Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 6

Vorlesung Theoretische Grundlagen der Informatik im WS 18/19

Ausgabe 8. Januar 2019
Abgabe 22. Januar 2019, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

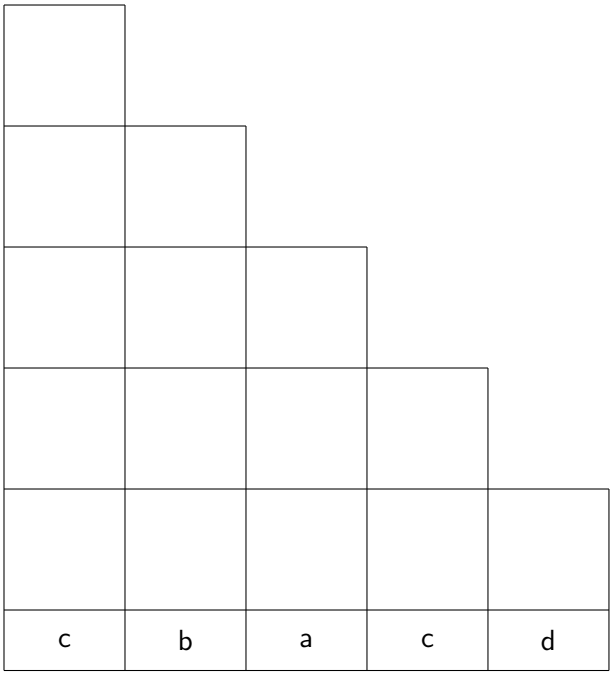
Aufgabe 1

(2 + 2 + 3 + 1 = 8 Punkte)

Gegeben sei die Grammatik $G = (\Sigma, V, S, R)$ mit Terminalen $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$, Nichtterminalen $V = \{S, A, B, C, D\}$, Startsymbol S und Produktionen

$$\begin{aligned} R = \{ & S \to AD \\ & A \to CB \mid \mathsf{a} \mid \mathsf{c} \\ & B \to AD \mid \mathsf{b} \mid \mathsf{d} \\ & C \to DB \mid \mathsf{c} \\ & D \to AC \mid \mathsf{c} \mid \mathsf{d}\} \end{aligned}$$

(a) Überprüfen Sie mit dem CYK-Algorithmus, ob das Wort cbacd in der Sprache $L(G)$ enthalten ist.



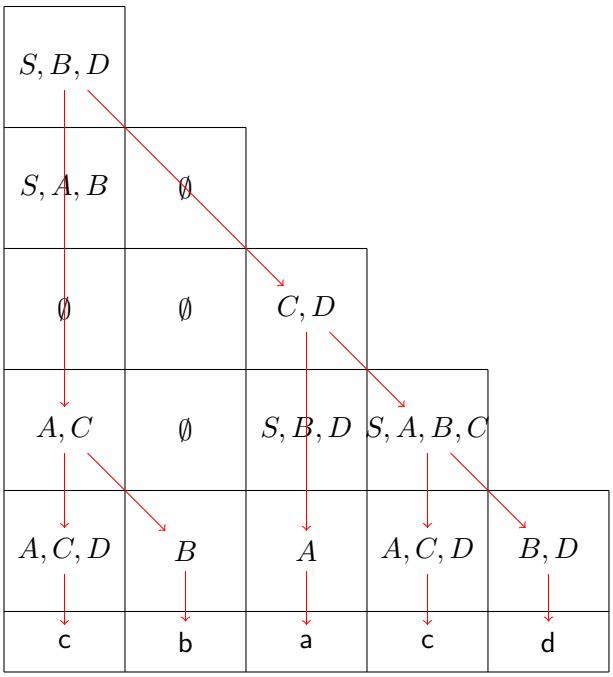
(b) Geben Sie einen Syntaxbaum für das Wort cbacd an.

(c) Erweitern Sie den CYK-Algorithmus so, dass falls das überprüfte Wort tatsächlich von der Grammatik erzeugt wird, ein Syntaxbaum ausgegeben wird.

