Grundlagen Theoretischer Informatik 2

WiSe 2009/10 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Grundlagen Theoretischer Informatik 2 Gesamtübersicht

- Organisatorisches; Einführung
- Ersetzungsverfahren: Grammatiken und Automaten
- Rekursionstheorie-Einführung
- Komplexitätstheorie-Einführung

Organisatorisches

Vorlesungen DI 8.15-9.45 im H 11

Übungsbetrieb in Form von einer Übungsgruppe
BEGINN: vorlesungsartig in der ersten Semesterwoche
FR 10-12, HZ 204

Dozentensprechstunde DO 13-14 in meinem Büro H 410 (4. Stock)

Mitarbeitersprechstunde (Stefan Gulan) FR 12-13 H 413

Tutorensprechstunde MO 13.30-14.30 & MI 14-15 H 407/H412

Benotung

Die Note ergibt sich ausschließlich aus der gezeigten Leistung bei der Abschlussprüfung.

Die Abschlussprüfung erfolgt schriftlich am 23.2.2010 bzw. mündlich (n.V.) für BEd-Studenten.

Zulassungskriterien

Um an der Abschlussprüfung teilnehmen zu dürfen, wird vorausgesetzt:

(1) Mindestens zweimaliges erfolgreiches Vorrechnen von Hausaufgaben sowie
(2) 40% der Hausaufgabenpunkte (Zweier-/ Dreiergruppenabgabe möglich) sowie
(3) Bestehen der Zwischenklausur.

Abgabe von Hausaufgaben

Diese sollte in Gruppen zu 2-3 Personen erfolgen.

Abgabeschluss ist MI 14 Uhr, im mit GTI 2 beschrifteten Kasten im 4. Stock vor dem gemeinsamen Sekretariat von Prof. Näher.
Verspätete Abgaben gelten als nicht abgegeben und werden dementsprechend mit 0 Punkten bewertet

In der darauffolgenden Übung (am Freitag) werden die korrigierten Lösungen zurückgegeben.
Wir nutzen die Zeit (i.d.R. zwei Tage) auch dazu, uns die Namen derjenigen herauszusuchen, die gewisse Aufgaben dann vorzurechnen haben.
Wer die Aufgaben auf Aufforderung **nicht erläuternd vorrechnen** kann, bekommt die durch Hausaufgabenabgabe erzielten Punkte für das gesamte Aufgabenblatt abgezogen.
Selbstverständlich gilt das Vorrechnen dann nicht als “erfolgreich” im Sinne der vorigen Folie.
Sollte ein Vorrechnen nicht erfolgreich sein, wird i.d.R. ein weiterer Studierender Gelegenheit zum Vorrechnen bekommen.

Die Lösungen sind handschriftlich anzufertigen; weder Schreibmaschinen- noch Computerausdrucke werden akzeptiert, erst recht keine Kopien.

Probleme ? Fragen ?

Klären Sie bitte Schwierigkeiten mit Vorlesungen oder Übungen möglichst umgehend in den zur Verfügung gestellten Sprechzeiten.

In der Tutorensprechstunde stehen Ihnen (alternierend) Studenten höherer Semester zu Rückfragen bereit.

Wir helfen Ihnen gerne!
. . . wir sind aber keine Hellseher, die Ihnen Ihre Schwierigkeiten an der Nasenspitze ansehen. . .

Chomsky-Normalform für kontextfreie Grammatiken

Eine kontextfreie Grammatik $G = (N, T, P, S)$ ist in Chomsky-Normalform, falls alle ihre Regeln die folgende Form haben:

$$A \longrightarrow BC \text{ oder } A \longrightarrow a$$

für $A, B, C \in N$ und $a \in T$.
Kurz: $P \subset (N \times N^2) \cup (N \times T)$

Satz: Zu jeder kontextfreien Grammatik G gibt es eine Grammatik G' in Chomsky-Normalform mit $L(G') = L(G) \backslash \{\lambda\}$.

