Übungen zur Vorlesung
Grundlagen der Programmierung I
Blatt 8

Aufgabe 1:

Wir erweitern die Überlegungen zu Aufgabe 4 von Blatt 7 zu einem einfachen Übersetzer von einer Teilmenge von PRO (genannt PRO-0) nach ASS. PRO-0 enthält nur den Typ Zahl. Bedingungen haben die Form A op B, wobei A und B Bezeichner oder Zahlen und op einer der Vergleichsoperatoren ist. Zuweisungen haben die Form A←B op C oder A←B mit A Bezeichner, B,C Bezeichner oder Zahl und op eine der vier arithmetischen Operationen +,-,*,/. Ferner gibt es die bekannten Ein- und Ausgabeanweisungen lies und zeige.

Schreiben Sie nun für jedes Sprachelement aus PRO-0 ein Übersetzungsschema, also für lies, zeige, A←B op C, für die Konkatenation, für die Deklarationen (wie werden die übersetzt??) usw.

Beschreiben Sie anschließend algorithmisch, wie man unter Nutzung Ihrer Schemata ein PRO-0-Programm nach ASS übersetzt.

Übersetzen Sie dann mit Ihren Schemata schrittweise das folgende Programm der Vorlesung:

```
def x, n, größer, kleiner: Zahl
größer←0
kleiner←0
lies(n)
lies(x)
solange x≠0 tue
        wenn x≤n dann
                kleiner←kleiner+x
        sonst
                größer←größer+x
        ende
        lies(x)
ende
zeige(größer)
zeige(kleiner).
```

Fällt Ihnen gegenüber dem ASS-Programm der Vorlesung etwas auf?

Zur Erinnerung: Für bedingte Anweisung und Schleife lauten die Übersetzungsschemata gem. Aufgabe 4 (Blatt 7):

```
wenn B dann                          solange B tue
        A              und                   A
sonst                                ende
        A'
ende
```

```
        c(B)                                 m:    c(B)
        iftruegoto m                               iftruegoto m'
        c(A')                                      c(A)
        goto m'                              goto m
m:      c(A)                                 m':   …
m':     …
```

mit neuen Marken m und m' mit neuen Marken m und m'

Aufgabe 2: (Kürzeste Wege nach E.W. Dijkstra)
Entwickeln Sie einen umgangssprachlichen Algorithmus, der das allgemeine Problem, kürzeste Wege in Autobahnnetzen zu finden, löst. Ein Autobahnnetz sei durch einen Graphen modelliert wie in Kapitel 7. Wie effizient ist Ihr Algorithmus?

Aufgabe 3:
Gesucht ist ein PRO-Programm zur Komprimierung von Zahlenfolgen. Das Programm liest eine Zahlenfolge ein, deren Elemente jeweils eine der Ziffern 1,...,9 sind. Die Zahlenfolge wird so komprimiert und ausgegeben, daß Teilfolgen gleicher Ziffern unterdrückt werden, die ursprüngliche Folge aber wiederhergestellt werden kann. Dieses Problem sei wie folgt gelöst: Jede Teilfolge der Form

[...,x,x,x,...,x,x,x,x,...]
n-mal

wird auf zwei Folgenelemente komprimiert, wobei das erste die Anzahl der aufeinanderfolgenden x und das zweite x selbst ist.
Beispiel: Die Zahlenfolge

[1,1,5,5,5,4,4,4,4,3,7,7,7,8,3,3,3,3]

wird komprimiert zu

[2,1,3,5,4,4,1,3,3,7,1,8,4,3].

Schreiben Sie eine funktionale Spezifikation für das gesuchte PRO-Programm und das Programm selbst.