Praktikum 3 zur TIWS

WS13/14

Gruppe 1: 21.11.2013

Gruppe 2: 28.11.2013

Ziele:
In diesem Versuch soll der Umgang mit den Datenstrukturen `Liste` und `Binärbaum` vertieft werden. Dabei werden Listen jetzt jedoch in der Prolog-Notation dargestellt:
- leere Liste: `[]`
- Liste mit Anfangselement `X` und Restliste `Xs`: `[X|Xs]`

Außerdem wird der Umgang mit der Arithmetik und Rekursion an einigen Funktionen und einem Sortierverfahren geübt.

Aufgabe 1: (Binärbäume und Listen)
Implementieren Sie die folgenden Prolog-Relationen für Binärbäume (siehe Übungsaufgabe 18):

- `präorder(Xb,Ys)` : `Ys` ist die Liste der Knotenbeschriftungen des Binärbaumes `Xb` in Präorder.
- `postorder(Xb,Ys)` : `Ys` ist die Liste der Knotenbeschriftungen des Binärbaumes `Xb` in Postorder.
- `tiefe(Xb,Ys)` : `Ys` ist die Liste der Knotenbeschriftungen des Binärbaumes `Xb`, die sich bei der Tiefensuche ergibt.
- `roots(Xbs,Ys)` : `Xbs` ist eine Liste von Binärbäumen (geschachtelte Induktion). Die Liste `Ys` ist die Liste der Wurzelbeschriftungen der Binärbäume in `Xbs` in der richtigen Reihenfolge. Beachten Sie, dass ein leerer Binärbaum keine Wurzelbeschriftung hat und diese somit auch nicht aufgeführt wird.

Sonderaufgabe (außer Konkurrenz):
- `breite(Xb,Ys)` : `Ys` ist die Liste der Knotenbeschriftungen des Binärbaumes `Xb`, die sich bei der Breitensuche ergibt.

Hinweis: Verwenden Sie ggf. eine zu der oben implementierten Relation `roots(Xbs,Ys)` ähnliche Relation.

Aufgabe 2: (Türme von Hanoi)
Bei dem Spiel "Türme von Hanoi" geht es darum, einen Turm mit n von unten nach oben immer kleiner werdenden Steinen von einem Stapel a zu einem Stapel b unter Verwendung eines Stapels c zu transportieren.
Dabei darf immer nur ein Stein gleichzeitig transportiert werden und es darf nie ein Größerer auf einen Kleineren gelegt werden.
Implementieren Sie eine Relation `tof(X,Ys)`, wobei `x` die Anzahl der Steine als natürliche Zahl in arithmetischer Darstellung enthält und `Ys` eine Liste der Züge, die zur Lösung des Problems erforderlich sind.

Aufgabe 3: (Quicksort)
Implementieren Sie den Quicksort-Algorithmus für natürliche Zahlen in arithmetischer Darstellung in Prolog.
Vergleichen Sie die Komplexität des Quicksort-Algorithmus mit der des Permutation Sort-Verfahrens aus der Vorlesung.