Hinweis: Mit dem Verfahren aus der vorigen Vorlesung können wir zunächst eine zu G äquivalente λ-freie kfG $\tilde{G}$ konstruieren mit $L(\tilde{G}) = L(G) \backslash \{\lambda\}$.

1. Lange rechte Seiten mit Terminalzeichen

Wir nennen eine Satzform *gemischt*, wenn sie sowohl Terminal- als auch Nichtterminalzeichen enthält.

Lemma: Zu jeder kfG G lässt sich eine äquivalente kfG G' konstruieren, bei welcher rechte Regelseiten mit Terminalzeichen die Länge höchstens Eins haben. Mit G ist auch G' λ-frei.

Damit enthält G' auch keine gemischten rechten Regelseiten.

1. Führe für jedes Terminalzeichen a ein spezielles Nichtterminal X_a ein.
2. Führe neue Regeln $X_a \longrightarrow a$ ein.
3. Ersetze a in rechter Regelseite w durch X_a, falls $w \neq a$.

$\Longrightarrow$ Nur noch Regeln der Formen

$$X \longrightarrow a \text{ mit } a \in T \text{ oder } X \longrightarrow w \text{ mit } w \in N^*$$

2. Lange rechte Seiten mit Nichtterminalzeichen

Entfernung aller Regeln $X \rightarrow B_1 \ldots B_k$ mit $k \geq 3$

Dazu neue Nichtterminale $C_1, C_2, \ldots, C_{k-2}$ und neue Regeln:

$$
\begin{array}{rcl}
X & \rightarrow & B_1 C_1 \\
C_1 & \rightarrow & B_2 C_2 \\
 & & \ldots \\
C_{k-2} & \rightarrow & B_{k-1} B_k
\end{array}
$$

$\Longrightarrow$ Nur noch Regeln der Formen

$$X \rightarrow a \text{ mit } a \in T \text{ oder } X \rightarrow B_1 B_2 \text{ mit } B_1, B_2 \in N$$

oder

$$X \rightarrow B \text{ mit } B \in N$$

Die letztgenannten Kettenregeln zu entfernen können wir bereits (letzte VL)!

Der Weg zur Chomsky-Normalform Zusammenfassung:

0. Eliminiere λ-Regeln.
1. Behandle zu lange rechte Seiten mit Terminalzeichen.
2. Behandle zu lange rechte Seiten mit Nichtterminalzeichen.
3. Entferne Kettenregeln (durch Abkürzungen).

Beachte:

Schritt i zerstört nicht die durch Schritt $j < i$ gewonnenen Eigenschaften!

Ein Beispiel: Gegeben sei $G = (\{S, T\}, \{a, b, c\}, P, S)$ mit

$$P = \{S \to aT|TbT|T, T \to S|c\}$$

1) neue Nichtterminale X_a, X_b, neue Regeln $X_a \to a$, $X_b \to b$

Ersetzung: $S \to a$Tdurch $S \to X_aT$; $S \to TbT$ durch $S \to TX_bT$

neue Regelmenge:

$$\{S \to X_aT|TX_bT|T, T \to S|c, X_a \to a, X_b \to b\}$$

Ein Beispiel (Forts.):

2) Neues Nichtterminal C

Ersetzung: $S \longrightarrow \mathrm{T}\mathrm{X}_\mathrm{b}\mathrm{T}$durch $S \longrightarrow \mathrm{T}\mathrm{C}$ und $C \longrightarrow \mathrm{X}_\mathrm{b}\mathrm{T}$

neue Regelmenge

$$\{S \longrightarrow \mathrm{X}_\mathrm{a}\mathrm{T}|\mathrm{T}\mathrm{C}|\mathrm{T}, \mathrm{T} \longrightarrow S|c, \mathrm{X}_\mathrm{a} \longrightarrow \mathrm{a}, \mathrm{X}_\mathrm{b} \longrightarrow \mathrm{b}, \mathrm{C} \longrightarrow \mathrm{X}_\mathrm{b}\mathrm{T}\}$$

