Prof. J. Esparza
S. Sickert, J. Krämer

Technische Universität München
KEINE ABGABE

Einführung in die theoretische Informatik

Sommersemester 2017 – Übungsblatt Lösungsskizze 8

Übungsblatt

Wir unterscheiden zwischen Übungs- und Abgabeblättern. Auf diesem *Übungsblatt* finden Sie eine Übersicht über die Kernaspekte, die Sie in Kalenderwoche 25 in den Tutorien diskutieren, üben und vertiefen. Die Aufgaben auf diesem Blatt dienen dem Üben und Verstehen des Vorlesungsstoffes, sowie dem *eigenständigen Erarbeiten* der Kernaspekte. Außerdem sollen Ihnen diese Aufgaben auch helfen, ein Gefühl dafür zu bekommen, was Sie inhaltlich in der Klausur erwartet. Klausuraufgaben können jedoch deutlich von den hier gestellten Aufgaben abweichen. Abschreiben und Auswendiglernen von Lösungen wird Ihnen daher keinen dauerhaften Erfolg in der Vorlesung bringen. Fragen zu den Übungsblättern können Sie montags bis donnerstags von 12 Uhr bis 14 Uhr in der *THEO-Sprechstunde* in Raum 03.11.034 stellen.

Kernaspekte

K8.1 korrektes Wiedergeben der folgenden Definitionen und Algorithmen

- PDA
- CYK-Algorithmus
- unterstes Kellerzeichen Z_0
- Stackalphabet Γ.

K8.2 zu einer gegebenen Sprache L einen PDA A angeben, so dass $\mathsf{L} = \mathsf{L}(\mathsf{A})$
K8.3 einen PDA A in allen verschiedenen Darstellungsformen angeben
K8.4 mithilfe des CYK-Algorithmus entscheiden, ob ein Wort w von einer Grammatik G erzeugt wird
K8.5 mithilfe des erweiterten CYK-Algorithmus alle Ableitungsbäume für ein Wort w bezüglich einer Grammmatik G angeben
K8.6 begründet entscheiden, ob gegebene Beispiele neu eingeführte Definitionen erfüllen
K8.7 Aussagen mit neu eingeführten Definitionen beweisen oder widerlegen

Notation

Erinnerung: Wir bezeichnen mit $\mathsf{L}_\varepsilon(A)$ die Sprache, die von einem PDA A mit leerem Stack akzeptiert wird. Weiterhin bezeichnen wir mit $\mathsf{L}_\mathsf{F}(\mathsf{A})$ die Sprache, die von einem PDA A mit Endzuständen F akzeptiert wird.

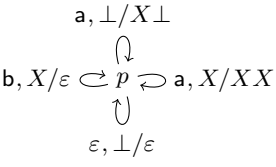
Notation von PDA-Regeln: Anstatt der in den Folien verwendeten Schreibweise $(q, YZ) \in \delta(p, \mathsf{a}, X)$ für die Ersetzungsregeln eines PDA schreibt man alternativ $pX \xrightarrow{\mathsf{a}} qYZ$ $(p, q \in \mathsf{Q},\ X, Y, Z \in \Gamma,\ \mathsf{a} \in \Sigma)$ oder stellt diese entsprechend als Graph mit Knotenmenge $\mathsf{Q}\Gamma^{\leq 2}$ dar, wobei die Kante (pX, qYZ) dann mit a beschriftet ist.
Für den PDA

$$\delta(p, \mathsf{a}, \bot) = \{(p, X\bot)\} \qquad \delta(p, \mathsf{a}, X) = \{(p, XX)\} \qquad \delta(p, \mathsf{b}, X) = \{(p, \varepsilon)\} \qquad \delta(p, \varepsilon, \bot) = \{(p, \varepsilon)\}$$

schreibt man daher alternativ:

$$p\bot \xrightarrow{\mathsf{a}} pX\bot \qquad pX \xrightarrow{\mathsf{a}} pXX \qquad pX \xrightarrow{\mathsf{b}} p \qquad p\bot \xrightarrow{\varepsilon} p$$

oder stellt diesen entsprechend als Graph mit Knotenmenge Q dar, wobei die Kante (p, q) dann mit "$\mathsf{a}, X/YZ$" beschriftet ist (siehe *Hopcroft et al., Introduction to Automata Theory, Kapitel 6*):



AUFGABE 8.1. (*Pushdown-Automata / Kellerautomaten*) Stufe B/C

Geben Sie für die folgenden Sprachen jeweils einen Kellerautomaten A_i in allen oben aufgeführten Darstellungsarten an, so dass $\mathsf{L}_i = \mathsf{L}(\mathsf{A}_i)$. Der Automat soll mit **leerem Stack** akzeptieren. Geben Sie dann zusätzlich für jeden Automaten jeweils ein nicht-leeres Wort w mit akzeptierendem Lauf an.

