

# Übung 7 zu TILO

SoSe20

Fragen bis 26.04.20

## Aufgabe 24: (Prolog-Beweisbaum)

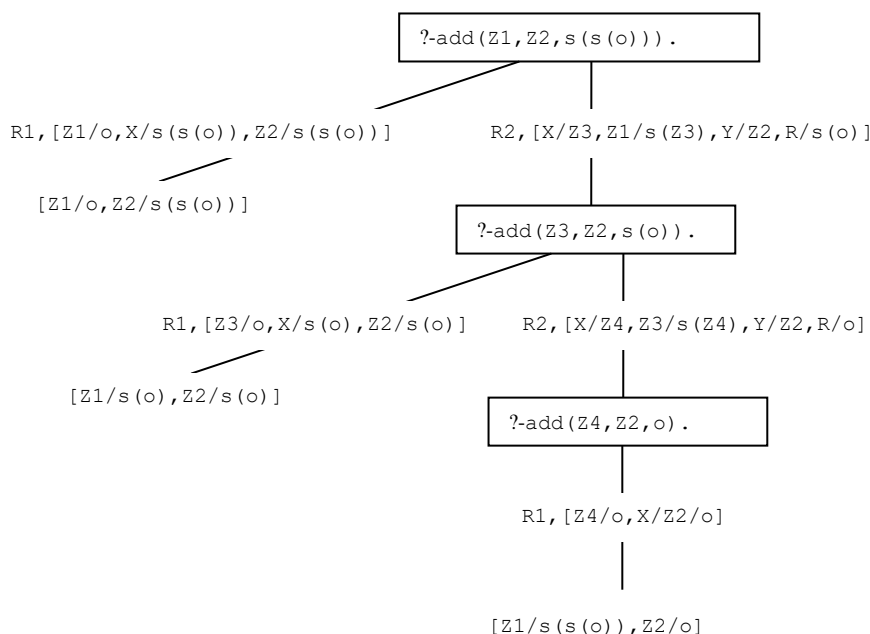
Geben Sie ein Beispiel für einen endlichen Prolog-Beweisbaum, dessen Höhe größer 1 ist, mit mehreren Variablen in der Query an. Dabei sollen zur Ermittlung der Antwortsustitutionen die Substitutionen entlang der einzelnen Äste sukzessive in mehreren Schritten von oben nach unten berechnet werden.

### Lösung:

```
add(o, X, X) .
```

```
add(s(X), Y, s(R)) :- add(X, Y, R) .
```

```
?- add(Z1, Z2, s(s(o))) .
```



## Aufgabe 25: (Durchläufe in Binärbäumen)

Binärbäume sind gemäß Aufgabe 21 dargestellt.

Implementieren Sie folgende Prolog-Relation:

- **tiefe(Xb, Ys)** : Ys ist die Liste der Knotenbeschriftungen des Binärbaumes Xb, die man bei einem depth-first left-to-right Durchlauf erhält.

### Lösung:

```
tiefe(empty, []).
```

```

tiefe(node(X, Lb, Rb), [X|Xs]) :-
    tiefe(Lb, LLs),
    tiefe(Rb, LRs),
    append(LLs, LRs, Xs) .

```

## Aufgabe 27: (induktive Operationen)

Implementieren Sie die folgenden Prädikate in Prolog, deren

- erstes Argument jeweils ein Baum aus der Datentyprelation `tree` aus Aufgabe 26 ist
- zweites Argument eine nat. Zahl in symbolischer Darstellung ist.

Verwenden Sie hierzu die abstrakten Definitionen aus den vorherigen Übungen und implementieren Sie ggf. noch Hilfsprädikate.

## Übung 7 zu TILO

## SoSe20

Fragen bis 26.04.20

- **height(T,N)** : **N** (nat. Zahl in symbolischer Darstellung) ist die Höhe des Baumes (siehe Aufgabe 7.c) **T**.
- **functs(T,N)** : **N** (nat. Zahl in symbolischer Darstellung) ist die Anzahl der Funktoren des Baumes (siehe Aufgabe 11.a) **T**.
- **varcons(T,N)** : **N** (nat. Zahl in symbolischer Darstellung) ist die Anzahl der Variablen und Konstanten des Baumes (siehe Aufgabe 11.b) **T**.

### Lösung:

```
%max(X,Y,M): M ist das Maximum von X und Y.
```

```
max(X,Y,X) :- lessequ(Y,X).
```

```
max(X,Y,Y) :- lessequ(X,Y).
```

```
%lessequ(X,Y) : X ist kleiner gleich Y.
```

```
lessequ(o,X).
```

```
lessequ(s(X),s(Y)) :- lessequ(X,Y).
```

```
%height(T,N) : N ist die Höhe von T.
```

```
height(T,o) :- const(T).
```

```
height(T,o) :- variable(T).
```

```
height(f(T1,T2),s(N)) :- height(T1,N1), height(T2,N2), max(N1,N2,N).
```

```
height(g(T),s(N)) :- height(T,N).
```

```
height(h(T1,T2,T3),s(N)) :- height(T1,N1), height(T2,N2), height(T3,N3),
                             max(N1,N2,H), max(H,N3,N).
```

```
add(o,X,X).
```

```
add(s(X),Y,s(Z)) :- add(X,Y,Z).
```

```
%functs(T,N) : N ist die Anzahl der Funktoren in T.
```

```
functs(T,o) :- const(T).
```

```
functs(T,o) :- variable(T).
```

```
functs(f(T1,T2),s(N)) :- functs(T1,N1), functs(T2,N2), add(N1,N2,N).
```

```
functs(g(T),s(N)) :- functs(T,N).
```

```
functs(h(T1,T2,T3),s(N)) :- functs(T1,N1), functs(T2,N2), functs(T3,N3),
                             add(N1,N2,H), add(H,N3,N).
```

```
%varcons(T,N) : N ist die Anzahl der Variablen und Konstanten in T.
```

```
varcons(T,s(o)) :- const(T).
```

```
varcons(T,s(o)) :- variable(T).
```

```
varcons(f(T1,T2),N) :- varcons(T1,N1), varcons(T2,N2), add(N1,N2,N).
```

```
varcons(g(T),N) :- varcons(T,N).
```

```
varcons(h(T1,T2,T3),N) :- varcons(T1,N1), varcons(T2,N2), varcons(T3,N3),
                             add(N1,N2,H), add(H,N3,N).
```

