Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Übungsblatt 7

- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.

Vorbereitung (vor der Übung selbständig zu bearbeiten)

Vorbereitungsaufgabe Ü7.1. (*Wichtige Begriffe*)

Überprüfen Sie, dass Sie die folgenden Begriffe, Notationen und Eigenschaften korrekt wiedergeben können.

- erzeugende, erreichbare, nützliche Nichtterminale
- Gängige Abschlusseigenschaften für kontextfreie Grammatiken
- CYK-Algorithmus

Vorbereitungsaufgabe Ü7.2. (*Automata Tutor: "CYK"*)

Lösen Sie die Aufgaben Ü7.2 (a–b) auf Automata Tutor.

Lösungsskizze.

(a)

1,5 S				
1,4 S	2,5 S			
1,3	2,4 S	3,5		
1,2	2,3 L	3,4	4,5	
1,1 S	2,2 A	3,3 S	4,4 B	5,5 S
x	a	x	b	x

(b)

1,4 O			
1,3 O	2,4 Y		
1,2 O, U	2,3 Y	3,4 S	
1,1 O, K, U	2,2 O, K, U	3,3 S	4,4 S
d	d	v	v

Übung und Nachbereitung

Übungsaufgabe Ü7.3. (*Abschlusseigenschaften*)

Gegeben seien die kontextfreien Grammatiken $G_1 = (V_1, \Sigma_1, P_1, S_1)$ und $G_2 = (V_2, \Sigma_2, P_2, S_2)$. Konstruieren Sie aus diesen neue Grammatiken für die Sprachen:

(a) $L(G_1) \cup L(G_2)$

(b) $L(G_1)L(G_2)$

(c) $L^*(G_1)$

Lösungsskizze. Wir nehmen oBdA an, dass $V_1 \cap V_2 = \emptyset$, weil wir im Zweifel die Nichtterminale umbenenen, also z.B. mit 1 und 2 indexieren können. Sei $S \notin V_1 \cup V_2$ ein neues Nichtterminal.

(a) $G_\cup = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 \mid S_2\}, S)$

(b) $G_{\text{konk}} = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S)$

(c) $G_{\text{stern}} = (V_1 \cup \{S\}, \Sigma_1, P_1 \cup \{S \to S_1 S \mid \varepsilon\}, S)$

Übungsaufgabe Ü7.4. (*Prächomsky-Normalform*)

Die CFG G bestehe aus folgenden Produktionen über dem Alphabet $\Sigma := \{a, b\}$:

$$
\begin{array}{lcl}
S & \to & ASA \mid aB \\
A & \to & B \mid S \mid CB \\
B & \to & b \mid \varepsilon \\
C & \to & aC \\
D & \to & aSCb \mid a
\end{array}
$$

(a) Beschreiben Sie in eigenen Worten, wann ein Nichtterminal *nützlich* in einer Grammatik ist.

(b) Reduzieren Sie die Grammatik G auf die nützlichen Nichtterminale indem Sie zunächst alle nicht erzeugenden und anschließend alle unerreichbaren Nichtterminale eliminieren.

Hinweis: Das Ergebnis ist die Grammatik aus Aufgabe Ü6.5.

Lösungsskizze.

(a) Ein Nichtterminal ist nützlich, wenn es eine Ableitung vom Startsymbol zu einem Wort (also einer Folge von Terminalen) gibt, in der das Nichtterminal verwendet wird. Nützliche Nichtterminale sind stets erzeugend und erreichbar. Erzeugend bedeutet, dass von dem Nichtterminal ausgehend ein Wort produziert werden kann; erreichbar bedeutet, dass vom Startsymbol ausgehend das Nichtterminal produziert werden kann.

Der Begriff ist relevant, da ähnlich wie bei Automaten angenommen wird, dass alle Zustände erreichbar sind, man bei Grammatiken annehmen möchte, dass sie nur nützliche Nichtterminale enthalten. Deswegen benötigt man eine klare Definition und eine Möglichkeit, eine Grammatik, die die Anforderung nicht erfüllt, in eine mit nur nützlichen Nichtterminalen zu überführen.

