

Datalog

Datalog

- Podrobnejšie teoretické základy na prednáškach
- Program v datalogu je množina pravidiel (implikácií), napr.
 - $\text{zlozene_cislo}(Z) \leftarrow \text{krat}(X, Y, Z), \text{int}(X), \text{int}(Y), \text{not } X = 1, \text{not } Y = 1.$
 - $\text{prvocislo}(Z) \leftarrow \text{int}(Z), \text{not } Z = 1, \text{not } \text{zlozene_cislo}(Z).$

- Syntax:

`<hlava>: <atom>`

`<hlava> :- <telo>`

`<telo>: <atom> | \+ <atom> | <telo>, <atom>`

Na vyhodnocovanie datalogových dotazov budeme používať Prolog, preto na ich zápis využijeme syntax Prologu (ktorý je nadmnožinou datalogu):

- `\+` je negácia, `:-` je “implikácia”

Datalog

- príklad datalogovského pravidla:
 - $\text{res}(N,J) \text{ :- emp}(_,N,J,_,_,S,_), S \geq 2000.$
- na ľavej strane vždy len jeden pozitívny atóm
- premenné začínajú veľkým písmenom
- konštanty malými písmenami
- každá premenná sa v tele pravidla uvedená aspoň v jednom *pozitívnom EDB kontexte*
- `_` znamená anonymnú premennú
- na vyhodnocovanie aritmetických výrazov slúži operátor *is*:
 - napr. $X \text{ is } 2+3$, nie $X = 2+3$
 - (symbol = by bol interpretovaný ako unifikácia termov a nedôjde k žiadnej aritmetickej operácii).

Práca s datalogom: SWI-Prolog

- tri možnosti:
 - na serveri *cvika*, pripojiť sa cez ssh na `cvika.dcs.fmph.uniba.sk`
(prihlasovacie meno / heslo ako v AISe)
 - v Linuxe v M217, alebo na vlastnom počítači, kde nainštalujete SWI-Prolog
 - online na <https://swish.swi-prolog.org/>
- odporúčame otvoriť si 3 okná
 - v jednom editujete súbor s dotazmi, napr. **vim queries_emp.pl**
 - v druhom okne máte spustené prostredie prologu: **swipl -s queries_emp.pl**
 - v treťom okne máte databázu (zoznam faktov)

Práca s datalogom

- po zapísaní dotazu do súboru ho treba uložiť na disk (vim: ESC, ":w", ENTER).
- potom novú verziu skompilovať: **make**. (aj s tou bodkou)
 - nezabudnite skontrolovať, či kompilátor hlási chyby a prípadne ich opraviť
- výpočet dotazov:
?- **q(job(J))**.
- predikát "q(_)" slúži na pekné formátovanie výstupu a elimináciu zdanlivých duplikátov (Prolog robí úplný backtracking a konkrétnu hodnotu môže nájsť vo viacerých vetvách)

Datalog a negácia

- Dvojice [D, J] také, že pozícia J nie je v oddelení D:

```
jobDept(D, J) :- emp(,,J,.,.,D).
```

```
nemaJob(D) :-
```

```
    emp(,,J,.,.,),
```

```
    emp(.,.,.,.,D),
```

```
    \+ jobDept(J,D).
```

- Prečo nestačí napísať toto?

```
nemaJob(D, J) :-
```

```
    emp(,,J,.,.,),
```

```
    emp(.,.,.,.,D),
```

```
    \+ emp(,,J,.,.,D).
```

Datalog a všeobecný kvantifikátor

- je nutné *prepísať* všeobecný kvantifikátor ako negáciu existenčného, čiže v pomocnom pravidle popíšeme protipríklad a negovaním pomocného predikátu povieme, že protipríklad neexistuje
- oddelenia, v ktorých je zastúpený každý druh práce:

hasAllJobs(D) :- dept(D,_,_), \+ missingJob(D).

missingJob(D) :- emp(_,_,J,_,_,_), emp(_____,D), \+ jobDept(J,D).

jobDept(J, D) :- emp(_____,J,_____,D).