(a) $\mathsf{L}_1 = \{\mathsf{a}^n\mathsf{b}^{3n} \mid n \geq 0\}$
(b) $\mathsf{L}_2 = \{\mathsf{a}^n\mathsf{b}^m \in \{\mathsf{a},\mathsf{b}\}^* \mid n \leq m \leq 2n\}$
(c) $\mathsf{L}_3 = \{w \in \{\mathsf{a},\mathsf{b}\}^* \mid 2 \cdot |w|_\mathsf{a} = 3 \cdot |w|_\mathsf{b}\}$

Lösungsskizze

Komponenten des PDAs angeben, die nicht die Transitionsrelation sind

(a) $qX \xrightarrow{\mathsf{a}} qBBB \quad qX \xrightarrow{\varepsilon} q\varepsilon \quad qB \xrightarrow{\mathsf{a}} qBBBB \quad qB \xrightarrow{\mathsf{b}} p\varepsilon \quad pB \xrightarrow{\mathsf{b}} p\varepsilon$
$(q, \mathsf{abbb}, X) \to (q, \mathsf{bbb}, BBB) \to (p, \mathsf{bb}, BB) \to (p, \mathsf{b}, B) \to (p, \varepsilon, \varepsilon)$

(b) Idee: Für jedes a lege nichtdeterministisch entweder ein oder zwei b auf den Stack und überprüfe dann, ob die geratene Anzahl von bs mit der gegebenen übereinstimmt.

$$qX \xrightarrow{\mathsf{a}} qB \quad qX \xrightarrow{\mathsf{a}} qBB \quad qX \xrightarrow{\varepsilon} q\varepsilon$$
$$qB \xrightarrow{\mathsf{a}} qBB \quad qB \xrightarrow{\mathsf{a}} qBBB \quad qB \xrightarrow{\mathsf{b}} p\varepsilon \quad pB \xrightarrow{\mathsf{b}} p\varepsilon$$

$(q, \mathsf{aabbb}, X) \to (q, \mathsf{abbb}, BB) \to (q, \mathsf{bbb}, BBB) \to (p, \mathsf{bb}, BB) \to (p, \mathsf{b}, B) \to (p, \varepsilon, \varepsilon)$

(c) Idee: Verwende Stack als (unären) Zähler und benutze explizites Bottom-Symbol, um auf 0 zu testen. Für jedes a zähle um 2 (codiert als XX) hoch und für jedes b ziehe 3 (codiert als YYY) ab.

$$q\bot \xrightarrow{\mathsf{a}} qXX\bot \qquad qX \xrightarrow{\mathsf{a}} qXXX \qquad qY \xrightarrow{\mathsf{a}} p^+\varepsilon$$
$$q\bot \xrightarrow{\mathsf{b}} qYYY\bot \qquad qY \xrightarrow{\mathsf{b}} qYYYY \qquad qX \xrightarrow{\mathsf{b}} p^{--}\varepsilon$$
$$p^+\bot \xrightarrow{\varepsilon} qX\bot \qquad p^+Y \xrightarrow{\varepsilon} q\varepsilon$$
$$p^{--}\bot \xrightarrow{\varepsilon} qYY\bot \qquad p^{--}X \xrightarrow{\varepsilon} p^-\varepsilon$$
$$p^-\bot \xrightarrow{\varepsilon} qY\bot \qquad p^-X \xrightarrow{\varepsilon} q\varepsilon$$
$$q\bot \xrightarrow{\varepsilon} q\varepsilon$$

$(q, \mathsf{abbaa}, \bot) \to (q, \mathsf{bbaa}, XX\bot) \to (p^{--}, \mathsf{baa}, X\bot) \to (p^-, \mathsf{baa}, \bot) \to (q, \mathsf{baa}, Y\bot) \to (q, \mathsf{aa}, YYYY\bot) \to (p^+, \mathsf{a}, YYY\bot) \to (q, \mathsf{a}, YY\bot) \to (p^+, \varepsilon, Y\bot) \to (q, \varepsilon, \bot) \to (q, \varepsilon, \varepsilon)$

AUFGABE 8.2. (*Komplemente von CFL*) Stufe C

Aus der Vorlesung wissen Sie bereits, dass kontextfreie Sprachen *nicht* über Schnitt und Komplement abgeschlossen sind. In dieser Aufgabe geht es darum zu sehen, dass es dennoch kontextfreie, nicht-reguläre Sprachen gibt, deren Komplement ebenfalls kontextfrei und nicht regulär ist. Sei $\mathsf{L} = \{\mathsf{a}^n\mathsf{b}^n\mathsf{c}^m \mid n, m \in \mathbb{N}_0\}$ über dem Alphabet $\Sigma = \{\mathsf{a},\mathsf{b},\mathsf{c}\}$ eine nicht-reguläre, kontextfreie Sprache.