Ein Beispiel (Forts.):

Schritt 3: Eliminiere die beiden Kettenregeln $S \rightarrow T$ und $T \rightarrow S$.
Zuerst $S \rightarrow T$: füge die beiden Regeln $S \rightarrow S$ und $S \rightarrow c$ hinzu.
Bei $T \rightarrow S$ kommen die Regeln $T \rightarrow X_a T$, $T \rightarrow T$ und $T \rightarrow TC$ hinzu.

Wir haben jetzt die Regelmenge

$$\{S \rightarrow X_a T|TC|T|S|c, T \rightarrow S|c|X_a T|TC|T, X_a \rightarrow a, X_b \rightarrow b, C \rightarrow X_b T\}$$

Neue Regeln entstehen nicht mehr,
$\leadsto$ Entferne die vier Regeln $S \rightarrow S|T$ und $T \rightarrow S|T$.
$\leadsto$ Zu G äquivalente Grammatik $G' = (\{S, T, X_a, X_b, C\}, \{a, b, c\}, P', S)$ in Chomsky-Normalform, wobei:

$$P' = \{S \rightarrow X_a T|TC|c, T \rightarrow c|X_a T|TC, X_a \rightarrow a, X_b \rightarrow b, C \rightarrow X_b T\}$$

Satz: Es gibt einen Algorithmus, der zu jeder vorgelegten kfG $G = (N, T, P, S)$ und jedem $w \in T^*$ entscheidet, ob $w \in L(G)$ gilt.

Beweis: Wird $w = \lambda$ angefragt, so berechnen wir N_1 wie im Beweis I in der vorigen VL. Es gilt: $\lambda \in L(G)$ gdw. $S \in N_1$.
Wird $w \neq \lambda$ angefragt, so überführen wir G in eine Chomky-Normalform kfG G' mit der Eigenschaft $w \in L(G)$ gdw. $w \in L(G')$.
Da G' keine λ-Regeln enthält, kann man "$w \in L(G)$?" durch Berechnen aller aus G' ableitbaren Satzformen der Länge höchstens $\ell(w)$ beantworten.

Cocke, Younger und Kasami haben gezeigt, wie man die Komplexität des beschriebenen Verfahrens durch dynamisches Programmieren erheblich vermindern kann.

Das Verfahren von Cocke, Younger und Kasami (*CYK-Algorithmus*)

Satz: Ist eine kfG G in Chomsky-Normalform fixiert, so lässt sich die Frage "$w \in L(G)$?" in einer Zeit beantworten, die sich durch ein kubisches Polynom in $\ell(w)$ abschätzen lässt.

Beweis: Betrachte $w = a_1 \ldots a_n$.
Lege 2–dimensionale dreiecksförmige Tabelle T an mit Nonterminalmengen als Einträgen, so dass $T[i,j] = \{A \in N \mid A \overset{*}{\Rightarrow} a_i \ldots a_j\}$. T kann man wie folgt "von oben nach unten" berechnen:

$$T[i,i] = \{A \in N \mid A \to a_i \in P\}$$

$$T[i,j] = \{A \in N \mid A \to BC \in P \land \exists i \leq k < j : B \in T[i,k], C \in T[k+1,j]\}$$

Die CYK-Tabelle

a_1	a_2	$\dots$	a_n
$T[1,1]$	$T[2,2]$	$\dots$	$T[n,n]$
$T[1,2]$	$T[2,3]$	$\dots$	
$\vdots$	$\vdots$		
$T[1,n-1]$	$T[2,n-1]$		
$T[1,n]$			

$a_1 \dots a_n \in L(G)$ gdw. das Startsymbol erscheint in $T[1,n]$.

