
Übungsblatt Nr. 3 (30 Punkte)

Ausgabedatum: 17. 5. 2010, 17:00 – Abgabedatum: 31. 5. 2010, 8:00

Die Lösung der Aufgaben soll in Gruppen zu je zwei Studierenden erfolgen. Die Aufgaben sind elektronisch abzugeben. Auf jedem Lösungsblatt sind Name und Matrikelnummer der beiden Gruppenmitglieder anzugeben! Von jedem Blatt werden mindestens 20% der Punkte (also 6) verlangt.

Aufgabe 1 (6 Punkte)

Geben Sie jeweils eine kontextfreie Grammatik G_1 und G_2 für folgende Sprachen an:

- a) (3 Punkte) $b^3 (ab)^n b^3$ für $n > 0$, d.h. $S_1 = \{bbbabbbb, bbbababbbb, \dots\}$
b) (3 Punkte) $S_2 = \{m, n, o, mm, nn, oo, mno, nmo\}$

Aufgabe 2 (6 Punkte)

Euklids Algorithmus für den größten gemeinsamen Teiler (GGT):

Es gilt

$$\begin{aligned} \text{GGT}(x, y) &= \text{GGT}(y, x \text{ MOD } y), \text{ falls } y > 0 \\ \text{GGT}(x, y) &= x \text{ falls } y = 0. \end{aligned}$$

Entwickeln Sie aus diesen Beziehungen einen Algorithmus in Groovy/Java zur Berechnung des größten gemeinsamen Teilers $\text{GGT}(a,b)$ mit $a,b \in \mathbb{N}$.

Aufgabe 3 (6 Punkte)

- a) Wandeln Sie die Zahl 01111011_2 in die folgenden Darstellungen um!
- Oktalzahl
 - Dezimalzahl
 - Hexadezimalzahl
 - Zahl zur Basis 11
- b) Wandeln Sie die Zahl -20_{10} ins Zweierkomplement zur Basis 2 und 8 um (jeweils 8 Bit)
- c) Rechnen Sie im Binärsystem $100_2 * 111_2$.
- d) Rechnen Sie im System zur 7-er Basis $1234_7 / 5_7$. Hinweis: Division mit Rest.

Aufgabe 4 (8 Punkte)

- a) (4 Punkte) Geben Sie eine rekursive Definition der Funktion *teilwort*, welche zu zwei gegebenen Wörtern u und v den Wert *true* liefert, genau dann, wenn u als Teilwort in v vorkommt.
- b) (4 Punkte) Geben Sie eine rekursive Definition der Funktion *ersetze*, welche zu gegebenen Wörtern u , v und w dasjenige Wort liefert, das entsteht, indem das erste Vorkommen von u in w durch v ersetzt wird.
- c) (**Zusatz**) Erweitern Sie Ihre Lösung aus Teilaufgabe b) so, dass jedes Vorkommen von u durch v ersetzt wird.

Hinweis: Als Programmiersprache soll Groovy/Java verwendet werden.

Aufgabe 5 (4 Punkte)

Eine spezielle Form der Rekursion ist die sogenannte „Tail-end“-Rekursion. Es sind Rekursionen, deren rekursiver Schritt als allerletztes geschieht, dem also keinerlei weitere Berechnungen folgen.

Schema:

```
func(n) {  
  if (Abbruchbedingung) {  
    return Wert(n)  
  } else {  
    m = berechne_einfacheres(n)  
    return func(m) // Rekursion  
  }  
}
```

- a) (4 Punkte) Welchen Vorteil hat eine solche spezielle Form der Rekursion gegenüber der „normalen“ Rekursion?
- b) (**Zusatz**) Geben Sie die Fakultäts-Funktion ($n!$) als „Tail-end“-Rekursion in Groovy/Java an.