

Zentralübung: 22. 1. 03

(1)

Listen in Prolog:

Beispiel 1: $[a, b, c]$ ist Liste in Prolog

$[]$: Leere Liste. $[H|T]$ ist eine Liste, die aus einem Kopf H und dem Rest T aufgebaut.

Beispiel 2:

$p([H|T], H, T).$

Abfrage: $?- p([a, b, c], H, T).$

$H = a$

$T = [b, c]$

yes

Rekursive Rechenvorschriften zum Bearbeiten von Listen in Prolog:

① Suche in Listen nach einem best. Element:

$is\ Element(Element, [Element|Rest]).$

$is\ Element(Element, [_|Rest]) :-$

$is\ Element(Element, Rest).$

$? is\ E(a, [x, y, a])$

$\vdash is\ E(a, [y, a])$

$\vdash is\ E(a, [a]).$ [a|[]]

\checkmark

② Konkatenieren von Listen:

concat ([], L, L).

concat ([X|L₁], L₂, [X|L₃]) :-

concat (L₁, L₂, L₃).

Aufruf:

?-concat ([a, b], [c, d], R)

=> R = [a | R']

concat ([b], [c, d], R')

=> R' = [b | R'']

concat ([], [c, d], R'')

=> R'' = [c, d]

-> R' = [b, c, d]

-> R = [a, b, c, d];

③ Extrahieren der j. Teilliste aus einer List ganzer Zahlen, für die alle El. < 7 sind

less7 ([], [])

less7 ([X|L], [X|Res]) :- X < 7,

less7 (L, Res).

less7 ([X|L], Res) :- less7 (L, Res).

Hinweis:

?- trace. (notrace schaltet den Debugger aus)

Rechnen mit ("echten") natürlichen Zahlen

Wichtiges Prädikat in Prolog: "is"

Syntax: is (<Var>, <Expr>).

(Alternativ <Var> is <Expr>)

Beispiel 1: is (X, 3*5) weist der Variabl. X den Wert 15 zu.

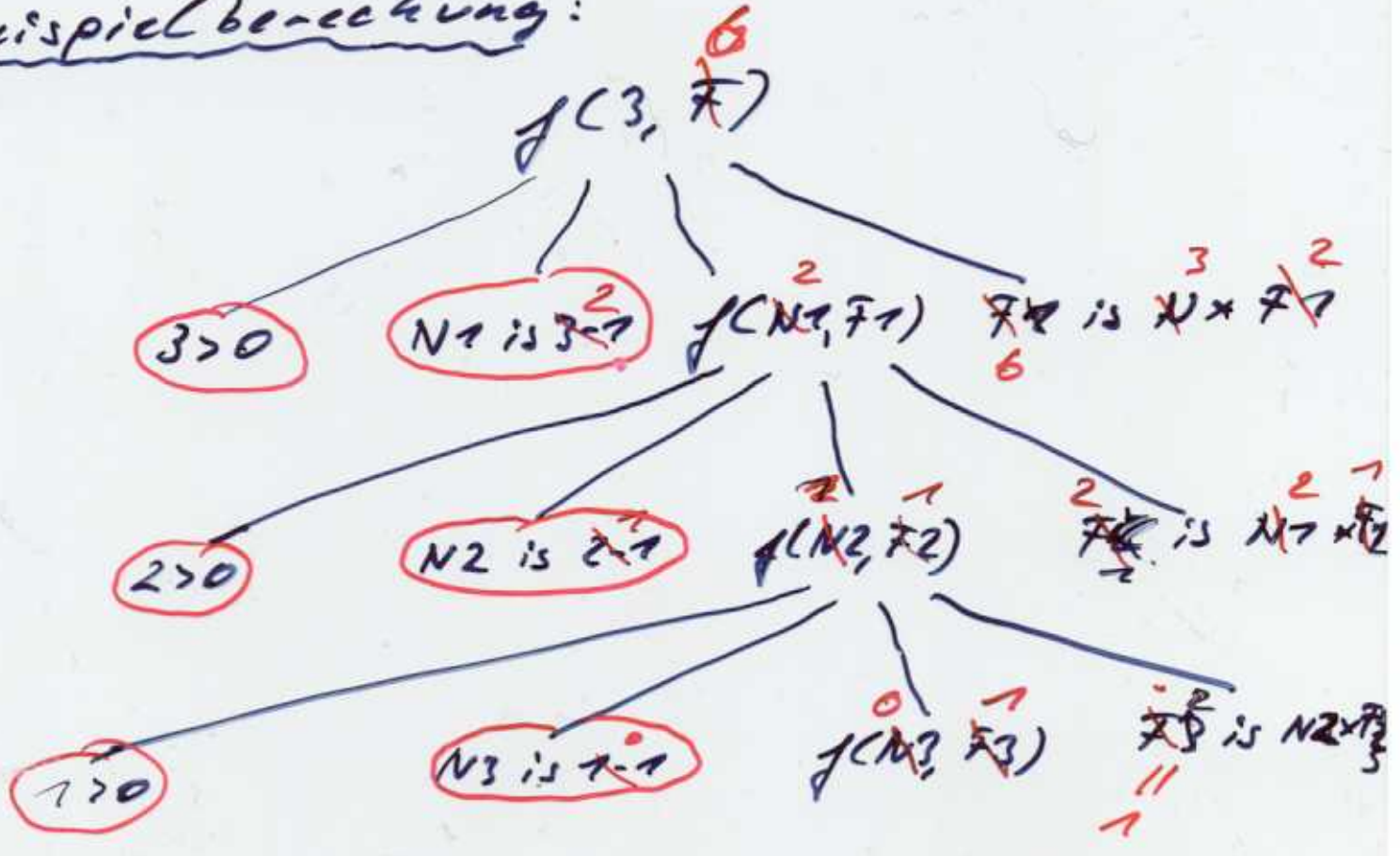
Beispiel 2: Fakultätsfunktion

f(0, 1)

f(N, F) :- N > 0, N1 is N-1,

f(N1, F1), F is N * F1

Beispielberechnung:



Bemerkung: Das "is"-Prädikat durchbricht die "normale" Vorgehensweise von Prolog.

Normalerweise wäre folg. Aufruf möglich $z-f(X, Y)$.

↗
Berechnet ohne Verwendung von "is" die gesamte Wertetabelle

Verbesserung: Verwendung von succ-Zahlen

$$f2(0, succ(0)).$$

$$f2(succ(N), F) :- f2(N, F2), \\ mult(succ(N), F2, F).$$

Multiplikationsfunktion: → geeignet zu definieren.