(a) Geben Sie eine kontextfreie Grammatik G für L und eine für $\overline{\mathsf{L}}$ an.
(b) Geben Sie einen PDA für L und für $\overline{\mathsf{L}}$ an.

Lösungsskizze

(a) Die Grammatik G für L:

$$\mathsf{S} \to XY \qquad X \to aXb \mid \varepsilon \qquad Y \to cY \mid \varepsilon$$

Die Grammatik G für $\overline{\mathsf{L}}$:

$$\mathsf{S} \to X \mid aUC \mid VbC \qquad U \to aUb \mid aU \mid \varepsilon \qquad V \to aVb \mid Vb \mid \varepsilon \qquad C \to cC \mid \varepsilon$$
$$X \to aX \mid bY \mid cZ \qquad Y \to aT \mid bY \mid cZ \qquad Z \to aT \mid bT \mid cZ \qquad T \to aT \mid bT \mid cT \mid \varepsilon$$

(b) Gleiche Konstruktionsidee wie in Aufgabe 8.1.

AUFGABE 8.3. (*CYK-Algorithmus*) Stufe C

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

$$\mathsf{S} \to T\mathsf{S} \mid CT \mid \mathsf{a} \qquad A \to \mathsf{a}$$
$$T \to AU \mid TT \mid \mathsf{c} \qquad B \to \mathsf{b}$$
$$U \to \mathsf{S}B \mid AB \qquad C \to \mathsf{c}$$

(a) Bestimmen Sie mit dem CYK-Algorithmus, ob $\mathsf{ccaab} \in \mathsf{L}(\mathsf{G})$ und $\mathsf{aabcc} \in \mathsf{L}(\mathsf{G})$. Geben Sie dabei auch die berechneten Tabellen an.
(b) Beschreiben Sie eine Erweiterung des CYK-Algorithmus, mit welcher für ein gegebenes $w \in \mathsf{L}(\mathsf{G})$ alle Ableitungsbäume bzgl. G berechnet werden können, und wenden Sie dieses Verfahren auf die Wörter aus (a)

an.

Lösungsskizze

(a) Nach dem CYK-Algorithmus ergeben sich folgende Berechnungstabelle:

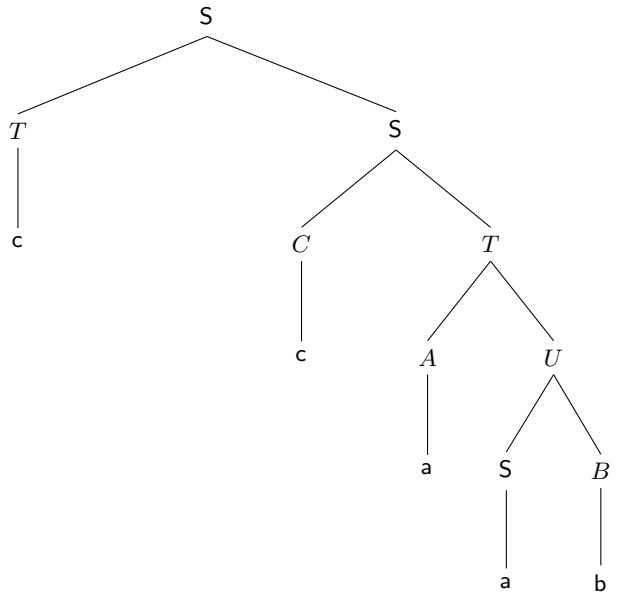
$_{15}$ S, T				
$_{14}$	$_{25}$ S, T			
$_{13}$ S	$_{24}$	$_{35}$ T		
$_{12}$ S, T	$_{23}$ S	$_{34}$	$_{45}$ U	
$_{11}$ C, T	$_{22}$ C, T	$_{33}$ S, A	$_{44}$ S, A	$_{55}$ B
c	c	a	a	b

$_{15}$ S, T				
$_{14}$ T	$_{25}$			
$_{13}$ T	$_{24}$	$_{35}$		
$_{12}$	$_{23}$ U	$_{34}$	$_{45}$ S, T	
$_{11}$ S, A	$_{22}$ S, A	$_{33}$ B	$_{44}$ C, T	$_{55}$ C, T
a	a	b	c	c