(d) Wie viel zusätzliche Laufzeit und wie viel zusätzlichen Speicher benötigt Ihr Algorithmus aus Teilaufgabe (c)? Begründen Sie!

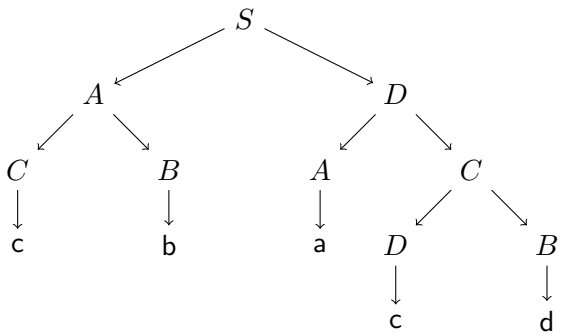
Lösung:

(a) Vergleiche folgende Tabelle:



Das Wort ist also in der Sprache enthalten.

(b) Vergleiche folgenden Syntaxbaum:



(c) Zu jedem Symbol in V_{ij} werden außerdem die ein oder zwei Vorgängersymbole gespeichert: Für $i \neq j$ sind das die zwei Symbole aus V_{ik} und V_{kj}, aus dem es berechnet wurde. Für $i = j$ ist es das eine Terminalsymbol, auf das es abgeleitet wurde. In der obigen Tabelle sind diese Vorgänger durch rote Pfeile markiert. Wird das Wort von der Grammatik erzeugt, kann der Syntaxbaum über die Vorgängersymbole rekursiv ausgegeben werden.

(d) Die Laufzeit zum Speichern der Vorgängersymbole erhöht die Laufzeit um einen konstanten Faktor. Der benötigte Speicherplatz steigt ebenfalls um einen konstanten Faktor. Die Ausgabe des Syntaxbaums eines Wortes der Länge n ist in $\mathcal{O}(n)$ Zeit möglich, da der Syntaxbaum höchstens n Knoten besitzt.

Aufgabe 2 (3 + 1 = 4 Punkte)

Sei eine Grammatik G durch $V = \{S, X, Y\}, \Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$ und folgende Regelmenge R gegeben:

$$\begin{aligned} S &\to \mathsf{a}S\mathsf{c} \mid X \mid \mathsf{d}Y \\ X &\to \mathsf{b}X\mathsf{c} \mid \varepsilon \\ Y &\to \mathsf{d}S \mid \varepsilon \end{aligned}$$

(a) Verwenden Sie das Verfahren aus der Vorlesung, um G in eine äquivalente Grammatik in erweiterter Chomsky-Normalform zu überführen. Es genügt dabei, wenn sie nach jedem Schritt bzw. jeder Phase das jeweilige Ergebnis angeben.

(b) In Schritt 4, Phase 2 wird beim Ersetzen von Kettenregeln der Form $A \to B$ in umgekehrter topologischer Sortierung vorgegangen. Warum ist dies nötig? Geben Sie eine Grammatik und eine Sortierung der Variablen an, für die das Verfahren nicht zum korrekten Ergebnis führt.

Lösung:

(a) **Schritt 1:** Ersetze gemischte Produktionen mit Terminal- und Nichtterminalsymbolen auf der rechten Seite.

$$\begin{aligned} S &\to ASC \mid X \mid DY \\ X &\to BXC \mid \varepsilon \\ Y &\to DS \mid \varepsilon \\ A &\to \mathsf{a} \\ B &\to \mathsf{b} \\ C &\to \mathsf{c} \\ D &\to \mathsf{d} \end{aligned}$$

Schritt 2: Ersetze Produktionen mit Länge ≥ 3.

$$\begin{aligned} S &\to AZ_1 \mid X \mid DY \\ X &\to BZ_2 \mid \varepsilon \\ Y &\to DS \mid \varepsilon \\ Z_1 &\to SC \\ Z_2 &\to XC \\ A &\to \mathsf{a} \\ B &\to \mathsf{b} \\ C &\to \mathsf{c} \\ D &\to \mathsf{d} \end{aligned}$$

Schritt 3: Ersetze Regeln der Form $A \to \varepsilon$.