Der CYK-Algorithmus

FOR $j = 1, \ldots, n$

$\quad T[i, i] = \{A \in N \mid A \to a_i \in P\}$

FOR $j = 1, \ldots, n - 1$

$\quad$ FOR $i = 1, \ldots, n - j$

$\quad\quad T[i, j + i] = \emptyset$

$\quad\quad$ FOR $k = i, \ldots, j + i - 1$

$\quad\quad\quad$ IF $\exists A \to BC \in P : B \in T[i, k], C \in T[k + 1, j + i]$

$\quad\quad\quad$ THEN $T[i, j + i] = T[i, j + i] \cup \{A\}$

Ein Beispiel

$S \longrightarrow AX$, $X \longrightarrow SB$, $S \longrightarrow c$, $A \longrightarrow a$, $B \longrightarrow b$ erzeugt $L = \{a^n c b^n \mid n \geq 0\}$.

a	a	c	b	b
A	A	S	B	B
		X		
	S			
	X			
S				

Ein **binärer Wurzelbaum** ist gegeben durch ein Tripel $B = (V, \phi, r)$ mit ausgezeichneter *Wurzel* $r \in V$ und einer *Vater-Abbildung* $\phi : V \setminus \{r\} \to V$ mit der Eigenschaft

$$\forall v \in V : \#\underbrace{\{u \in V \mid \phi(u) = v\}}_{\kappa(v):=} \leq 2$$

$\kappa(v)$ liefert also die *Kinder* von v.
Knoten v mit $\kappa(v) = \emptyset$ heißen *Blätter*.

Lemma: Der Ableitungsbaum eines jeden von einer kontextfreien Grammatik in Chomsky-Normalform akzeptierten Wortes ist als binärer Wurzelbaum auffassbar.

Die *Höhe* eines binären Wurzelbaumes $B = (V, \phi, r)$ ist gegeben durch

$$h(B) = \max_{v \in V \setminus \{r\}} \{k \in \mathbb{N} \mid \phi^k(v) = r\}.$$

Lemma: Hat ein binärer Wurzelbaum B mehr als 2^h Blätter, so gilt $h(B) > h$.

Beweis: Wir zeigen die *Kontraposition* per Induktion:

Gilt $h(B) \leq h$, so hat $B = (V, \phi, r)$ höchstens 2^h viele Blätter.

$h = 0$ ist trivial.

Es gelte $h > 0$. Daher gilt $\kappa(r) \neq \emptyset$.

Betrachte $r' \in \kappa(r)$. $V' := \{v \in V \mid \exists k \in \mathbb{N} : \phi^k(v) = r'\}$.

ϕ' sei die Einschränkung von ϕ auf $V' \setminus \{r'\}$.

$B' = (V', \phi', r')$ ist ein binärer Wurzelbaum der Höhe höchstens $h - 1$.

Auf B' ist die Induktionshypothese anwendbar: B' hat höchstens 2^{h-1} viele Blätter.

$\rightsquigarrow$ B hat höchstens $\#\kappa(r) \cdot 2^{h-1} \leq 2^h$ viele Blätter.

Lemma: Ist $G = (N, T, P, S)$ eine kfG in Chomsky-Normalform und ist $w \in L(G)$ mit $\ell(w) > 2^{\#N}$, so gilt für jeden Ableitungsbaum von w bzgl. G, dass es einen Weg von S zu einem Blatt gibt, auf dem mehr als $\#N$ viele Nichtterminalzeichen ersetzt werden.

Beweis: Nach dem vorigen Lemma hat der unterliegende binäre Wurzelbaum die Höhe größer als $\#N$. Es gibt also einen Weg in besagtem Ableitungsbaum von der mit S beschrifteten Wurzel zu einem Blatt, dem mehr als $\#N$ Regelanwendungen entspricht $\rightsquigarrow$ Behauptung.

Folg.: Auf besagtem Weg von der Wurzel zum Blatt im Ableitungsbaum von w finden wir also nach dem Schubfachprinzip zwei Regelanwendungen $A \to v$ und $A \to u$ mit gleicher linker Seite.