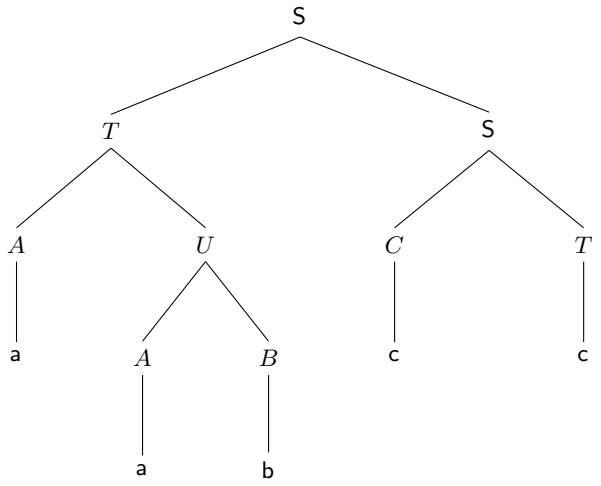
Also ist $ccaab \in \mathsf{L}(\mathsf{G})$ und $aabcc \in \mathsf{L}(\mathsf{G})$.

(b) Bei der Berechnung von V_{ij} annotiert man die Elemente $X \in V_{ij}$ mit den verwendeten Regeln und dem Index. Im Basisfall $X \in V_{ii}$ werden alle Variablen der Produktion $X \to w_i$ annotiert. Sei $X \to YZ$ und sei $Y \in V_{ik}$ und $Z \in V_{(k+1)j}$. Dann wird $X \in V_{ij}$ mit $(X \to YZ, k)$ annotiert.

$_{15}$ $(T \to TT, 1)$, $(T \to TT, 2)$, $(\mathsf{S} \to T\mathsf{S}, 1)$, $(\mathsf{S} \to CT, 1)$				
$_{14}$	$_{25}$ $(T \to TT, 2)$, $(\mathsf{S} \to CT, 2)$			
$_{13}$ $(\mathsf{S} \to T\mathsf{S}, 2)$, $(\mathsf{S} \to T\mathsf{S}, 1)$	$_{24}$	$_{35}$ $(T \to AU, 3)$		
$_{12}$ $(T \to TT, 1)$, $(\mathsf{S} \to CT, 1)$	$_{23}$ $(\mathsf{S} \to T\mathsf{S}, 2)$	$_{34}$	$_{45}$ $(U \to SB, 4)$, $(U \to AB, 4)$	
$_{11}$ $(C \to c)$, $(T \to c)$	$_{22}$ $(C \to c)$, $(T \to c)$	$_{33}$ $(\mathsf{S} \to a)$, $(A \to a)$	$_{44}$ $(\mathsf{S} \to a)$, $(A \to a)$	$_{55}$ $(B \to b)$
c	c	a	a	b



$_{15}$ $(T \to TT, 4)$, $(T \to TT, 3)$, $(\mathsf{S} \to T\mathsf{S}, 3)$				
$_{14}$ $(T \to TT, 3)$	$_{25}$			
$_{13}$ $(T \to AU, 1)$	$_{24}$	$_{35}$		
$_{12}$	$_{23}$ $(U \to AB, 2)$, $(U \to SB, 2)$	$_{34}$	$_{45}$ $(T \to TT, 4)$, $(\mathsf{S} \to CT, 4)$	
$_{11}$ $(\mathsf{S} \to a)$, $(A \to a)$	$_{22}$ $(\mathsf{S} \to a)$, $(A \to a)$	$_{33}$ $(B \to b)$	$_{44}$ $(C \to c)$, $(T \to c)$	$_{55}$ $(C \to c)$, $(T \to c)$
a	a	b	c	c



AUFGABE 8.4. (*Abschlusseigenschaften kontextfreier Sprachen*) Stufe D

Beweisen Sie die folgenden beiden Aussagen:

(a) Der Präfixabschluss $\mathsf{L}_{pre} \coloneqq \{u \mid \exists v.\ uv \in \mathsf{L}\}$ einer kontextfreien Sprache $\mathsf{L} \subseteq \Sigma^*$ über einem Alphabet Σ ist wieder kontextfrei. Geben Sie hierzu ein Übersetzung an, die die Grammatik G für L in eine Grammatik G' umwandelt, sodass $\mathsf{L}_{pre} = \mathsf{L}(G')$ übersetzt.

(b) Kontextfreie Sprachen sind unter Schnitt mit regulären Sprachen abgeschlossen.

