Universität Paderborn
Prof. Dr. Johannes Blömer
Prof. Dr. Eyke Hüllermeier

13. Januar 2017

Modellierung – WS 2016/2017

Heimübung 11
Abgabe: 23. Januar 2017 – 14:00 Uhr
(Dieser Übungszettel enthält 5 Aufgaben mit insgesamt 32 Punkten)

Hinweis: Die Lösungen der Hausaufgaben sind in die Kästen im D3-Flur einzuwerfen. Bilden Sie Gruppen von 3-4 Personen zur Lösung der Aufgaben. Die Lösung muss die Namen und Matrikelnummern derjenigen enthalten, die die Aufgaben gelöst haben, sowie die **Übungsgruppennummer**. Nicht getackerte Abgaben werden nicht korrigiert.

Aufgabe 1 (Abhängigkeitsgraph) (5 Punkte)
Ein Koch serviert seinen Gästen klassische Pasta mit Tomatensauce. Hierbei sind die Arbeitsschritte A_1 bis A_8 nötig, die der Koch mit seinem Team erledigt. Dabei ist die Reihenfolge der Ausführung der einzelnen Arbeitsschritte zu beachten.

A_1 Tisch decken, Dauer: 6 Minuten

A_2 Wasser zum Kochen bringen, Dauer: 5 Minuten

A_3 Nudeln in kochendem Wasser kochen, Dauer: 9 Minuten

A_4 Tomaten waschen, Tomaten und Zwiebeln schneiden, Dauer: 8 Minuten

A_5 Tomatensauce abschmecken, Dauer: 1 Minute

A_6 Tomatensauce kochen, Dauer: 15 Minuten

A_7 Essen servieren, Dauer: 3 Minuten

A_8 Töpfe Abwaschen, Dauer: 15 Minuten

1. Modellieren Sie die Abhängigkeiten zwischen den Aktionen und ihre Ausführungsreihenfolgen durch einen gerichteten Graphen $G = (V, A)$ mit weniger als 10 Kanten. Geben Sie eine Knotenmarkierung $m : V \to \mathbb{N}$ an, die die Dauer der jeweiligen Aktionen beschreibt.

2. Geben Sie eine topologische Sortierung des Graphen an.

3. Geben Sie den kritischen Pfad des Graphen nach der erweiterten Definition aus Präsenzübung 11, Aufgabe 1 an.

4. Berechnen Sie die Zeit die mindestens benötigt wird, bis das Essen serviert werden kann.

Aufgabe 2 (Grammatiken) (7 Punkte)

Gegeben Sei die folgende Grammatik $G = (T, N, P, S)$ mit

$$
\begin{aligned}
T &= \{(\ ,)\} \\
N &= \{Klammerung, Liste\} \\
S &= \ \ Klammerung \\
P &= \{Klammerung ::= \text{‘(‘}Liste\text{‘)‘}, \\
& \quad\ Liste ::= Klammerung\ Liste, \\
& \quad\ Liste ::= \epsilon \\
& \quad \}
\end{aligned}
$$

1. Modifizieren Sie die Grammatik G zu $\tilde{G}$, so dass diese die Sprache der Klammerausdrücke erzeugt[1].

2. Definieren Sie nun eine kontextfreie Grammatik G_2 indem Sie $\tilde{G}$ so modifizieren, dass die von G_2 erzeugte Sprache zusätzlich auch Wörter mit geschweiften Klammerpaaren { } enhält. Dabei soll jede Klammer immer von einer Klammer des gleichen Typs wieder geschloßen werden. In der von G_2 erzeugten Sprache soll also zum Beispiel *nicht* das Wort ({)} enthalten sein.

3. Definieren Sie nun eine kontextfreie Grammatik G_3 indem Sie G_2 so modifizieren, dass in Wörtern aus $L(G_3)$ zwischen runden Klammern () nie geschweifte Klammern {} auftreten. So soll zum Beispiel das Wort {()} in $L(G_3)$ enthalten sein, nicht jedoch das Wort ({}). Dabei soll diese Grammatik maximal 8 Produktionen enthalten.