(b) *Erzeugende Nichtterminale*:

- Intuition: C kann kein Wort produzieren, da jede Ableitung von C beginnend stets mindestens ein Nichtterminal enthält.
- Formal bauen wir iterativ die Menge aller Nichtterminale, die eine Möglichkeit haben, ein Wort aus nur Terminalen zu erzeugen:

(1) $P_0 := \{X \mid \exists(X, \gamma) \in P.\ \gamma \in \Sigma^*\}$

(2) Wiederhole $P_{k+1} := P_k \cup \{X \in V \mid \exists(X, \gamma) \in P.\ \forall Y \in V.\ (|\gamma|_Y > 0 \to Y \in P_k)\}$ solange bis $P_{k+1} = P_k$.

- Führt auf:

$$P_0 = \{B, D\} \qquad P_1 = \{B, D, A, S\} = P_2$$

Damit kann C samt $C \to aC$, $A \to CB$ und $D \to aSCb$ entfernt werden:

$$\begin{array}{lcl} S & \to & ASA \mid aB \\ A & \to & B \mid S \\ B & \to & b \mid \epsilon \\ D & \to & a \end{array}$$

Erreichbare Nichtterminale:

- Intuition: D kann von S aus nicht erreicht werden.
- Formal bauen wir die Menge aller Nichtterminale, die von S aus erreicht werden können:

(1) $R_0 := \{S\}$

(2) Wiederhole $R_{k+1} := R_k \cup \{Y \in V \mid \exists(X, \gamma) \in P.\ X \in R_k \land |\gamma|_Y > 0\}$ solange bis $R_{k+1} = R_k$.

- Führt auf:

$$R_0 = \{S\} \qquad R_1 = \{S, A, B\} = R_2$$

Damit kann D und $D \to a$ entfernt werden.

Somit erhalten wir

$$S \to ASA \mid aB \qquad A \to B \mid S \qquad B \to b \mid \varepsilon$$

Übungsaufgabe Ü7.5. (*CYK*)

(a) Beschreiben Sie in eigenen Worten, wie die Indizes in einer CYK-Tabelle zu verstehen sind.

(b) Beschreiben Sie in eigenen Worten, wie man den Inhalt eines Feldes in der CYK-Tabelle berechnet.

(c) Wir betrachten die Grammatik $G = (\{S, T, U, A, B, C\}, \{a, b, c\}, P, S)$ in CNF mit den folgenden Produktionen P:

$$\begin{array}{ll} S \to TS \mid CT \mid a & A \to a \\ T \to AU \mid TT \mid c & B \to b \\ U \to SB \mid AB & C \to c \end{array}$$

Bestimmen Sie mit dem CYK-Algorithmus, ob $ccaab \in L(G)$ und $aabcc \in L(G)$. Geben Sie dabei auch die berechneten Tabellen an.

(d) Erweitern Sie den CYK-Algorithmus, sodass er zusätzlich alle möglichen Syntaxbäume für ein Wort w erzeugt.

(e) Verwenden Sie den erweiterten CYK-Algorithmus, um alle Syntaxbäume des Wortes $caab$ bezüglich G zu bestimmen.

Lösungsskizze.

(a) Das Feld $F_{x,y}$ beinhaltet alle Nichtterminale, die das Teilwort vom x-ten Terminal bis zum y-ten Terminal erzeugen können. (Wir fangen hier mit 1 an zu zählen.) Beispiel: Wenn das Wort $w = abaaab$ ist, dann beinhaltet $F_{2,4}$ die Nichtterminale, die baa erzeugen.

(b) Für Felder $F_{x,x}$ (in der untersten Zeile) sucht man nach allen Nichtterminalen X, die eine Produktion $X \to a$ haben wobei a das x-te Zeichen des betrachteten Wortes ist.
Für Felder $F_{x,y}$ betrachtet man alle Kombinationen aus Feldern $F_{x,x} \times F_{x+1,y}$, $F_{x,x+1} \times F_{x+2,y}$ und so weiter.[1] Wenn (A, B) in einer der Mengen vorkommt und es eine Produktion $X \to AB$ gibt, dann fügt man X zum Feld $F_{x,y}$ hinzu.