Lösungsskizze

(a) Sei $\mathsf{G} = (V, \Sigma, \mathsf{P}, \mathsf{S})$ CFG in CNF mit $\mathsf{L} = \mathsf{L}(\mathsf{G})$ und $\mathsf{L}' := \{u \in \Sigma^* \mid \exists v \in \Sigma^*.\ uv \in \mathsf{L}(\mathsf{G})\}$.
Idee:
Es gilt: $u \in \mathsf{L}'$ gdw. es gibt einen Ableitungsbaum zu uv bzgl. G für ein $v \in \Sigma^*$.
Betrachte in diesem Ableitungsbaum zu uv den eindeutigen Pfad vom letzten Zeichen von u zur Wurzel. Wir erweitern G so zu G', dass G' diesen Pfad raten kann und die Teilbäume, die Zeichen von v erzeugen, weglassen kann.
Konstruiere $\mathsf{G}' = (V \uplus V' \uplus \{\mathsf{S}''\}, \Sigma, P \uplus \mathsf{P}', \mathsf{S}'')$ hierzu wie folgt:
- $V' = \{X' \mid X \in V\}$ – die Markierung rät den Pfad von der Wurzel zum letzten Zeichen von u und steht immer am rechtesten NT einer Ableitung bzgl. G'
- $\mathsf{P}' = \{X' \to YZ', X' \to Y' \mid X \to YZ\} \cup \{X' \to a \mid X \to a \in \mathsf{P}\} \cup \{\mathsf{S}'' \to \mathsf{S}' \mid \varepsilon\}$ – P' erlaubt G' zu raten, ob das letzte Zeichen von u durch Y oder Z erzeugt wird. Falls es durch Y erzeugt wird, wird der durch Z erzeugte Suffix fallengelassen (Fall $X' \to Y'$).

Behauptung:
$\mathsf{L}(\mathsf{G}') = \mathsf{L}' := \{u \in \Sigma^* \mid \exists v \in \Sigma^*.\ uv \in \mathsf{L}(\mathsf{G})\}$.
Beweis:
Sei $u \in \mathsf{L}'$. Betrachte entsprechend der obigen Beschreibung einen Ableitungsbaum zu $uv \in \mathsf{L}(\mathsf{G})$ bzgl. G. Markiere die Knoten entlang des Pfades vom letzten Zeichen von u bis zur Wurzel. Schneide die Teilbäume zu v ab. Nach Konstruktion erhält man einen Ableitungsbaum zu u bzgl. G'.
Sei $u \in \mathsf{L}(\mathsf{G}')$. Betrachte Ableitungsbaum zu u bzgl. G'. Für jeden inneren Knoten, der (1) kein Terminal erzeugt, (2) einem markierten NT X' entspricht und (3) nur ein Kind hat, das nach Definition von P' damit auch einem markierten NT Y' entsprechen muss, gibt es eine Regel $X \to YZ \in \mathsf{P}$ bzgl. der originalen Grammatik G. OBdA. ist G auf die nützlichen Symbole reduziert, womit es mindestens einen Ableitungsbaum zu Z gibt. Füge diesen als rechten Teilbaum bei dem inneren Knoten an. Ersetze dann X' durch X im Baum, um einen Ableitungsbaum bzgl. G zu erhalten, der ein Wort der Form uv erzeugt.

(b) Sei L kontextfrei und L' regulär. Ohne Einschränkung gibt es $\mathsf{G} = (V, \Sigma, \mathsf{P}, \mathsf{S})$ CFG in CNF und $\mathsf{A} = (\mathsf{Q}, \Sigma, \delta, q_I, \mathsf{F})$ DFA mit $\mathsf{L} = \mathsf{L}(\mathsf{G})$ und $\mathsf{L}' = \mathsf{L}(\mathsf{A})$.
Idee: Wir erweitern G so, dass eine akzeptierende Berechnung von A geraten wird.
Definiere $\mathsf{G}' = (V', \Sigma, \mathsf{P}', \mathsf{S}')$ wie folgt:
- $V' = \mathsf{Q} \times V \times \mathsf{Q}$
- $\mathsf{P}' = \mathsf{P}'_{X \to YZ} \uplus \mathsf{P}'_{X \to \mathsf{a}} \uplus \mathsf{P}'_{\mathsf{S}'}$ mit

$$\begin{aligned} \mathsf{P}'_{X \to YZ} &:= \{(q, X, q') \to (q, Y, q'')(q'', Z, q') \mid X \to YZ \in P\} \\ \mathsf{P}'_{X \to \mathsf{a}} &:= \{(q, X, q') \to a \mid X \to a \in P \wedge \delta(q, a) = q'\} \\ \mathsf{P}'_{\mathsf{S}'} &:= \{\mathsf{S}' \to (q_I, \mathsf{S}, q_F) \mid q_F \in F\} \cup \{\mathsf{S}' \to \varepsilon \mid \varepsilon \in \mathsf{L}(\mathsf{G}) \cap \mathsf{L}(A)\} \end{aligned}$$

