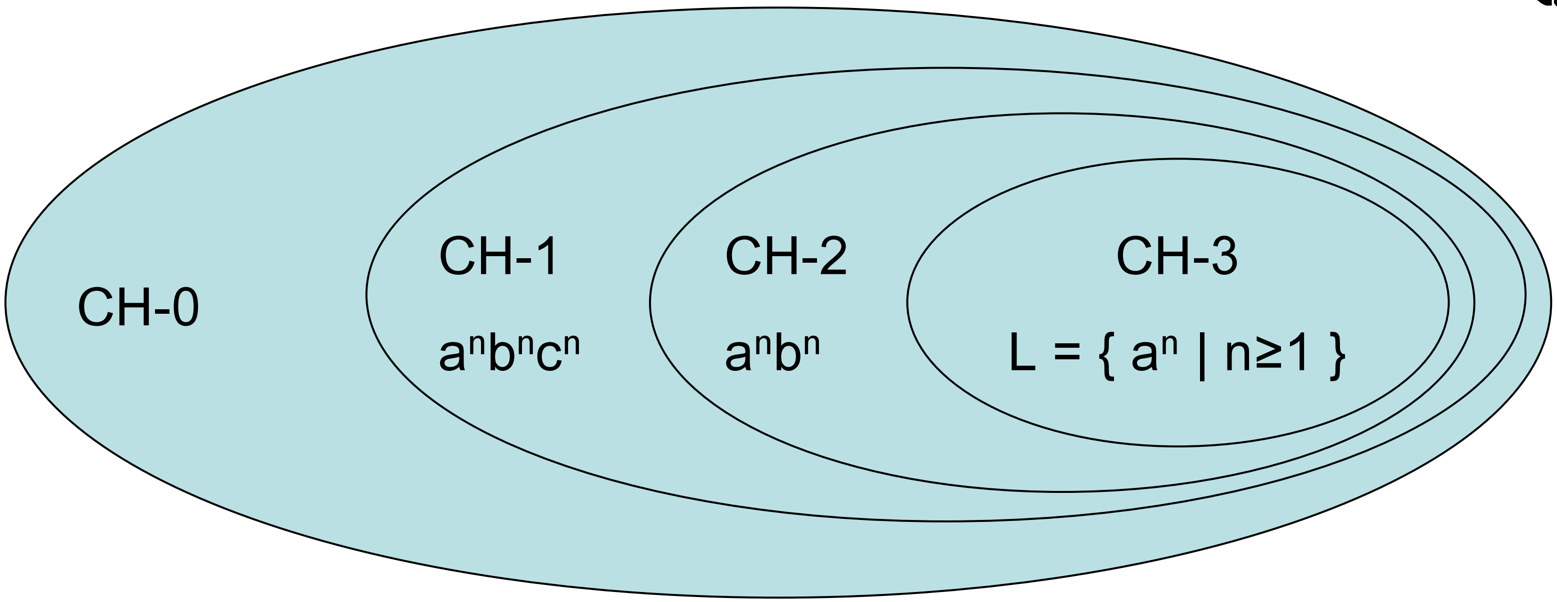
terminierende Produktion: $A \mapsto x \;;\; A \in \mathcal{N} \;;\; x \in \Sigma$

ε-Produktion: $A \mapsto \varepsilon \;;\; A \in \mathcal{N}$

1.6 Chomsky-Hierarchie

Beobachtung: In der Praxis sind reguläre und kontextfreie Grammatiken wichtig

aaaaaabbbbbbcccccc



$CH\text{-}3 \subset CH\text{-}2 \subset CH\text{-}1 \subset CH\text{-}0$

<u>CH-2 ⊂ CH-1 gilt nur</u>

- wenn wir ε-Produktionen in CH-2 ausschließen (→Praxisbezug?)
- oder die Ausnahme „$S \mapsto \varepsilon$" in CH-1 einschließen (→Randbedingung!)