(c) Nach dem CYK-Algorithmus ergeben sich folgende Berechnungstabellen:

1,5 S,T				
1,4	2,5 S,T			
1,3 S	2,4	3,5 T		
1,2 S,T	2,3 S	3,4	4,5 U	
1,1 C,T	2,2 C,T	3,3 S,A	4,4 S,A	5,5 B
c	c	a	a	b

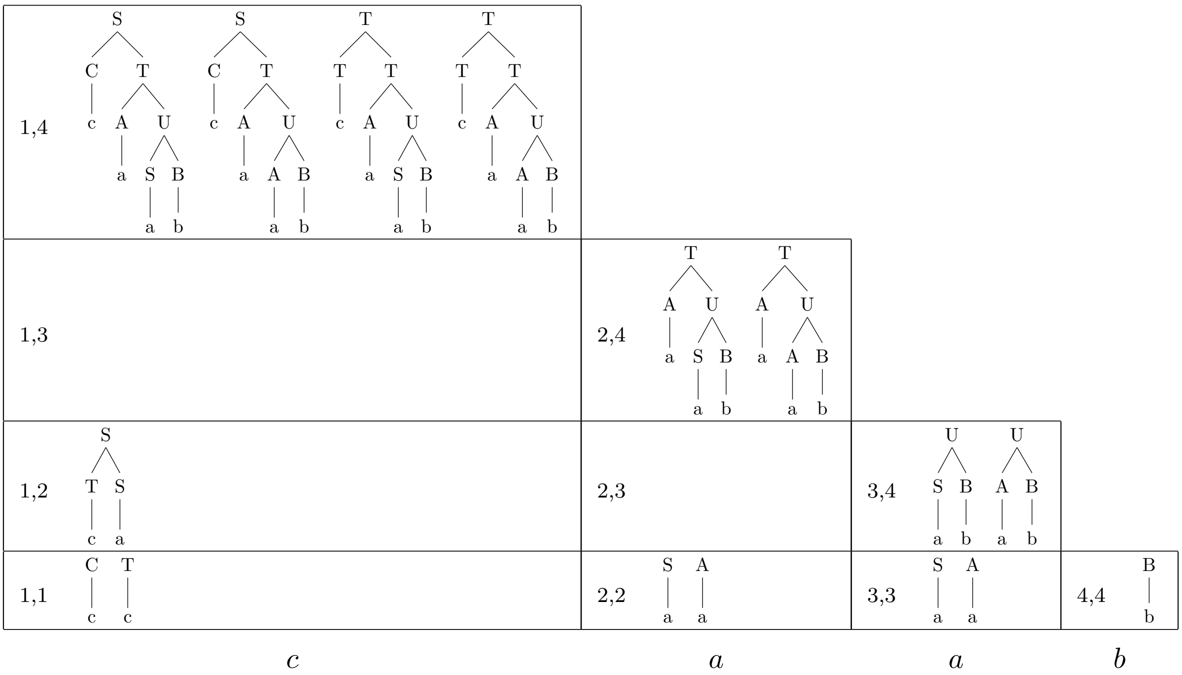
1,5 S,T				
1,4 T	2,5			
1,3 T	2,4	3,5		
1,2	2,3 U	3,4	4,5 S,T	
1,1 S,A	2,2 S,A	3,3 B	4,4 C,T	5,5 C,T
a	a	b	c	c

Also ist $ccaab \in L(G)$ und $aabcc \in L(G)$.

(d) Für $w = w_1 \cdots w_n$ sei $w_{i,j} := w_i w_{i+1} \cdots w_j$. Für die Zellen $V_{i,i}$ gibt es die Syntaxbäume $X \to w_{i,i}$. Die Syntaxbäume mit Wurzel X für $w_{i,j}$ können wir dann rekursiv zusammensetzen: Prüfe dafür für alle $i \leq k < j$ und $Y \in V_{i,k}$ und $Z \in V_{k+1,j}$, ob die Regel $X \to YZ$ existiert. Falls ja, müssen die bereits berechneten Bäume für $Y \in V_{i,k}$ und $Z \in V_{k+1,j}$ zusammengesetzt werden. Bei y Bäumen für Y und z Bäumen für Z entstehen $y \cdot z$ mögliche Bäume für X. Diese bestehen aus der Wurzel X an die links jeweils ein Baum für Y und rechts ein Baum für Z angehängt wird. Die Syntaxbäume für w sind dann am Ende die berechneten Bäume für $S \in V_{1,n}$.

(e)

[1]Tipp: Man legt seinen linken Zeigefinger auf das Feld $F_{x,x}$ und seinen rechten Zeigefinger auf $F_{x+1,y}$. Der linke Finger geht immer ein Feld höher, der rechte immer ein Feld nach unten rechts.


1,4
1,3
2,4
1,2
2,3
3,4
1,1
2,2
3,3
4,4
c
a
a
b