Behauptung: $\mathsf{L}(\mathsf{G}') = \mathsf{L}(\mathsf{G}) \cap \mathsf{L}(A)$

Sei $w = a_1 a_2 \dots a_l \in \mathsf{L}(\mathsf{G}')$. Dann gibt es eine Ableitung der Form

$$\mathsf{S}' \to (q_I, \mathsf{S}, q_F) \to^* (q_0, X_1, q_1)(q_1, X_2, q_2) \dots (q_{l-1}, X_l, q_l)$$

mit $(q_{i-1}, X_i, q_i) \to a_i$.
Nach Def. von $\mathsf{P}_{X \to YZ}$ muss $q_0 = q_I$ und $q_l = q_F$ gelten.
Nach Def. von $\mathsf{P}_{X \to \mathsf{a}}$ muss $q_i = \delta(q_{i-1}, a)$ gelten.
Damit existiert eine akzeptierende Berechnung von A zu w, also $w \in \mathsf{L}(A)$.
Nach Definition erhält man durch Vergessen der q, q', also durch Übergang von (q, X, q') zu X aus der obigen Ableitung auch eine Ableitung in G, womit auch $w \in \mathsf{L}(\mathsf{G})$ folgt.
Insgesamt $w \in \mathsf{L}(A) \cap \mathsf{L}(\mathsf{G})$.
Sei $w \in \mathsf{L}(A) \cap \mathsf{L}(\mathsf{G})$. Dann existiert eine akzeptierende Berechnung $q_i = \delta(q_{i-1}, a_i)$ von A zu w und ein Ableitungsbaum zu w bzgl. G. Letzteren erweitert man induktiv zu einem Ableitungsbaum von G': Den inneren Knoten, an dem das Blatt zu a_i hängt, schreibt man von X auf (q_{i-1}, X, q_i) um. Bottom-up/induktiv schreibt man dann X nach (q_i, X, q_k) um, falls seine beiden Kinder (da CNF) gerade $(q_i, Y, q_j), (q_j, Z, q_k)$ sind. Alternativ: aus X wird (q_i, X, q_k), falls X das Teilwort $\mathsf{a}_{i+1} \dots \mathsf{a}_k$ erzeugt. Da G' die Zwischenzustände q'' frei mittels $\mathsf{P}_{X \to YZ}$ raten kann, handelt es sich tatsächlich um einen Ableitungsbaum von w bzgl. P', also $w \in \mathsf{L}(\mathsf{G}')$.

AUFGABE 8.5. (*CYK-Algorithmus — Teil 2*) Stufe C
Sei G durch folgende Produktionen gegeben:

$$\mathsf{S} \to AB \mid CD \mid AT \mid CU \mid \mathsf{SS} \qquad T \to \mathsf{S}B \qquad U \to \mathsf{S}D \qquad A \to (\qquad B \to) \qquad C \to \{ \qquad D \to \}$$

(a) Wenden Sie den CYK-Algorithmus auf folgende Wörter an:

$$w_1 = (){()} \qquad w_2 = (){(\qquad w_3 = ()},$$

indem Sie die Tabellen, die Sie mithilfe des CYK-Algorithmus berechnen, angeben.

(b) Diskutieren Sie, wie man beim CYK-Algorithmus das mehrmalige Berechnen der gleichen Tabelle verhindern kann. Bestimmen Sie dazu, in welcher Beziehung die Worte w_1 bis w_3 zueinander stehen und warum es in Aufgabenteil (a) reicht, eine einzige Tabelle zu berechnen.

(c) Wir möchten den CYK-Algorithmus so abändern, dass er Eingabefehler (z.B. eine öffnende Klammer ohne passende schließende Klammer) erkennt und die Position des Fehlers bestimmt. *Geben Sie hierzu eine neue Variante des CYK-Algorithmus so an, dass er für gegebenes $w \in \Sigma^*$ den längsten Präfix u bestimmt, so dass sich u zu einem Wort in $\mathsf{L}(\mathsf{G})$ ergänzen lässt.*
Beispiel:

$$\mathsf{G}\colon \mathsf{S} \to AB \mid \mathsf{SS} \qquad A \to \mathsf{a} \qquad B \to \mathsf{b}$$

Dann gilt $w = \mathsf{ababb} \notin \mathsf{L}(\mathsf{G})$, aber $\mathsf{ab}, \mathsf{abab} \in \mathsf{L}(\mathsf{G})$; weiterhin gibt es kein $v \in \Sigma^*$, so dass $wv \in \mathsf{L}(\mathsf{G})$ gilt, womit abab das gesuchte Präfix ist. Im Fall $w = \mathsf{aba}$ wäre aba das gesuchte Präfix, da $w\mathsf{b} \in \mathsf{L}(\mathsf{G})$ gilt.

