

Neprocedurální programování 1

Vojtěch Šára

March 9, 2021

Program v prologu se skládá z klauzulí. Funktory + atomy = termy, s tečkou na konci z toho je klauzule.

```
muz(adam).  
muz(kain).  
muz(abel).  
zena(eva).  
rodic(adam,kain).  
rodic(adam,abel).  
rodic(eva,abel).  
rodic(eva,kain).
```

Toto definuje například predikát muz() - ten je definovaný právě na těch prvcích, které vyjmenujeme. Pak se můžeme ptát na dotazy a zkompileovaný program prologu nám odpoví true nebo false. Nebo můžeme zadat dotaz "muz(X)" což nám vypíše všechny hodnoty, pro které platí, že jsou muž.

Více dotazů můžeme spojit čárkou (= konjunkcí).

Chceme přidat predikát otec. To uděláme pomocí pravidla:

```
otec(Kdo, Dite) :- rodic(Kdo, Dite), muz(Kdo)
```

Toto se při dotazu "Otec(adam,kain)" vyhodnocuje rekurzivně - deklarovali jsme cíl, ten se rozdělí na podcíle, splnění všech podcílů indukce úspěch - splnění cíle.

Procedura s více pravidly

clovek(C) :- zena(C). clovek(C) :- muz(C). ! klauzule jsou procházeny v pořadí, ve kterém jsou napsány. To, co jsme napsali, je skrytá disjunkce. Jinak disjunkci mohu napsat pomocí středníku.

bratr(Bratr,Osoba):- rodic(R,Bratr),rodic(R,Osoba),muz(Bratr). Nedefinuje bratra korektně! Potřebujeme, aby Bratr a Osoba byly různé. Na to potřebujeme negaci, ta se píše backslash =.

Anonymní proměnná - _ na názvu této proměnné nezáleží - nechci jí použít nikdy znovu.

term - (jednoduchý - (konstanta - atom / číslo / []) / proměnná) / (složený)
složený - muz(adam) = funktor s jednoduchým nebo složeným termem, rekurzivní definice

Program se skládá z procedur, procedura je posloupnost klauzulí.

Fakt = pravidlo s prázdným tělem.

Direktiva = pravidlo s prázdnou hlavou, např :- consult(demo). toto je command pro překladač.

Procedurální versus neprocedurální

Proc.: proměnné - deklarovány, lokální / globální, změna hodnoty přiřazením

Neproc.: proměnné - dynamická alokace paměti - GC, platnost proměnné je omezena na klauzuli, v níž se vyskytuje. Proměnná volná / vázaná.

Proc.: datový typ záznam - položky identifikovány jménem.

Neproc.: složený term - položky identifikovány polohou, stromová struktura.

Řešení Einsteinovy hádanky

Strategie - použijeme složené termy.

kolej(→,pokoj(→,→,skyrim,→),pokoj(gates,→,→,→),pokoj(→,brno,→,→)) - toto je dobrý první krok, další neuvádíme.

Syntax - Operátory

Proc.: výrazy - $a+b$

Neproc.: adam ēva

Unifikace - dva terma lze unifikovat, pokud jsou identické, nebo se stanou identickými po substituci vhodné hodnoty proměnným v obou termech.

Unifikační algoritmus - ten je vyvolán $=$. Algoritmus se pokusí najít substituci, která unifikuje.

Peano aritmetika v prologu

prirozene_cislo(X) :- X je přirozené číslo.

prirozene_cislo(0).

prirozene_cislo(s(X)) - s je sucesor

mensi(X,Y):- X je ostře menší než Y

mensi(0,s(X)) :- prirozene_cislo(X).

mensi(s(X),s(Y)) :- mensi(X,Y).

Strukturální rekurze

Řízena strukturou argumentů - rekurzivní datová struktura. Složená z báze a kroku rekurze, realizovaným rekurzivním pravidlem.

soucet(0,X,X) :- prirozene_cislo(X).

soucet(s(X),Y,s(Z)) :- soucet(X,Y,Z).

Seznamy

Seznam realizovaný pomocí relace. Zápis [a,b,c].