"c"

- Reguläre Ausdrücke
  - Das leere Wort ε ist ein Regulärer Ausdruck
  - Jedes Terminalzeichen ist ein Regulärer Ausdruck (Bedeutung: an dieser Stelle muss genau das angegebene Terminal stehen)
  - Ist R ein Regulärer Ausdruck
    - (R) ist Regulärer Ausdruck (Klammerung, es gilt (R)=R)
    - R^* ist Regulärer Ausdruck (0 … bel. häufige Wiederholung)
    - R^+ ist Regulärer Ausdruck (1 … bel. häufige Wiederholung)
    - a^n a^5 R^n ist Regulärer Ausdruck (genau n-malige Wiederholung) baaaaac
    - [R] ist Regulärer Ausdruck (optionales Vorkommen von R)
  - Sind Q und R Regulärere Ausdrücke
    - QR ist Regulärer Ausdruck (Verkettung) a b ab
    - (Q|R) ist Regulärer Ausdruck (Bedeutung: an dieser Stelle steht <u>entweder</u> der Ausdruck Q <u>oder</u> der Ausdruck R)

ε-Produktion: Regelung

- Um den Sonderfall $\varepsilon \in \mathcal{L}(G)$ für CH-1/2/3 Grammatiken zuzulassen, definieren wir die **„$\mathcal{S}$ε-Produktion“ $\mathcal{S} \mapsto \varepsilon$ als zulässig** für CH-1, wenn diese Regel jedoch vorkommt, darf **keine Produktion $\mathcal{S}$ auf ihrer rechten Seite** beinhalten.

- Trick: Eine Grammatik mit $\mathcal{S}$ auf mindestens einer der rechten Seiten nachträglich um $\mathcal{S}$ε-Produktion erweitern:
  - Erzeuge neues Startsymbol S‘
  - Füge folgende Regeln ein:
    - $\mathcal{S}' \mapsto \varepsilon$
    - $\mathcal{S}' \mapsto \mathcal{S}$

1.6 Klassifikation der Produktionen – Überblick

Gegeben: Grammatik G = $(\Sigma, \mathcal{N}, \mathcal{P}, \mathcal{S})$

allgemeine Form: $\ell \mapsto r \;;\; \ell, r \in \mathcal{V}^*$

ε-Produktion: $\ell \mapsto \varepsilon,\ \ell \in \mathcal{V}^*$

beschränkte Produktion: $\ell \mapsto r \;;\; \ell, r \in \mathcal{V}^* \;;\; 1 \leq |\ell| \leq |r|$

kontextsensitive Produktion: $uAv \mapsto urv \;;\; A \in \mathcal{N} \;;\; r \in \mathcal{V}^+ \;;\; u, v \in \mathcal{V}^*$

kontextfreie Produktion: $A \mapsto r \;;\; A \in \mathcal{N} \;;\; r \in \mathcal{V}^*$

aABc

linkslineare Produktion: $A \mapsto Bx \;;\; A, B \in \mathcal{N} \;;\; x \in \Sigma$

aA$\underline{B}$ → aaB

rechtslineare Produktion: $A \mapsto xB \;;\; A, B \in \mathcal{N} \;;\; x \in \Sigma$

terminierende Produktion: $A \mapsto x \;;\; A \in \mathcal{N} \;;\; x \in \Sigma$

- G1 = { $\sum$, $\mathcal{N}$, $\mathcal{P}$, $\mathcal{S}$ }
  - $\mathcal{N}$ = { $\mathcal{S}$, **L**, **O**, **K** }
  - $\sum$ = { l, o }
  - $\mathcal{P}$ = {

$\mathcal{S}$	↦	**LOK**,	①
O	↦	**LOK** | lol,	②
Ll	↦	l**L**,	③
Lo	↦	loo,	④
K	↦	l }	⑤

  - „Kleinstes Wort“:
    - $\mathcal{S}$ ⇒ **LOK** ①
    - **LOK** ⇒ **L**lol**K** ②
    - **L**lol**K** ⇒ **L**loll ⑤
    - **L**loll ⇒ l**L**oll ③
    - l**L**oll ⇒ llooll ④

