

Logik für Informatiker (Diplom)

Prof. Dr. B. Nebel, Prof. Dr. W. Burgard
Wintersemester 2007/2008

Universität Freiburg
Institut für Informatik

Präsenzübung 2

Aufgabe 2.1 (Größter gemeinsamer Teiler)

Der euklidische Algorithmus zur Berechnung des größten gemeinsamen Teilers zweier positiver ganzer Zahlen kann folgendermaßen beschrieben werden:

$$\text{ggt}(x, y) := \begin{cases} x, & \text{falls } x = y \\ \text{ggt}(x - y, y), & \text{falls } x > y \\ \text{ggt}(x, y - x), & \text{falls } x < y \end{cases}$$

Implementieren Sie den Algorithmus in Prolog.

Aufgabe 2.2 (Programmiermethodik I)

Ein ganzzahlig markierter Binärbaum ist entweder ein Blatt oder eine Verzweigung mit einem ganzzahligen Wert, die zu einem linken und einem rechten Teilbaum führt (wobei die Teilbäume wieder ganzzahlig markierte Binärbäume sind):

```
bintree(leaf).  
bintree(branch(Left,X,Right)) :-  
    integer(X),  
    bintree(Left),  
    bintree(Right).
```

Ein ganzzahlig markierter Binärbaum ist sortiert, falls für alle inneren Knoten der Form `branch(Left,X,Right)` gilt, dass alle Werte im linken Teilbaum `Left` kleiner sind als `X`, das wiederum kleiner ist als alle Werte im rechten Teilbaum `Right`.

Schreiben Sie ein Prolog-Programm `find(Value,Tree,Subtree)`, das einen sortierten ganzzahlig markierten Binärbaum `Tree` nach einem Knoten mit einem bestimmten Wert `Value` durchsucht und als Ausgabe ggf. den Teilbaum `Subtree` von `Tree` zurückgibt, dessen Wurzel den Wert `Value` hat. Gehen Sie bei der Konstruktion nach dem Schema aus der Vorlesung vor, also

- Geben Sie eine deklarative Spezifikation an.
- Geben Sie das Programmskelett an.
- Vervollständigen Sie die Regelrümpfe.
- Setzen Sie in die Ausgabeargumente ein.

Aufgabe 2.3 (Programmiermethodik II)

- (a) Schreiben Sie ein Prädikat `tree_ins(Elem,Tree,NewTree)`, das das Element `Elem` in den sortierten ganzzahlig markierten Binärbaum `Tree` an der richtigen Stelle einfügt und das Ergebnis als `NewTree` zurückliefert.

Führen Sie die folgenden Anfragen durch:

- (a) `tree_ins(4,leaf,T1), tree_ins(5,T1,T2), tree_ins(2,T2,T3), tree_ins(7,T3,T4).`
- (b) `tree_ins(7,leaf,T1), tree_ins(5,T1,T2), tree_ins(4,T2,T3), tree_ins(2,T3,T4).`
- (c) `tree_ins(2,leaf,T1), tree_ins(4,T1,T2), tree_ins(5,T2,T3), tree_ins(7,T3,T4).`

Was fällt Ihnen auf?

- (b) Schreiben Sie ein Prädikat `print_tree(Tree)`, das für jedes Element in dem Baum `write/1` aufruft, so dass alle Elemente von links nach rechts ausgegeben werden.
- (c) Schreiben Sie ein Prädikat `sum_tree(Tree,Sum)`, das die Summe der Elemente des Baumes berechnet.
- (d) Schreiben Sie ein Programm, das die Höhe eines Baumes, also die maximale Länge eines Pfades von der Wurzel zu einem Blatt, berechnet.