### Aufgabe 29: (Arithmetik)

Ändern Sie die Lösung aus Aufgabe 10.b) so ab, dass die natürlichen Zahlen nicht in symbolischer Darstellung, sondern in numerischer Darstellung angegeben werden.

Listen sind dabei in Prolog-Notation zu definieren.

### Lösung:

```
invList1([]).
```

```
invList1([0]).
```

```
invList1([Y,X|Xs]) :- invList1([X,Xs]), Y is X+1.
```

**Übung 7 zu TILO**

SoSe20

Fragen bis 26.04.20

**Aufgabe 30: (Arithmetik)**

- a) Implementieren Sie das Prädikat `listlength(Xs,N)`, so dass der zweite Parameter `N` die Länge der Liste `Xs` in numerischer Darstellung liefert.
- b) Implementieren Sie das Prädikat `anz(X,Xs,N)`, so dass der dritte Parameter die Häufigkeit des Auftretens des Elements `X` in der Liste `Xs` in numerischer Darstellung enthält.

Listen sind dabei in Prolog-Notation zu definieren.

**Lösung:**

```
listlength([],0).
listlength([X|Xs],NN) :- listlength(Xs,N),NN is N+1.
```

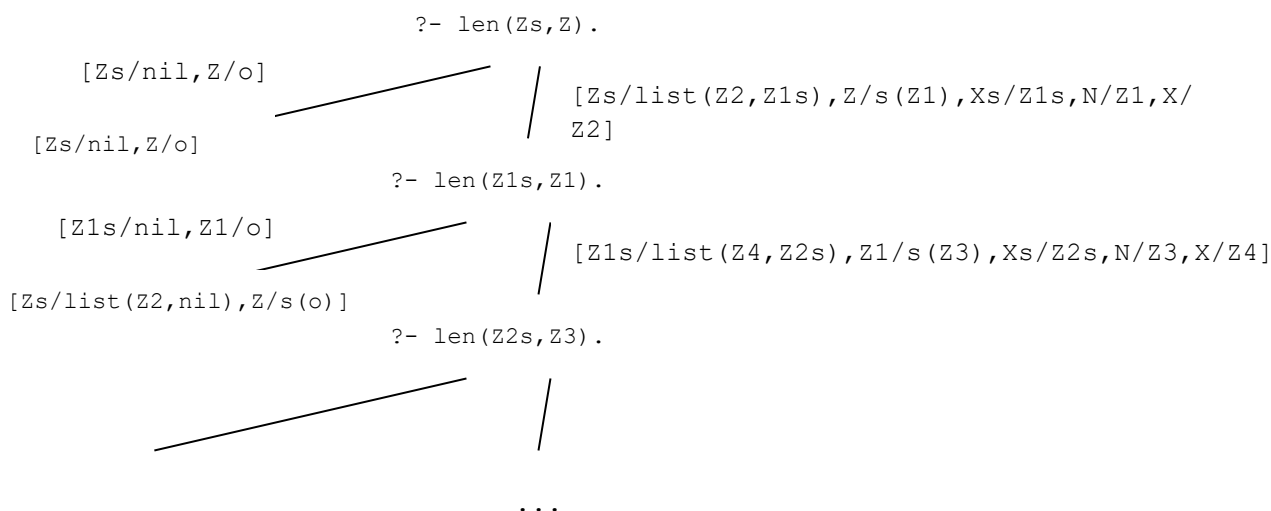
```
anz(X,[],0).
anz(X,[X|Xs],NN) :- anz(X,Xs,N), NN is N+1.
anz(X,[Y|Xs],N) :- X\==Y, anz(X,Xs,N).
```

**Aufgabe 31: (Prolog-Beweisbaum)**

Gegeben sei folgendes Prolog-Programm:

```
len(nil,o).
len(list(X,Xs),s(N)) :- len(Xs,N).
```

Zeichnen Sie den Prolog-Beweisbaum für obiges Programm und folgender Query `?- len(Zs,Z)`. Führen Sie dabei mindestens die beiden am weitesten links stehenden Äste und den am weitesten rechts stehenden Ast auf. Beim am weitesten rechts stehenden Ast sind mindestens die ersten drei Knoten aufzuführen.

**Lösung:**

**Für Donnerstag, 30.4., wird keine Übung hochgeladen!!!**

**Die Übung 8 wird am Montag, 4.5., besprochen.**

**Fragen können bis zum 3.5. gestellt werden.**