- $\mathcal{P} = \{$ $S \mapsto$ **LOK**, ①
  - O $\mapsto$ **LOK**, ②
  - O $\mapsto$ 1o1
  - **L**1 $\mapsto$ 1**L**, ③
  - **L**o $\mapsto$ 1oo, ④
  - **K** $\mapsto$ 1 $\}$ ⑤

$L(G_1) = \{1^n 0^n 1^n \mid n \geq 2\}$

- Noch ein Wort:
  - S ⇒ **LOK** ①
  - **LOK** ⇒ **LLOKK** ⇒ LLLOKKK ②
  - **LLOKK** ⇒ **LL**1o1**KK** ②
  - **LL**1o1**KK** ⇒ **LL**1o11**K** ⇒ **LL**1o111 ⑤ (2x)
  - **LL**1o111 ⇒ **L**1**L**o111 ③
  - **L**1**L**o111 ⇒ 1**LL**o111 ③
  - 1**LL**o111 ⇒ 1**L**1oo111 ④
  - 1**L**1oo111 ⇒ 11**L**oo111 ③
  - 11**L**oo111 ⇒ 111ooo111 ④

Beispiel G_1 : der Chomsky-Typ

- $\mathcal{P}$ = {

$\mathcal{S}$	$\mapsto$ **LOK**,		①	kontextfrei
O	$\mapsto$ **LOK** | lol,		②	kontextfrei, term., beschränkt
Ll	$\mapsto$ l**L**,		③	
Lo	$\mapsto$ loo,		④	
K	$\mapsto$ l	}	⑤	

- Erinnerung:

ε-Produktion: $\ell \mapsto \varepsilon,\ \ell \in \mathcal{V}^*$

beschränkte Produktion: $\ell \mapsto r\ ;\ \ell, r \in \mathcal{V}^*\ ;\ 1 \le |\ell| \le |r|$

kontextsensitive Produktion: $uAv \mapsto urv\ ;\ A \in \mathcal{N}\ ;\ r \in \mathcal{V}^+\ ;\ u, v \in \mathcal{V}^*$

kontextfreie Produktion: $A \mapsto r\ ;\ A \in \mathcal{N}\ ;\ r \in \mathcal{V}^*$

linkslineare Produktion: $A \mapsto Bx\ ;\ A,B \in \mathcal{N}\ ;\ x \in \Sigma$

rechtslineare Produktion: $A \mapsto xB\ ;\ A,B \in \mathcal{N}\ ;\ x \in \Sigma$

terminierende Produktion: $A \mapsto x\ ;\ A \in \mathcal{N}\ ;\ x \in \Sigma$

Beispiel G_1 : der Chomsky-Typ

- Feststellungen: —▷ Typ CH-1
  - Produktion ① ist beschränkt
  - Produktion ② ist beschränkt
  - Produktion ③ ist beschränkt und kontextsensitiv
  - Produktion ④ ist beschränkt und kontextsensitiv
  - Produktion ⑤ ist beschränkt

- Erinnerung:

CH-0 Grammatik: alle Produktionen haben allgemeine Form
$\ell \mapsto r$; $\ell, r \in \mathcal{V}^*$ beliebig

CH-1 Grammatik: (auch **kontextsensitive Grammatik**)
alle Produktionen sind kontextsensitiv

CH-2 Grammatik: (auch **kontextfreie Grammatik**)
alle Produktionen sind kontextfrei

CH-3 Grammatik: (auch **reguläre Grammatik**)
alle Produktionen sind linkslinear, terminierend oder ε-Produktionen

Markov-Algorithmen – Beispielaufgabe

- Geben Sie einen Markov-Algorithmus an, der einen Klammerausdruck auf Korrektheit prüft. Verdeutlichen Sie den Algorithmus anhand der folgenden Beispiele:
  - 1. ()(()())
  - 2. (()
  - 3. ()))

  Anmerkung: dieses Beispiel lässt sich auch leichter lösen, hier jedoch eine Markov-Version mit Schiffchen!