Ein Pumping-Lemma für Typ-2

Satz: Zu jeder kfS L gibt es eine Konstante $n > 0$, sodass jedes Wort $w \in L$ mit $\ell(w) \geq n$ als Konkatenation $w = uvxyz$ dargestellt werden kann mit geeigneten u, v, x, y, z mit folgenden Eigenschaften:

1. $\ell(v) > 0$ oder $\ell(y) > 0$;
2. $\ell(vxy) \leq n$;
3. $\forall i \geq 0 : uv^ixy^iz \in L$.

Beweis: Mit L ist auch $L \setminus \{\lambda\}$ vom Typ 2. Sei also o.E. L λ-frei.
L werde durch kfG $G = (N, T, P, S)$ in Chomsky-Normalform beschrieben.
Es sei $n = 2^{\#N}$.
Gibt es kein $w \in L$ mit $\ell(w) \geq n$, so stimmt die Aussage leer.
Sonst wähle ein $w \in L$ mit $\ell(w) \geq n$.
Nach obiger Folgerung gibt es zwei Regelanwendungen $A \to u'$ und $A \to v'$ auf einem Weg in einem Ableitungsbaum von w.
$\rightsquigarrow$ Es gibt Linksableitung

$$S \overset{*}{\Rightarrow} uA\zeta \overset{*}{\Rightarrow} uvA\eta\zeta \overset{*}{\Rightarrow} uvx\eta\zeta \overset{*}{\Rightarrow} uvxy\zeta \overset{*}{\Rightarrow} uvxyz.$$

Wegen Chomsky-NF (keine Kettenregeln oder λ-Regeln) gilt $\ell(v) > 0$ oder $\ell(y) > 0$.
$\ell(vxy) \leq n$ ergibt sich aus dem Beweis der Folgerung sowie wegen Chomsky-NF.
Daher gibt es auch Ableitungen

$$S \overset{*}{\Rightarrow} uA\zeta \overset{*}{\Rightarrow} ux\zeta \overset{*}{\Rightarrow} uxz$$

und allgemeiner

$$S \to uA\zeta \overset{*}{\Rightarrow} uvA\eta\zeta \overset{*}{\Rightarrow} uvvA\eta\eta\zeta \overset{*}{\Rightarrow} uv^iA\eta^i\zeta \overset{*}{\Rightarrow} uv^ixy^iz.$$

Ein Beispiel zur Anwendung des Pumping-Lemmas:

$$L = \{a^k b^k c^k \mid k \in \mathbb{N}\} \notin \mathcal{L}_2$$

Wäre L kontext-frei, so gäbe es Pumping-Konstante n.
Wähle $w = a^n b^n c^n$ als genügend langes Wort.
Diskutiere Zerlegungen $w = uvxyz$.
$\ell(vxy) \leq n \rightsquigarrow vxy$ enthält höchstens a's gefolgt von b's oder b's gefolgt von c's.
Beide Fälle sind analog; diskutiere also den ersten.
"Nullpumpen" liefert ein Wort, in dem alle a- und b-Vorkommen zusammen weniger als die doppelte Anzahl von c's ausmachen: $\lightning$ Wortstruktur.

Schlusswort zur Chomsky-Hierarchie für diese Vorlesung:

Dank der Pumping-Lemmata für Typ-3 bzw. Typ-2 konnten wir Trennsprachen für $\mathcal{L}_3 \subsetneq \mathcal{L}_2$ und für $\mathcal{L}_2 \subsetneq \mathcal{L}_1$ angeben.

Später sehen wir die Echtheit der Inklusion $\mathcal{L}_1 \subset \mathcal{L}_0$.

Zusammenfassend folgt dann:

$$\mathcal{L}_3 \subsetneq \mathcal{L}_2 \subsetneq \mathcal{L}_1 \subsetneq \mathcal{L}_0$$