

**Übungsblatt 1, Besprechungstermin: Mittwoch, 12. Oktober 2022**

**Rahmenbedingung:** Die Übungsaufgaben (ab Übungsblatt 1) werden immer ca. ein Woche vor der Besprechung ausgegeben. In der darauffolgenden Übungsgruppe werden die Aufgaben besprochen. Dabei sollten die Aufgaben von Studierenden an der Tafel vorgeführt werden.

Jede Student\*in muss **mindestens einmal** eine (Teil)aufgabe an der Tafel vorführen.

## 1.1 Begriffsdefinitionen

Machen Sie sich folgende Begriff klar:

- Menge
- Relation
- Prädikat
- Tupel
- Menge von Mengen
- Potenzmenge
- Funktion

## 1.2 Addition, Multiplikation, Potenz

- a) Programmieren Sie die Additionsfunktion `add` aus dem Beispiel der der Vorlesung und testen Sie die Funktion für einige Wertepaare.
- b) Programmieren Sie analog die Multiplikationsfunktion `mul`, in dem Sie die Additionsfunktion benutzen.
- c) Programmieren Sie ebenfalls analog die Funktion `pot(x, y) = xy`, in dem Sie die Multiplikationsfunktion benutzen.

Geben Sie jeweils die Funktionen  $g$  und  $h$  des primitiven Rekursionsschemas an.

## 1.3 Addition, Multiplikation, Potenz als LOOP–Programme

Programmieren Sie die Programme aus Aufgabe 1.2 als LOOP–Programme.

**Hinweise:**

- a) Laden Sie die Datei `LOOP.h` von der Homepage down.

- b) Jede Datei eines LOOP-Programms muss mit `#include "LOOP.h"` beginnen.
- c) Nennen Sie ihre Datei `addmulhoch.c`
- d) Das LOOP-Programm muss mit dem Befehl `PSTART` beginnen und mit dem neuen Befehl `PEND` enden.
- e) Es stehen die Variablen `x[0]`, `x[1]`,  $\dots$ , `x[99]` zur Verfügung. Sie stehen für  $x_1, \dots, x_{99}$ .
- e) Die LOOP-Variable muss in Klammern stehen: z.B. `LOOP(x[18]) DO ... END`
- f) Kompilieren Sie Ihr Programm mit dem Aufruf
 

```
cc -o addmulhoch addmulhoch.c
```
- g) Die Eingabe-Werte werden beim Aufruf des kompilierten Programms durch Blank getrennt angegeben: `./addmulhoch arg1 arg2 ...`

## 1.4 Subtraktion

- a) Schreiben Sie die  $-1$ -Funktion als primitives Rekursionsschema
- b) Programmieren Sie die  $-1$ -Funktion als LOOP-Programm
- c) Schreiben Sie die  $\dot{-}$ -Funktion als primitives Rekursionsschema.
- d) Programmieren Sie die  $\dot{-}$ -Funktion als LOOP-Programm.  
**Hinweis:**  $x \dot{-} y = x - y$ , wenn  $x > y$  und sonst gilt  $x \dot{-} y = 0$ .

## 1.5 Einfache Funktionen

Seien folgende Funktionsdefinitionen gegeben:

- $h(y, x_1) = x_1 \dot{-} 2 \cdot y$
- $h(y, x_1) = x_1 \dot{-} 2^y$

1. Bestimmen Sie jeweils  $f(x_1) = (\mu y)[h(y, x_1) = 0]$
2. Geben Sie jeweils in Worten an welche Funktion  $f$  beschreibt.
3. Geben Sie die zugehörigen WHILE-Programme an. Verwenden Sie dazu die LOOP-Programm aus Aufgabe 1.4 und Aufgabe 1.3.

**Hinweise:**

- Laden Sie die Datei `WHILE.h` von der Homepage.
- Jede C-Datei eines LOOP-Programms muss mit `#include "WHILE.h"` beginnen (`LOOP.h` sollte dann nicht mehr included werden).