- Lösung:
  - $\alpha(\Rightarrow (\alpha$ — *öffnende Klammer überspringen*
  - $(\alpha) \Rightarrow \alpha$ — *korrektes Klammernpaar erkannt, entfernen*
  - $(\alpha \Rightarrow.$ Fehler1 — *zu viele öffnende Klammern (keine schließende mehr)*
  - $\alpha) \Rightarrow.$ Fehler2 — *zu viele schließende Klammern (keine öffnende mehr)*
  - $\alpha \Rightarrow.$ Korrekt — *alle Klammern entfernt, keine übrig*
  - $\varepsilon \Rightarrow \alpha$ — *Schiffchen erzeugen*

Markov-Algorithmen – Beispielaufgabe

- Regeln:
  - $\alpha($ $\Rightarrow$ $(\alpha$ — *öffnende Klammer überspringen*
  - (α) $\Rightarrow$ α — *korrektes Klammernpaar erkannt, entfernen*
  - $(\alpha$ $\Rightarrow.$ Fehler1 — *zu viele öffnende Klammern (keine schließende mehr)*
  - $\alpha)$ $\Rightarrow.$ Fehler2 — *zu viele schließende Klammern (keine öffnende mehr)*
  - α $\Rightarrow.$ Korrekt — *alle Klammern entfernt, keine übrig*
  - ε $\Rightarrow$ α — *Schiffchen erzeugen*
- Aufgabe: $()(()())$

$\Rightarrow^6$ $\alpha()(()())$

$\Rightarrow^1$ $(\alpha)(()())$

$\Rightarrow^2$ $\alpha(()())$

$\Rightarrow^1$ $(\alpha()())$

$\Rightarrow^1$ $((\alpha)())$

$\Rightarrow^2$ $(\alpha())$

$\Rightarrow^1$ $((\alpha))$

$\Rightarrow^2$ (α)

$\Rightarrow^2$ α

$\Rightarrow^5.$ Korrekt

Markov-Algorithmen – Beispielaufgabe

- Regeln:
  - $\alpha($ $\Rightarrow$ $(\alpha$ — *öffnende Klammer überspringen*
  - (α) $\Rightarrow$ α — *korrektes Klammernpaar erkannt, entfernen*
  - $(\alpha$ $\Rightarrow.$ Fehler1 — *zu viele öffnende Klammern (keine schließende mehr)*
  - $\alpha)$ $\Rightarrow.$ Fehler2 — *zu viele schließende Klammern (keine öffnende mehr)*
  - α $\Rightarrow.$ Korrekt — *alle Klammern entfernt, keine übrig*
  - ε $\Rightarrow.$ α — *Schiffchen erzeugen*

- Aufgabe: (()

$\Rightarrow^6$	$\alpha(()$
$\Rightarrow^1$	$(\alpha()$
$\Rightarrow^1$	$((\alpha)$
$\Rightarrow^2$	$(\alpha$
$\Rightarrow^3.$	Fehler1

Markov-Algorithmen – Beispielaufgabe

- Regeln:
  - α(⇒ (α — *öffnende Klammer überspringen*
  - (α) ⇒ α — *korrektes Klammernpaar erkannt, entfernen*
  - (α ⇒. Fehler1 — *zu viele öffnende Klammern (keine schließende mehr)*
  - α) ⇒. Fehler2 — *zu viele schließende Klammern (keine öffnende mehr)*
  - α ⇒. Korrekt — *alle Klammern entfernt, keine übrig*
  - ε ⇒. α — *Schiffchen erzeugen*

- Aufgabe: ()))

$\Rightarrow^6$	α()))
$\Rightarrow^1$	(α)))
$\Rightarrow^2$	α)))))
$\Rightarrow^4$.	Fehler2)))

Übungsblätter

- Sowie die Musterlösungen finden Sie unter

http://www.infoeins.de/uebungsblaetter.php