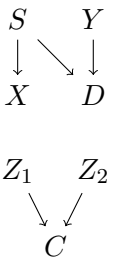
Die Menge der Variablen, die sich auf ε ableiten lassen, ist $V' = \{S, X, Y\}$. Für X und Y ist das offensichtlich. Für S lässt sich die Ableitungsfolge $S \to X \to Y$ finden. A, B, C, D sind

offensichtlich nicht in V', während Z_1 und Z_2 sind wegen des Vorkommens von C auf der rechten Seite nur in Wörter ableiten lassen, die mindestens ein c enthalten.

$$
\begin{aligned}
S &\to AZ_1 \mid X \mid DY \mid D \\
X &\to BZ_2 \\
Y &\to DS \mid D \\
Z_1 &\to SC \mid C \\
Z_2 &\to XC \mid C \\
A &\to \mathsf{a} \\
B &\to \mathsf{b} \\
C &\to \mathsf{c} \\
D &\to \mathsf{d}
\end{aligned}
$$

Schritt 4: Ersetze Kettenregeln der Form $A \to B$.

Abhängigkeitsgraph:



Zyklische Abhängigkeiten der Form $A_1 \to A_2 \to \cdots \to A_n \to A_1$ gibt es in diesem Produktionssystem nicht. Eine mögliche topologische Sortierung ist S, Y, X, D, Z_1, Z_2, C.

$$
\begin{aligned}
S &\to AZ_1 \mid BZ_2 \mid DY \mid \mathsf{d} \\
X &\to BZ_2 \\
Y &\to DS \mid \mathsf{d} \\
Z_1 &\to SC \mid \mathsf{c} \\
Z_2 &\to XC \mid \mathsf{c} \\
A &\to \mathsf{a} \\
B &\to \mathsf{b} \\
C &\to \mathsf{c} \\
D &\to \mathsf{d}
\end{aligned}
$$

Schritt 5: Stelle die Ableitung $S \xrightarrow{\star} \varepsilon$ wieder her.

Das Endergebnis ist eine Grammatik G' mit $V' = \{S', S, X, Y, Z_1, Z_2, A, B, C, D\}, \Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$ und Regelmenge R':

$$
\begin{aligned}
S' &\to S \mid \varepsilon \\
S &\to AZ_1 \mid BZ_2 \mid DY \mid \mathsf{d} \\
X &\to BZ_2 \\
Y &\to DS \mid \mathsf{d} \\
Z_1 &\to SC \mid \mathsf{c} \\
Z_2 &\to XC \mid \mathsf{c} \\
A &\to \mathsf{a} \\
B &\to \mathsf{b} \\
C &\to \mathsf{c} \\
D &\to \mathsf{d}
\end{aligned}
$$

(b) Betrachte z.B. die Grammatik G mit $V = \{A, B, C\}$, $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$, Startsymbol A und Regelmenge R:

$$
\begin{aligned}
A &\to B \mid \mathsf{a} \\
B &\to C \mid \mathsf{b} \\
C &\to AB \mid \mathsf{c}
\end{aligned}
$$

Schritt 1 bis Schritt 4, Phase 1 lassen diese Grammatik unverändert. In Schritt 4, Phase 2 ist die einzige mögliche topologische Sortierung A, B, C. Wird beim Ersetzen der Kettenregeln aber z.B. in der Reihenfolge A, B, C vorgegangen, erhalten wir folgendes Ergebnis:

$$
\begin{aligned}
A &\to C \mid \mathsf{a} \\
B &\to AB \mid \mathsf{b} \\
C &\to AB \mid \mathsf{c}
\end{aligned}
$$

Es gibt danach also immer noch eine Kettenregel $A \to C$, bei der C noch durch AB ersetzt werden müsste. Das wurde aber nicht gemacht, weil A vor C abgearbeitet wurde.