(d) Wenden Sie den Algorithmus aus (c) auf die Wörter aus (a) an, um das maximalen Präfix zu bestimmen, der sich noch zu einem Wort in $\mathsf{L}(\mathsf{G})$ ergänzen lässt.

Lösungsskizze

(a) Berechnen der CYK-Tabelle:

$_{16}$ S					
$_{15}$	$_{26}$				
$_{14}$	$_{25}$	$_{36}$ S			
$_{13}$	$_{24}$	$_{35}$	$_{46}$ U		
$_{12}$ S	$_{23}$	$_{34}$	$_{45}$ S	$_{56}$	
$_{11}$ A	$_{22}$ B	$_{33}$ C	$_{44}$ A	$_{55}$ B	$_{66}$ D
(	)	{	(	)	}

Ablesen: $w_1 \in \mathsf{L}(\mathsf{G})$, da $\mathsf{S} \in V_{16}$. $w_2 \notin \mathsf{L}(\mathsf{G})$, da $\mathsf{S} \notin V_{14}$. $w_3 \notin \mathsf{L}(\mathsf{G})$, da $\mathsf{S} \notin V_{46}$.

(b) Da V_{ij} nur von $\mathsf{a}_i \dots \mathsf{a}_j$ abhängig ist, können Lösungen von Teilwörtern einfach aus der Tabelle abgelesen werden.

(c) Gegeben G konstruiere G' mit $\mathsf{L}(\mathsf{G}') = \{u \mid uv \in \mathsf{L}(\mathsf{G})\}$ entsprechend Aufgabe 8.4 zzgl. Zusammenziehen von Kettenproduktionen $X' \to Y'$. Sei $w = \mathsf{a}_1 \dots \mathsf{a}_l$ gegeben und $u = \mathsf{a}_1 \dots \mathsf{a}_k$ ein Präfix von w. u lässt sich zu einem Wort aus $\mathsf{L}(\mathsf{G})$ ergänzen gdw. $u \in \mathsf{L}(\mathsf{G}')$ gdw. $\mathsf{S}' \in V_{1,k}$ wenn man den CYK bzgl. G' auf w anwendet. Suche daher nach dem maximalen k, so dass $\mathsf{S}' \in V_{1,k}$ gilt, um den maximalen Präfix von w zu bestimmen, der sich zu einem Wort auf $\mathsf{L}(\mathsf{G})$ ergänzt lässt.

(d) Berechnete Grammatik für Präfixsprache ohne Start-ε-Produktion:

$$
\begin{array}{lcl}
\mathsf{S}' & \to & \mathsf{S}\mathsf{S}' \mid AT' \mid CU' \mid AB \mid CD \mid (\mid \{ \\
\mathsf{S} & \to & AB \mid CD \mid AT \mid CU \mid \mathsf{S}\mathsf{S} \\
T' & \to & \mathsf{S}\mathsf{S}' \mid AT' \mid CU' \mid AB \mid CD \mid \mathsf{S}B \mid (\mid \{ \\
T & \to & \mathsf{S}B \\
U' & \to & \mathsf{S}D \mid \mathsf{S}\mathsf{S}' \mid AT' \mid CD \mid CU' \mid AB \mid (\mid \{ \\
U & \to & \mathsf{S}D \\
A & \to & (\qquad B \to) \qquad C \to \{ \qquad D \to \}
\end{array}
$$

Berechnen der CYK-Tabelle:

$_{16}$ S', S, T', U'										
$_{15}$ S', T', U'	$_{26}$									
$_{14}$ S', T', U'	$_{25}$		$_{36}$ S', S, T', U'							
$_{13}$ S', T', U'	$_{24}$		$_{35}$ S', T', U'		$_{46}$ U, U'					
$_{12}$ S', S, T', U'	$_{23}$		$_{34}$ S', T', U'		$_{45}$ S', S, T', U'		$_{56}$			
$_{11}$ S',T',U',A	$_{22}$ B		$_{33}$ S', T', U', C		$_{44}$ S', T', U', A		$_{55}$ B		$_{66}$ D	
(		)	{		(		)		}	

Längste Präfixe:

- $w_1 \in \mathsf{L}(\mathsf{G})$
- $w_2 \in \mathsf{L}(\mathsf{G}')$ und $w_2)\} \in \mathsf{L}(\mathsf{G})$
- $w_3 \notin \mathsf{L}(\mathsf{G}')$, aber $() \in \mathsf{L}(\mathsf{G}')$ und $() \in \mathsf{L}(\mathsf{G})$.