Aufgabe 3 (Sprachen, Grammatiken) (8 Punkte)

Gegeben seien die folgende Sprachen

$$
\begin{aligned}
L_1 &= \{(010)^n aa1^m \text{ mit } n \geq 1 \text{ und } m \geq 0\} \cup \{(010)^n aa0^m \text{ mit } n \geq 0 \text{ und } m \geq 1\} \\
L_2 &= \{(010)^n aa1^m \text{ mit } n \geq 0 \text{ und } m \geq 1\} \cup \{(010)^n aa0^m \text{ mit } n \geq 0 \text{ und } m \geq 1\}
\end{aligned}
$$

1. Geben Sie eine Grammatik G_1 an, die die Sprache L_1 erzeugt.

2. Geben Sie eine Grammatik G_2 an, die die Sprache L_2 erzeugt.

3. Wie sieht die formale Beschreibung von $L_1 \cap L_2$ aus? Geben Sie sie an.

4. Geben Sie eine Grammatik G_3 an, die den Durchschnitt der beiden Sprachen erzeugt.

Anmerkung: $(010)^n$ bedeutet (analog zu 1^n), dass das Wort 010 n-mal wiederholt wird.

[1] Siehe Foliensatz 08-Grammatiken, Folie 2

Aufgabe 4 (Grammatiken, Mehrdeutigkeit) (7 Punkte)
Grammatiken finden praktische Anwendung in der Übersetzung höherer Programmiersprachen. Ein *Parser* überprüft die syntaktische Korrektheit eines Quellcodes anhand einer zuvor definierten sprachspezifischen Grammatik. Betrachten Sie folgenden (vereinfachten) Ausschnitt aus der ANSI C Grammatik:

$$G_C = (T_C, N_C, P_C, statement)$$
$$T_C = \{IF, ELSE, (,), expression\}$$
$$N_C = \{statement, selection\_statement\}$$

$P_C =$

$$\begin{array}{llll} \{ & statement & ::= & selection\_statement\ , \\ & statement & ::= & expression\ , \\ & selection\_statement & ::= & IF\ (\ expression\ )\ statement \\ & selection\_statement & ::= & IF\ (\ expression\ )\ statement\ ELSE\ statement \quad \} \end{array}$$

1. Zeigen Sie, dass G_C mehrdeutig ist, indem Sie ein Wort und zwei unterschiedliche zugehörige Ableitungsbäume angeben.
2. Was ist der semantische Unterschied zwischen den beiden von Ihnen angegebenen Bäumen?
3. Recherchieren und erläutern Sie: Wie wird dieses Problem typischerweise aufgelöst?
4. Tritt dieses Problem auch in den Sprachen Java und Python auf? Wenn nein, warum nicht?

Aufgabe 5 (Grammatik bilden) (5 Punkte)
Die Sprache der Palindrome lässt sich durch die drei folgenden Bildungsregeln beschreiben:

- Das leere Wort ε sowie 0 und 1 sind Palindrome.
- Falls w ein Palindrom ist, sind $0w0$ und $1w1$ Palindrome.
- Alle Palindrome lassen sich durch endlich viele Anwendungen der ersten beiden Regeln erzeugen.

1. Geben Sie eine kontextfreie Grammatik $G = (T, N, P, S)$ mit höchstens 7 Ableitungsregeln an, die die Sprache der Palindrome erzeugt.
2. Untersuchen Sie, ob die folgenden Worte Element der durch G erzeugten Sprache sind:
   a) 010111010
   b) 1101101011

   Geben Sie jeweils den zugehörigen Ableitungsbaum an bzw. begründen Sie, warum das Wort kein Element der durch Ihre Grammatik definierten Sprache ist.