

Prolog: fakta, pravidla, dotazy, aritmetika, způsob vyhodnocení dotazů, unifikace, řízení výpočtu, operátor řezu

- postavený na first-order predicate logice
- program je postaven z Hornových klausulí (axiomy)
  - facts, rules, queries

fakta: male (josef).  
parent (josef, jan).

pravidla: mother (X,Y) :- parent (X,Y), female (X).  
predecessor (X,Y) :- parent (X,Y).  
predecessor (X,Y) :- parent (X,Z), predecessor (Z,Y).  
→ rekurzivní vyjádření vztahu

dotazy: ?- parent (josef, sarka). → yes  
?- predecessor (X, sarka). → X = josef X = jana .... no

všechno jsou termíny: atomy, čísla, proměnné, struktury (functor a list argumentů)  
- atomy vždy lower case, proměnné first uppercase

- Prolog většinou funguje DFS, protože k tomu může využít stack (může ale i BFS)  
→ prohlídkou se dá kontrolovat operátorem řezu

Unifikace - konstanta pouze se sebou  
- proměnná s číselním, ale rovněž ekvivalencí  
- struktury pokud je functor stejný (včetně počtu argumentů) & argumenty se shodují

- na počátku argumentů řádku → lepší by měly být proměnné  
- špatné počáteční může způsobit nekonečnou rekurzi

path (X,Y) :- path (X,Z), edge (Z,Y)

Arithmetika - přirození pomoci is

, - konjunkce ; - disjunkce

fact (0, 1).

fact (N, F) :- N > 0, N1 is N - 1, fact (N1, F1), F is N \* F1

Lists - [] - prázdný list

[1, 2, 3] - prázdný list

| - separator prvního prvků a zbytku listu

[a | [b, c]]

member (Elem, [Elem | Rest]).

member (Elem, [X | Rest]) :- member (Elem, Rest)

append ([], F, F)

append ([A | L1], L2, [A | Rest]) :- append (L1, L2, Rest)

+ X, - Y, ~~??~~ ?Z - vstupní, výstupní a obj. argument

Operator řízení !

- vždy úspěšně, nepovoluje backtracking

- kamery vyřadí ostatních pravidel

marada (jane, X) :- plesaty (X), !, fail

marada (jane, X) :- muz (X)

built-in predicates (some)

true, fail, not (Q)

integer (X), var (X), nonvar (X)

Factorial example :

factorial (0, 1).

factorial (N, F) :- N > 0,

N1 is N - 1,

factorial (N1, F1),

F is N \* F1.