

Übungsblatt 6

Vorlesung Theoretische Grundlagen der Informatik im WS 19/20

Ausgabe: 8. Dezember 2019

Abgabe: 21. Januar 2020, 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 $\Sigma = \{a, b, c, d\}$, Nichtterminalen $V = \{S, A, B, C, D\}$, Startsymbol S und Produktionen

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

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

c	b	a	c	d

- (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!

Aufgabe 2

(2 + 2 = 4 Punkte)

Betrachten Sie eine Relaxierung der Chomsky-Normalform, bei der die rechten Seiten von Ableitungsregeln auch aus drei aufeinanderfolgenden Nichtterminalsymbolen bestehen dürfen.

- (a) Passen Sie den CYK-Algorithmus an, um mit solchen Grammatiken umgehen zu können, d.h., geben Sie einen Algorithmus an, der das Wortproblem für derartige Grammatiken in Polynomialzeit löst.
- (b) Was ist das Laufzeitverhalten Ihres Algorithmus? Beweisen Sie!

Aufgabe 3

(2 + 2 + 2 = 6 Punkte)

Beim CYK-Algorithmus wird ausgenutzt, dass ein Wort w von einem Nichtterminalsymbol V genau dann ableitbar ist, wenn eine der folgenden Aussagen wahr ist:

- w ist ein einzelnes Terminalsymbol und es existiert eine Regel $V \rightarrow w$.
- w lässt sich in zwei Teile w_1w_2 zerlegen, und es existiert eine Regel $V \rightarrow WX$ sodass w_1 von W und w_2 von X ableitbar ist.

Der CYK-Algorithmus benutzt diesen Zusammenhang und dynamische Programmierung, um das Wortproblem für Grammatiken von Typ 2 in kubischer Zeit zu lösen.

Betrachten Sie nun Typ-3-Grammatiken.

- (a) Wann lässt sich in diesem Fall ein Wort w aus einem Symbol V ableiten?
- (b) Beschreiben Sie einen **möglichst effizienten** Algorithmus, der – analog zum CYK-Algorithmus für Grammatiken von Typ 2 – das Wortproblem für Grammatiken von Typ 3 löst.
- (c) Nennen und begründen Sie Laufzeit- und Speicherkomplexität Ihres Algorithmus.

Aufgabe 4

(3 + 1 = 4 Punkte)

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

$$\begin{aligned} S &\rightarrow aSc \mid X \mid dY \\ X &\rightarrow bXc \mid \varepsilon \\ Y &\rightarrow dS \mid \varepsilon \end{aligned}$$

- (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 $A \rightarrow B$ 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.

Aufgabe 5

(2 + 4 = 6 Punkte)

Zeigen Sie, dass die kontextsensitiven Sprachen den Sprachen der Klasse $\text{NTAPE}(n)$ entsprechen¹, 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.

Aufgabe 6

(4 Punkte)

Zeigen Sie, dass folgende Sprachen nicht kontextfrei sind²:

- (a) $L_1 = \{0^n 1^{2n} 0^n \mid n \in \mathbb{N}\}$
- (b) $L_2 = \{aa \mid a \in \{0, 1\}^*\}$
- (c) $L_3 = \{a\#b\#c \mid a, b, c \in \{0, 1\}^*, a + b = c \text{ als Binärzahlen aufgefasst}\}$

¹Vergleiche Vorlesung 13, Folie 18.

²Dazu können Sie z.B. das Pumping-Lemma für kontextfreie Sprachen verwenden, das am 09.01. in der Vorlesung behandelt wird.