Informatik I WS 2005/2006

Übungsblatt 4

Abgabe: 28.11.05 - 30.11.05

Aufgabe 1

Punkte: 6

a) Die Toy-Machine beherrscht bislang nur die arithmetischen Operationen „Addition“, „Subtraktion“ und die in der Vorlesung bereits vorgestellte „Multiplikation“. Nun sollen Sie ihr auch noch die „Divison“ beibringen. Mit Divison ist hier die ganzzahlige Divison ohne Rest und Runden gemeint.
Also ist z.B. 22 DIV 7 = 3 und auch 27 DIV 7 = 3.
Schreiben Sie ein Programm div.toy, welches zwei Zahlen a und b vom Stdin einliesst und a DIV b auf dem Stdout ausgibt.

b) Schreiben Sie ein Programm kgv.toy, das zwei Zahlen vom Stdin einliesst und das kleinste gemeinsame Vielfache dieser beiden Zahlen auf dem Stdout ausgibt.
Tip: Verwenden Sie bereits bekannte Programme (dazu zählen auch die Beispielprogramme der Toy-Machine) als Unterprogramme.

Aufgabe 2

Punkte: 3

In dieser Aufgabe soll ein Algorithmus zur Wechselgeldrückgabe entwickelt werden. Dabei soll ein fester Betrag zwischen 0 und 100 Cent gewechselt werden. Es stehen dazu genügend Münzen im Wert von 1, 2, 5, 10, 20, 50 Cent und 1 Euro zur Verfügung. Ziel ist es, mit möglichst wenigen Münzen auszukommen.

a) Entwerfen Sie eine Spezifikation für einen solchen Algorithmus.

b) Geben Sie den Algorithmus im Pseudocode an.

Tip: Verwenden Sie die in Aufgabe 1 beschriebene ganzzahlige Division ohne Runden, sowie den Modulo-Operator.

Aufgabe 3

Punkte: 3

Im Kapitel 1 der Vorlesung („Repräsentation voz Daten“) haben Sie bereits die „Betrag und Vorzeichen“-Darstellung von ganzen Zahlen kennengelernt.
In dieser Aufgabe soll nun ein total korrekter Algorithmus zur Umrechnung von ganzen Zahlen im Dezimalsystem in die binäre „Betrag und Vorzeichen“-Darstellung entwickelt werden. Dabei soll das niederwertigste Bit der Einfachheit halber zuerst ausgegeben werden. Zur Ausgabe der Zahlen stehen Ihnen die Funktionen gib_0_aus() und gib_1_aus() zur Verfügung.

a) Entwerfen Sie eine Spezifikation für einen solchen Algorithmus.

b) Veranschaulichen Sie die Funktionsweise des Algorithmus durch ein Flußdiagramm.

Aufgabe 4

Punkte: 8

a) Entwerfen Sie eine Turingmaschine zur Berechnung des logischen „OR“ zweier binärer Zeichenketten x und y. Nehmen Sie dazu an, dass die Eingabe folgendes Format hat:
$ x # y $ $ $...
Die beiden Eingabeworte x und y sollen dabei die gleiche Länge haben. Zu Beginn steht der Lese-/Schreibkopf unter dem ersten Zeichen von x. Nach der Berechnung soll das Band dann folgendermassen aussehen:
$ z # y $ $ $ ...
Dabei ist z = x+y und „+“ die logische „OR“-Verknüpfung. Der Lese-/Schreibkopf soll nach Beendigung des Programms wieder an seiner ursprünglichen Position stehen.
Beispiel:
Vor der Berechnung:

$	1	0	0	#	0	1	0	$	$	...
	↑									

Nach der Berechnung:

$	1	1	0	#	0	1	0	$	$	...
	↑									

Verwenden Sie als Bandalphabet Σ={$, #, 0, 1, $\overline{0}$, $\overline{1}$}.
Tip: Die Zeichen $\overline{0}$ und $\overline{1}$ kann man zur Markierung bestimmter Zeichen verwenden.

b) Zeichnen Sie den Übergangsgraphen für die in Teilaufgabe a) entworfene Turingmaschine.

Eine Neuerung in den Übungen:
Ab sofort besteht die Möglichkeit, sich durch Vorrechnen an der Tafel während der Übungen zusätzliche Punkte zu verdienen!
Dadurch kann man beispielsweise ein Unterschreiten der 30%-Marke bei einem Übungszettel wieder ausgleichen. Auch bei der Bewertung von Grenzfällen bei der abschliessenden Scheinvergabe (wenn also beispielsweise nur ein paar Punkte zur 50%-Gesamtmarke fehlen) wird die Mitarbeit in den Übungen positiv berücksichtigt werden.
Es empfiehlt sich daher, von dieser neuen Regelung regen Gebrauch zu machen.
Darüber hinaus stellt das Vorrechnen an der Tafel auch eine gute Vorbereitung für spätere Seminarvorträge dar und durch Ihre aktive Mitarbeit werden die Übungsstunden auch gleich wesentlich interessanter.
Also: Tun Sie sich was Gutes und ran an die Tafel.