RWTH Aachen **SS 2017**
Lehrgebiet Theoretische Informatik Blatt 6
Rossmanith–Dreier–Hark–Kuinke 12.6.2017

Übung zur Vorlesung Formale Sprachen, Automaten und Prozesse

Aufgabe T18

Entwerfen sie für die folgenden Sprachen L_i jeweils eine kontextfreie Grammatik, die L_i erzeugt.

a) $L_1 = \{\, a^n b^n \mid n \in \mathbf{N} \,\}$ über dem Alphabet $\{a, b\}$

b) $L_2 = \{\, w \in \Sigma^* \mid w = w^R \,\}$, die Sprache der Palindrome über $\Sigma = \{a, b\}$

c) $L_3 = \{\, w \in \Sigma^* \mid w \neq w^R \,\}$, die Sprache der Nicht-Palindrome über $\Sigma = \{a, b\}$

d) L_4, die Sprache der Texte über dem Alphabet $\{a, \ldots, Z, 0, \ldots, 9\}$ mit `<i>`*italic*`</i>`- und `<b>`**boldface**`</b>`-Tags. Achten Sie darauf, daß die Tags vernünftig geschachtelt sind. Zum Beispiel ist `<b> <i> </b> </i>` nicht gültig.

Aufgabe T19

Erstellen Sie Ableitungsbäume und Linksableitungen für die Grammatik

$$S \to ABS \mid B, \quad A \to AB \mid a, \quad B \to AS \mid BS \mid b \mid \epsilon$$

und die Wörter *aba* und *ababb*. Ist diese Grammatik eindeutig?

Aufgabe T20

Gegeben sei die Grammatik mit den Produktionen

$$S \to aA \mid bB \mid SAB, \quad A \to Sa \mid b, \quad B \to BA \mid AS \mid ab$$

Benutzen Sie einen saturierten NFA für eine geeignete Sprache L, um anhand von $pre_G^*(L)$ zu überprüfen, ob G das Wort *ababba* erzeugen kann.
Bilden Sie zwei Gruppen, um zu untersuchen, ob die folgenden α auf *ababba* ableitbar sind. Die erste Gruppe soll dies anhand der Grammatik G selbst, die zweite anhand von $pre_G^*(L)$ erledigen: (a) $\alpha = aSA$, (b) $\alpha = BBa$, (c) $\alpha = bBAS$, (d) $\alpha = aA$, (e) $\alpha = B$, (f) $\alpha = Ba$, (g) $\alpha = BA$, (h) $\alpha = AS$

Aufgabe H12 (10 Punkte)

Gegeben sei die Grammatik G mit den Produktionen

$$S \to 01A, \quad A \to 0 \mid 0S \mid AA$$

Ferner sei L die Sprache aller Wörter über $\{0, 1\}^*$, in denen 010 nicht als Teilwort vorkommt. Was ist $pre_G^*(L)$? Für welche $\alpha \in L$ gilt $A01AA1100 \Rightarrow^* \alpha$? Was ist $L(G) \cap L$?

Aufgabe H13 (15 Punkte)

In Aufgabe T17 wurde ein Vorschlag erarbeitet, wie man *first longest match* implementieren könnte. Dies ist sehr aufwendig, umfaßt es ja Konstruktionen wie das Entfernen von ϵ-Kanten, die Potenzmengenkonstruktion und das Minimieren von DFAs. Sie können dennoch das ganze Verfahren implementieren, wenn Sie wollen, sollten das aber in einer Gruppe machen und die Arbeit aufteilen.

Stattdessen können Sie auch eine einfachere Aufgabe lösen: Überlegen Sie sich ein einfacheres Verfahren, um einen Spezialfall von T17 zu lösen: Die regulären Ausdrücke umfassen neben r_{num} und r_{id}, welche am Ende der Liste stehen müssen, nur reguläre Ausdrücke, welche ein einzelnes Wort beschreiben (wie es auch im Beispiel in T17 der Fall ist). Es ist nicht schwer, in diesem Fall direkt einen geeigneten DFA zu konstruieren und während dieser simuliert wird, die entsprechenden u_i zu bestimmen.

Beschreiben Sie, wie Ihr Verfahren funktioniert und implementieren Sie es. Testen Sie es an aussagekräftigen Beispielen.

Ihr Programm sollte als Ausgabe eine Liste von *Token* liefern, welche wenigstens das jeweilige u_i und einen symbolischen Namen, der den regulären Ausdruck identifiziert, beinhaltet.

Geben Sie auch den Quelltext des Programms sowie Protokolle von Testläufen ab. Diese Protokolle mögen auch die jeweiligen Listen von *Token* in übersichtlicher Form enthalten.

Aufgabe H14 (7 Punkte)

Gegeben sei die kontextfreie Grammatik aus Aufgabe T20:

$$S \to aA \mid bB \mid SAB, \quad A \to Sa \mid b, \quad B \to BA \mid AS \mid ab$$

a) Gilt $babb \in L(G)$?

b) Gilt $babba \in L(G)$?

Beweisen Sie Ihre Antworten!

Aufgabe H15 (viele Bonuspunkte)

Bonuspunkte werden nur für vollständige und korrekte Lösungen vergeben! Falls Sie große Lust haben, diese Aufgabe zu lösen, aber es in einer Woche nicht schaffen können, dürfen Sie sie auch in zwei Wochen abgeben.

Ist folgende kontextfreie Grammatik eindeutig?

$$\begin{aligned} S &\to A \mid B \\ A &\to aAa \mid abAb \mid bAc \mid aca \mid abcb \mid bcc \\ B &\to bBa \mid aBb \mid babBc \mid bca \mid acb \mid babcc \end{aligned}$$

Beweisen Sie entweder die Eindeutigkeit oder geben Sie zwei verschiedene Linksableitungen für dasselbe Wort an.