Aufgabe 3

(2 + 4 = 6 Punkte)

Zeigen Sie, dass die kontextsensitiven Sprachen den Sprachen der Klasse $\mathsf{NTAPE}(n)$ entsprechen[1], indem Sie in zwei Schritten vorgehen.

(a) Sei L eine kontextsensitive Sprache. Dann existiert eine nichtdeterministische Turingmaschine M, die L mit linearem Platzbedarf akzeptiert.

(b) Sei M eine nichtdeterministische Turingmaschine, die die Sprache $L(M)$ mit linearem Platzbedarf akzeptiert. Dann existiert eine kontextsensitive Grammatik G, so dass $L(G) = L(M)$ gilt.

Lösung:

[1] Vergleiche Vorlesung 13, Folie 18.

(a) Sei L eine kontextsensitive Sprache, und G eine Typ-1-Grammatik, die L erzeugt. Zeige, dass L von einer nichtdeterministischen Turingmaschine T mit linearem Platzbedarf entschieden werden kann. Die Grammatik G hat konstante Größe und lässt sich damit in $\mathcal{O}(1)$ Platz auf das Band von T schreiben. Sei nun w eine Eingabe für T. Es gilt zu entscheiden, ob $w \in L$ ist. Im Falle $w = \varepsilon$ gilt $w \in L$ genau dann, wenn G die Ableitungsregel $S \to \varepsilon$ enthält. Andernfalls wird wie folgt vorgegangen. Enthält w ein Teilwort v, so dass G eine Regel $u \to v$ enthält, ist $u \to v$ eine *mögliche* Ableitung. Die NTM T sucht in jedem Schritt erst alle möglichen Ableitungen und wählt danach eine nichtdeterministisch aus. Dann wird v durch u ersetzt. Steht irgendwann nur noch S auf dem Band, hält T und akzeptiert. Da T nur Ableitungen vornimmt, die durch G kodiert werden, gilt $w \in L$, falls T akzeptiert. Umgekehrt akzeptiert T, falls $w \in L$ gilt, z.B. indem die Ableitungen, die von S zu w führen, in umgekehrter Reihenfolge vorgenommen werden. Man beachte, dass die Eingabe beim umgekehrten Anwenden der Ableitungen immer kürzer wird. Zusammen mit der in $\mathcal{O}(1)$ kodierten Grammatik wird so also höchstens linear viel Platz benötigt.

(b) Sei nun $L \in \mathsf{NTAPE}(n)$ und $T = (Q, \Sigma, \Gamma, s, \delta, F)$ eine NTM, die L akzeptiert. Wir konstruieren eine Grammatik G, die gerade die Sprache L erzeugt. Das Terminalalphabet von G sei gerade Σ. Die Idee ist es, die Kopfbewegungen und Zustandsübergänge von T als Ableitungen in G zu kodieren. Als erstes brauchen wir ein Band, das aus mindestens einem Bandsymbol besteht. Jedes Bandsymbol ist ein Tripel der Form $(Q \cup \{\bot\}) \times \Sigma \times \Gamma$. Dabei steht der erste Wert für den Zustand, bzw. $\bot$ falls der Lesekopf nicht auf diesem Zeichen steht. Der zweite Wert steht für das Eingabesymbol auf dem Eingabeband und der dritte Wert für das Bandsymbol auf dem Arbeitsband.

$$\begin{aligned} \forall \sigma \in \Sigma: \quad & S \;\to\; (s, \sigma, \sqcup) \mid (s, \sigma, \sqcup)\, B \\ \forall \sigma \in \Sigma: \quad & B \;\to\; BB \mid (\bot, \sigma, \sqcup) \end{aligned}$$

