

Praktikum 2 zur TIWS**Gruppe 1: 14.11.2013****WS13/14****Gruppe 2: 07.11.2013****Ziel:**

In diesem Versuch werden die Definition und der Umgang mit induktiven Datenstrukturen in Prolog geübt.

Hinweis:

Auch diesmal werden während der Durchführung noch weitere Aufgaben gestellt.

Programmieren Sie möglichst nichtdeterministisch!

Aufgabe 1: (Listenstruktur und -operationen)

Listen seien mittels der Konstanten `nil` und dem 2-stelligen Funktor `list`, wie in der Vorlesung beschrieben, definiert.

Bsp.: `nil`, `list(a, list(b, nil))` sind zwei Beispiele für Listen.

- a) Definieren Sie ein Datentypprädikat **liste(Xs)**, das überprüft, ob es sich beim Argument um eine gültige Liste handelt, wobei die Listeneinträge beliebig sind. (vergleiche hierzu: Übungsaufgabe 10)
- b) Implementieren Sie die folgenden Prädikate, wobei jeweils mittels des in a) implementierten Prädikats überprüft wird, ob es sich um gültige Listen handelt:
 - **anz(Xs, N)** : N (nat. Zahl in symbolischer Darstellung) ist die Anzahl der Einträge von Xs.
 - **präfix(Xs, Ys)** : Ys beginnt mit der Liste Xs.
 - **infix(Xs, Ys)** : Ys enthält die Liste Xs.
 - **postfix(Xs, Ys)** : Ys endet mit der Liste Xs.
 - **member(X, Ys)** : Ys enthält das Element X.

Hinweis: Sie können Prädikate zur Definition anderer Prädikate verwenden. Insbesondere ist die Verwendung des Prädikats **app** aus der Übung zu empfehlen!

Aufgabe 2: (Binärbaumstruktur und -operationen)

Ein Binärbaum ist eine Datenstruktur, die leer ist oder bei der jeder Knoten einen Eintrag enthält und 2 Nachfolgerbäume hat.

- a) Überlegen Sie, wie man Binärbäume in Prolog darstellen kann, dabei kann man mit einer Konstanten und einem dreistelligen Funktor auskommen.
- b) Schreiben Sie, analog zu Aufgabe 1 a), ein Datentypprädikat **binbaum(X)**, das überprüft, ob es sich beim Argument um einen gültigen Binärbaum handelt. Auch dabei sind die Einträge der Knoten beliebig.
- c) Implementieren Sie die folgenden Prädikate, wobei jeweils mittels des in b) implementierten Prädikats überprüft wird, ob es sich um gültige Binärbäume handelt:
 - **root(Xb, Y)** : Y ist der Wurzeleintrag des Binärbaumes Xb.
 - **left(Xb, Yb)** : Yb ist der linke Teilbaum des Binärbaumes Xb.
 - **right(Xb, Yb)** : Yb ist der rechte Teilbaum des Binärbaumes Xb.
 - **construct(Root, Lb, Rb, Xneub)** : Xneub ist der Baum mit
 - Wurzelbeschriftung Root,
 - linkem Teilbaum Lb und
 - rechtem Teilbaum Rb.
 - **knotenanz(Xb, N)** : N (nat. Zahl in symbolischer Darstellung) ist die Anzahl der Knoten des Baumes Xb.