

Übungen zur Vorlesung Grundlagen der Programmierung II Blatt 4

Aufgabe 1:

Implementieren Sie einen abstype Queue auf folgende Weise: Die Queue wird durch ein Paar von Listen `entry` und `exit` dargestellt. `exit` verknüpft mit der gespiegelten Version von `entry` ist die gesamte Queue. Man fügt neue Elemente immer vorne in `entry` ein und löscht sie immer vorne in `exit`. Ist `exit` leer, so kopiert man die gespiegelte Version von `entry` nach `exit`.

Aufgabe 2:

Hier nochmal das prädikative Programm für das Mischen zweier Zahlenfolgen:

$$(\forall Y) (\text{misch}([], Y, Y)).$$

$$(\forall Y) (\text{misch}(Y, [], Y)).$$

$$(\forall A, B, X, Y, Z) (A \leq B \wedge \text{misch}(X, [B|Y], Z) \Rightarrow \text{misch}([A|X], [B|Y], [A|Z])).$$

$$(\forall A, B, X, Y, Z) (B < A \wedge \text{misch}([A|X], Y, Z) \Rightarrow \text{misch}([A|X], [B|Y], [B|Z])).$$

Man beweise die beiden Behauptungen

$$\text{misch}([1,7], [2,5], [1,2,5,7]) \text{ und}$$

$$(\exists Y) (\text{misch}([2,5], [1,2,8], Y)).$$

Aufgabe 3:

Definieren Sie den Datentyp `queue` aus Aufgabe 1 als `structure`.

Aufgabe 4:

Man schreibe einen Funktor von der Ergebnissignatur `order` mit zwei formalen Parametern `order1` und `order2` ebenfalls jeweils von der Signatur `order`, der die lexikographische Ordnung auf den Paaren bezgl. `order1` und `order2` realisiert.

Erläuterung: Sind $(M, <)$ und $(M', <<)$ zwei Ordnungen, so ist $(M \times M', \ll)$ ebenfalls eine Ordnung, wobei \ll (= lexikographische Ordnung) definiert ist durch:

$$(x, y) \ll (x', y') : \Leftrightarrow x < x' \text{ oder } (x = x' \text{ und } y << y').$$