Man beachte, dass der Kopf von T zunächst im Zustand s auf dem linkesten Symbol der Eingabe steht. Außerdem lässt sich durch die obigen Regeln jedes Wort auf dem Eingabeband erzeugen. Die Idee ist es nun, dass eine solche Eingabe genau dann ableitbar sein soll, wenn M diese akzeptiert. Betrachte deshalb für alle $q \in Q \setminus F, \sigma, \sigma' \in \Sigma, \gamma, \gamma' \in \Gamma$ den Übergang $\delta(q, \gamma) = (\hat{q}, \hat{\gamma}, d)$ mit $d \in \{L, N, R\}$ und füge folgende Regeln hinzu:

$$\begin{aligned} (\bot, \sigma', \gamma')\ (q, \sigma, \gamma) &\;\to\; (\hat{q}, \sigma', \gamma')\ (\bot, \sigma, \hat{\gamma}) && \text{falls } d = L \\ (q, \sigma, \gamma) &\;\to\; (\hat{q}, \sigma, \hat{\gamma}) && \text{falls } d = N \\ (q, \sigma, \gamma)\ (\bot, \sigma', \gamma') &\;\to\; (\bot, \sigma, \hat{\gamma})\ (\hat{q}, \sigma', \gamma') && \text{falls } d = R \end{aligned}$$

Sobald ein akzeptierender Zustand $f \in F$ erreicht ist, muss das ursprüngliche Wort entpackt werden.

$$\begin{aligned} (f, \sigma, \gamma) &\;\to\; \sigma \\ (\bot, \sigma, \gamma)\ \sigma' &\;\to\; \sigma\ \sigma' \\ \sigma'\ (\bot, \sigma, \gamma) &\;\to\; \sigma'\ \sigma \end{aligned}$$

Ist $\varepsilon \in L$, fügen wir außerdem noch die Regel $S \to \varepsilon$ hinzu.

Aufgabe 4 (3 Punkte)

Betrachten Sie eine Verallgemeinerung von kontextfreien Grammatiken, bei der die rechte Seite von Produktionen nicht ein Wort über dem Alphabet $\Sigma \cup \Gamma$ ist, sondern ein regulärer Ausdruck.

Beim Anwenden einer Produktionsregel $A \to R(A)$ kann dann ein Nichtterminalsymbol A durch ein beliebiges Wort aus der durch den regulären Audruck $R(A)$ erzeugten Sprache $L(R(A))$ ersetzt werden.

Beweisen Sie, dass solche verallgemeinerten kontextfreien Grammatiken nicht mächtiger sind als „herkömmliche“ kontextfreie Grammatiken, dass sie also genau dieselben Sprachen erzeugen.

Lösung:
Wir konstruieren zu einer beliebigen verallgemeinerten kontextfreien Grammatik eine „herkömmliche“ kontextfreie Grammatik. Zu jedem regulären Ausdruck R gibt es eine rechtslineare Grammatik G mit $L(G) = L(R)$. Konstruiere also zu jeder Produktionsregel $A \to R(A)$ eine Grammatik $G(A)$ mit $L(G(A)) = L(R(A))$. Ersetze dann die Regel $A \to R(A)$ durch die Regel $A \to S(G(A))$, wobei $S(G(A))$ das Startsymbol von $G(A)$ bezeichnet.

Aufgabe 5 (1 + 5 = 6 Punkte)

(a) Zeigen Sie, dass reguläre Sprachen unter Schnitt abgeschlossen sind.

(b) Zeigen Sie, dass die Sprache

$$L = \{\langle G_1, G_2\rangle \mid G_1, G_2 \text{ rechtslineare Grammatiken mit } L(G_1) = L(G_2)\}$$

entscheidbar ist.

Anmerkung: Für kontextfreie Grammatiken ist die entsprechende Sprache nicht entscheidbar.

Lösung:

(a) Seien L_1, L_2 reguläre Sprachen und δ_1, δ_2 die Übergangsfunktionen der deterministischen endlichen Automaten, die L_1, L_2 entscheiden. Dann entscheidet der Produktautomat mit Übergangsfunktion $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta(q_2, a))$ den Schnitt von L_1, L_2.

