

JP typy (T) := v době vyhodnocování je T daného typu.

- val ...
- nonval ... ^{proměnná} _{ground}
- number, integer, float,
- atom
- compound
- atomic :=

is_list($+L$) := L je seznam seznamů

is_list(X) := nonval(X), $X = []$. $X = []$.

is_list(X) := is_list(B), $X = []$.

functor(T, N, A) := T etm je N/A .

arg(K, T, H) := n -tý parametr T je H .

Pomocí $T = \dots [N | P]$:= T má n řev N , parametry P .

functor(T, N, A) := $T = \dots [N | L]$, length(L, A).

arg(K, T, H) := $T = \dots [- | L]$, nth(K, L, H).

Meta-programování.

pred(A) := P .

call($: P, \dots$) := $P(\dots)$ pro každý prvek ($: P, L$).

maplist(P, L_1, L_2, \dots) := call(P, P_1, P_2, \dots), maplist(P, Z_1, Z_2, \dots).

($- | []$).

($P, [X | Z]$) := call(P, X), ... (P, Z).

setizeny ($: Najycu + L$) := t sousední $A, B \in L$ splnají Nejycse.

setizeny ($- | []$).

setizeny ($- | []$).

setizeny ($N, [A, B | L]$) := call(N, A, B), setizeny ($N, [B | L]$).

Akumulátor... (P, L, PH, V) :=

Aku($[]$, A, A).

Aku($[X | Z]$, A, V) := n zco, Aku($Z, f(A, V)$).

pred(P, A, NA)

call(pred)

NA

Stromy nil, $t(L, X, P)$.

strom(t) := t je strom.

strom(nil).

strom($t(L, -, P)$) := strom(L), strom(P).

postorder(t , X).

pre ($\left. \begin{array}{l} \text{postorder}(t(L, -, -)) \text{ := postorder}(L, X) \\ \text{postorder}(t(-, -, P), X) \text{ := postorder}(P, X) \\ \text{postorder}(t(-, X, -), X) \end{array} \right\}$ in order.



Vyhodnot(T, V) := V je vyhodnocení T .

nil .. čísla vnitřní uzoly... $X \in \{+, *\}$.

$t(5, +, t(2, *, 3))$

vyhodnot(H, H) := number(H).

$t(L_1, +, R), V$:= v yh(L_1, V), v yh(R, Rv), V is $L_1 + Rv$.

preorder_list(T, L).

preorder_list(nil, $[]$).

preorder_list($t(L, X, R), [X | Z]$) :=

preorder_list(L, PL);

(P, PR);

append(PL, PR, Z).