AUFGABE 8.6. (*Ogdens Lemma*) Stufe E

Sei L eine CFL. Dann existiert ein $p \in \mathbb{N}_0$, so dass für jedes $z \in \mathsf{L}$ mit $|z| \geq p$ und jede *Markierung* von mindestens p Zeichen in z gilt: Es gibt eine Zerlegung $z = uvwxy$ mit

- vx enthält mindestens ein markiertes Zeichen.
- vwx enthält höchstens p markierte Zeichen.
- $uv^iwx^iy \in \mathsf{L}$ für jedes $i \in \mathbb{N}_0$.

Eine geeignete Markierung kann z.B. Unterstreichung $\underline{\mathsf{a}}$, Apostroph a' oder Einfärbung a sein. Markiert man stets alle Zeichen, so erhält man das ursprüngliche Pumping-Lemma für CFL.

(a) Sei $\mathsf{L} = \{a^ib^jc^kd^l \mid i = 0 \vee j = k = l\}$.
Zeigen Sie mittels Ogdens Lemma, dass L nicht kontextfrei ist.
(Warum reicht das Pumping-Lemma für CFL nicht aus, um zu zeigen, dass L nicht kontextfrei ist?)

(b) Beweisen Sie Ogdens Lemma.

Lösungsskizze

(a) Sei L kontextfrei. Dann gilt Ogdens Lemma für L. Sei p entsprechend dem Lemma für L gewählt. Wir wählen $z = ab^pc^pd^p \in \mathsf{L}$, wobei genau b^p markiert sei.
Damit sind in vwx stets höchstens p Zeichen markiert. Nach Ogdens Lemma enthält vx mindestens ein b.
Mögliche Fälle für vx:
- vx enthält kein c. Dann hat uv^2wx^2y mindestens ein a, $p+1$ bs, aber immer noch p cs.
- vx enthält kein d. Analog.
- vx enthält sowohl ein c als auch ein d zzgl. mindestens einem b. Damit müssen in v oder x mindestens zwei verschiedene Zeichen vorkommen (Schubfachprinzip), womit uv^2wx^2z nicht mehr von der Form $a^*b^*c^*d^*$ sein kann.

Mit dem PL für CFL gelingt der Nachweis, dass L nicht kontextfrei ist, allerdings nicht:
Sei L kontextfrei und n eine somit existierende Pumping-Lemma-Konstante für L. Da $z \in \mathsf{L}$ gelten muss, kann man für die Struktur von z in Abhängigkeit von der Anzahl der a zwei Fälle unterscheiden:
$|z|_a > 0$: Somit $z = a^ib^jc^jd^j$ mit $i > 0$ und $j \geq 0$. Hier erlaubt das PL nicht die Position von vwx einzuschränken, womit man nicht ausschließen kann, dass vx nur aus as besteht, womit man keinen Widerspruch herleiten kann. (Hier konnte man mittels Ogdens Lemma erzwingen, dass vx mindestens ein b enthält.)
$|z|_a = 0$: Somit $z = b^jc^kd^l$. Hier kann man nicht ausschließen, dass vwx vollständig in einem Block zu liegen kommt. Da man kein a durch das Pumpen erzeugen kann, kann man in diesem Fall wiederum keinen Widerspruch herleiten.

(b) Sei G eine CFG in CNF mit $\mathsf{L} = \mathsf{L}(\mathsf{G})$. Sei $p = 2^{|V|}$. Man betrachte einen beliebigen Ableitungsbaum für ein beliebiges Wort $z \in \mathsf{L}$, in welchem mindestens p Zeichen markiert sind.
Beginnend bei der Wurzel steigt man stets in den Teilbaum ab, in welchem die größere Anzahl an Blättern markiert ist. In jedem Schritt kann sich somit die Anzahl der erreichbaren markierten Blätter höchstens halbieren, womit man einen Pfad der Länge $\geq |V| + 2$ in dem Ableitungsbaum konstruiert, auf dem sich mindestens ein Nichtterminal wiederholen muss. Wie im normalen PL für CFL wählt man nun von unten aufsteigend das erste sich wiederholende NT, um den Pumpbaum zu definieren, welcher der zyklischen

Ableitung $A \to^* vAw$ entspricht. Insbesondere entspricht $A \to^* vwx$ einem Ableitungsbaum der Höhe $\leq |V|$ (wieder unter Vernachlässigung der Terminale), womit vwx höchstens Länge $2^{|V|} = p$ (da G in CNF/Binärbaum) haben kann. Ebenso folgt, dass $|vx| > 0$ gilt, aus der CNF (binäre Ableitungsbäume und keine ε-Regeln).