(b) Zu jeder rechtslinearen Grammatik G lässt sich einfach ein endlicher Automat $\mathcal{A}$ mit $L(\mathcal{A}) = L(G)$ konstruieren. In diesem existiert zu jeder Produktionsregel $Q \to aQ'$ aus G ein entsprechender Übergang $\delta(Q, a) = Q'$. Mit der Potenzmengenkonstruktion haben wir ein Verfahren kennengelernt, mit dem wir sicherstellen können, dass es einen zu $\mathcal{A}$ äquivalenten deterministischen endlichen Automaten $\mathcal{A}'$ gibt. Reguläre Sprachen sind unter Komplementbildung abgeschlossen, das Komplement von $L(\mathcal{A}')$ lässt sich durch einen deterministischen endlichen Automaten $\mathcal{A}''$, der aus $\mathcal{A}'$ durch Invertierung des Akzeptanzverhaltens aller Zustände hervorgeht, entscheiden. Schließlich erkennt ein endlicher Automat genau dann die leere Sprache, wenn er nach entfernen aller überflüssigen (also nicht erreichbaren) Zustände keinen akzeptierenden Zustand mehr hat.

Mit dem beschriebenen Vorgehen lässt sich also prüfen, ob $L(G_1) \cap L(G_2)^c = \emptyset$ und $L(G_1)^c \cap L(G_2) = \emptyset$ gilt, was die Entcsheidbarkeit von L bezeugt.

Aufgabe 6 (4 Punkte)

Zeigen Sie, dass folgende Sprachen nicht kontextfrei sind[2]:

(a) $L_1 = \{\mathtt{0}^n\mathtt{1}^n\mathtt{2}^n \mid n \in \mathbb{N}\}$

(b) $L_2 = \{\mathtt{0}^i\mathtt{1}^j\mathtt{2}^k \mid i < j < k \in \mathbb{N}\}$

(c) $L_3 = \{0^{n^3}1^k \mid k, n \in \mathbb{N}\}$

Lösung:

(a) Wende das Pumpinglemma an und wähle $w = \mathtt{0}^n\mathtt{1}^n\mathtt{2}^n$. Wegen $|vwx| \leq n$ enthält vx mindestens ein Zeichen a aus $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$ nicht. Dann enthält uv^2wx^2y zu wenige Vorkommen des Zeichens a.

(b) Wende das Pumpinglemma an und wähle $z = \mathtt{0}^n\mathtt{1}^{n+1}\mathtt{2}^{n+2}$. Wegen $|vwx| \leq n$ enthält vx mindestens ein Zeichen aus $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$ nicht. Enthält vx keine $\mathtt{0}$, so enthält uv^0wx^0y zu wenige Einsen, um zu L_3 zu gehören. Sonst enthält uv^3wx^3y zu wenige Zweien, um zu L_2 zu gehören.

(c) Sei n die vermeintliche Konstante aus dem Pumpinglemma. Wähle $z = \mathtt{0}^{n^3}\mathtt{1}$. Kommt die $\mathtt{1}$ in vx vor, so gilt $uv^0wx^0y \notin L_3$. Ansonsten bestehen v und x aus höchstens n Vorkommen des Zeichens $\mathtt{0}$. Dann enthält das Wort uv^2wx^2y mindestens n^3+1 und höchstens n^3+n Mal das Zeichen $\mathtt{0}$. Da aber das Intervall $(n^3, (n+1)^3 = n^3 + 3n^2 + 3n + 1)$ keine Kubikzahl enthält gilt $uv^2wx^2y \notin L_4$. Aus dem Pumpinglemma folgt dann, dass L_3 nicht kontextfrei ist.

[2]Dazu können Sie z.B. das Pumpinglemma für kontextfreie Sprachen verwenden, dass am 17.01. in der Vorlesung behandelt wird.