

## Übungen zu Einführung in die Informatik I

### Aufgabe 27 Terminierung

Für welche Eingabewerte terminieren die folgenden Funktionen? Beweisen Sie Ihre Aussagen!

- a) 

```
let rec fun f_a (m,n) =  
    if (n = 0) then 1  
    else f_a (m+1,n-1) + f_a (m+2,n-1);;
```
- b) 

```
let rec f_b (m,n) =  
    if (m = n) then 1  
    else f_b (m,n-1) + f_b (m+1,n);;
```
- c) 

```
let rec f_c (m,n) =  
    if (m = n) then 1  
    else f_c (m+1,n-1);;
```
- d) 

```
let rec f_d n =  
    if (n < 100) then f_d (200-n-1)  
    else if (n > 100) then f_d (200-n+1)  
    else n;;
```

### Aufgabe 28 Ordnungen über Zeichensequenzen

Gegeben sei ein Zeichenvorrat  $C$ . Ein Wort  $y$  heißt ein Präfix eines Wortes  $x \in C^*$ , wenn es ein  $u \in C^*$  gibt mit  $x = y \circ u$ . Wir sagen:  $y \sqsubseteq x$  genau dann, wenn  $y$  ein Präfix von  $x$  ist.

- a) Beweisen Sie, daß  $C^*$  bezüglich der Präfixrelation  $\sqsubseteq$  partiell geordnet ist.
- b) Zeigen Sie, dass die Präfixrelation  
keine totale Ordnung,  
eine noethersche (fundierte), jedoch keine wohlgeordnete Ordnung ist.
- c) Beweisen Sie, dass die lexikographische Ordnung über  $C^*$  nicht noethersch (fundiert) ist. Die lexikographische Ordnung ist dabei wie folgt definiert: Sei  $(C, \leq_C)$  ein total geordneter, endlicher Zeichenvorrat. Bezeichne  $\sqsubseteq$  die oben definierte Präfixordnung. Dann ist die lexikographische Ordnung  $\leq$  auf  $C$  definiert durch
  - (i)  $\varepsilon \leq x$  für alle  $x \in C^*$
  - (ii)  $x \leq y$  für  $x, y \in C^*$  und  $x \sqsubseteq y$
  - (iii)  $xay \leq xby'$  für  $x, y, y' \in C^*$ ,  $a, b \in C$  und  $a \neq b$  und  $a \leq_C b$ .
- d) Geben Sie eine fundierte, totale Ordnung über  $C^*$  an.

### Aufgabe 29      Vergleich von Bäumen

In dieser Aufgabe wenden wir uns nochmals sortierten binären Bäumen zu. Zwei sortierte Binärbäume können strukturell verschieden sein, obwohl sie die gleichen Elemente enthalten. Schreiben Sie eine Funktion `compare_tree` die zwei Binärbäume sowohl strukturell als auch inhaltlich vergleicht und `true` zurückliefert, falls beide Bäume identisch sind.

- a) Implementieren Sie die Funktion `compare_tree` indem Sie aus dem Baum zuerst eine Liste erstellen und diese dann überprüfen.
- b) (Optional) Implementieren Sie die Funktion `compare_tree` indem Sie rekursiv den Baum durchlaufen.

### Aufgabe 30 (H) Teilerrelation

(10 Punkte)

Es sei die Menge  $M = \mathbb{N} \setminus \{1\}$  gegeben und  $a, b \in M$ . Dann ist die Teiler-Relation  $|$  durch das Prädikat

$$a|b \Leftrightarrow \exists x \in \mathbb{N} : a \cdot x = b$$

gegeben.

- a) Zeigen Sie: Die Relation  $|$  definiert eine partielle, jedoch keine totale Ordnung.
- b) Geben Sie für  $b = 135$  zwei unterschiedliche absteigende Ketten an.
- c) Weshalb definiert die Teiler-Relation eine fundierte Ordnung, jedoch keine Wohlordnung?

### Aufgabe 31 (H) Terminierung

(5 Punkte)

Für welche Eingabewerte terminiert die folgende Funktion? Beweisen Sie Ihre Aussage!

```
let rec f (n,m,k) =  
    if (n=m) then k  
    else if (n<m) then f (2*m-n-1,m,k+1)  
    else f (2*m-n+1,m,k+1) ; ;
```