

Intelligentné rozhodovacie systémy



Prolog

Marian.Mach@tuke.sk

<http://people.tuke.sk/marian.mach>

December, 2018

Hornova logika

Najviac jeden pozitívny literál:

■ Pravidlá (poz¹ + neg⁺)

$\neg \text{luhar}(Kto) \vee \neg \text{povedal}(Kto, Co) \vee \text{loz}(Co)$ $\neg \text{klame}(kto) \vee \text{luhar}(Kto)$
 $\text{luhar}(Kto) \wedge \text{povedal}(Kto, Co) \rightarrow \text{loz}(Co)$ $\text{klame}(Kto) \rightarrow \text{luhar}(Co)$

■ Fakty (poz¹)

$\text{povedal}(\text{jano}, \text{prsi})$ $\text{luhar}(\text{fero})$ $\text{klame}(\text{jano})$
 $\top \rightarrow \text{povedal}(\text{jano}, \text{prsi})$ $\top \rightarrow \text{luhar}(\text{fero})$ $\top \rightarrow \text{klame}(\text{jano})$

■ Ciele (neg⁺)

$\neg \text{loz}(\text{prsi})$ $\text{loz}(\text{prsi}) \rightarrow \perp$

Odvodzovanie (hypotéza: $\text{loz}(\text{prsi})$)

Rezolvencia

Prehľadávanie do hĺbky

Jazyky

Prolog

- logické (deklaratívne) programovanie
- spätné reťazenie pravidiel (dokazovanie cieľov)
- prehl'adávanie do hĺbky
- zabudovaný odvodzovací mechanizmus

Nasledovníci

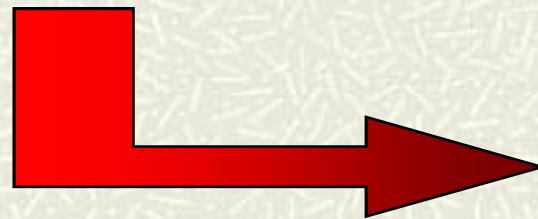
- CLP (+ spĺňanie ohraničení) - Eclipse
- Objekty – Logtalk, Prolog++
- Funkcionálnosť (zakrytie inferencie) - Mercury
- Pravdepodobnosť - Problog

Prolog

- # 1972 - oficiálny dátum vzniku
- # Robert Kowalski (University of Edinburgh)
 - teoretický rámec na ktorom je Prolog postavený (procedurálna interpretácia logiky Hornových klauzulí)
- # Alain Colmerauer (University of Aix-Marseille)
 - so svojou skupinou vytvorili theorem prover ktorý bol použitý pre implementáciu komunikačného systému v prirodzenom jazyku – tento systém nazvali PROLOG (PROgrammation et LOGique - PROgramming in LOGic).

Programmation et Logique

**Every psychiatrist is a person.
Every person he analyzes is sick.
Jacques is a psychiatrist in Marseille.
Is Jacques a person?
Where is Jacques?
Is Jacques sick?**



**Yes.
In Marseille.
I don't know.**

Vývoj Prologu

Dva z hlavných dialektov pochádzajú zo zakladateľských univerzít:

- Marseille: prvý prologovský interpreter (Roussel)
 - bežal na počítači IBM 360-67 v Grenobli, Marseille mala teletypové pripojenie na ten počítač
- Edinburg: prvý prologovský kompilátor (D. Warren)
 - mnoho ďalších riešení bolo tiež postavených na WAM

Prolog bol dlho ignorovaný, až kým sa nestal súčasťou japonského ‘Fifth Generation Project’

ISO Prolog štandardizácia: ISO/IEC 13211-1 bol publikovaný v 1995

Prolog na KKUI

doc. Ing. Július Csontó, CSc.

KKUI - jedno z prvých centier Prologu v ČSSR

- aktívne obdobie: od 1986 až do pol. 90-tych rokov

- Publikácie

- 1986 - prvé skriptá v ČSSR (Csontó - Slota)

- 1988, 1989 v JZD Agrokombinát Slušovice

- Výuka: Aplikácia jazyka Prolog v UI

- Výskum: aplikačné použitie Prologu

- Projekty:

- Prolex - prázdny expertný systém (MVS-CP/M, XT)

- Kravex - diagnostikovanie chorôb hov. dobytku (XT)

Prolog (1)

Interaktívny režim

- kladenie otázok, ktoré sa systém snaží zodpovedať na základe svojej databázy
- otázka je považovaná za výrok
 - ktorý sa systém snaží dokázať
 - systém stanoví podmienky za ktorých je výrok platný
- systém obsahuje zabudovaný odvodzovací mechanizmus
 - stačí povedať čo sa má robiť, nie je potrebné definovať ako sa to má urobiť
 - možno ho využiť vo väčšej alebo menšej miere

Prolog (2)

Objekty jazyka (termy):

- konštanty
 - čísla
 - atómy (napr. textové konštanty začínajúce malým písmenom)
- premenné
 - `Januar` (veľké písmeno), `_` (špeciálna anonymná premenná)
- štruktúry
 - funktor a argumenty (árnosť = počet argumentov)
 - `ma(jano, rok(20))` - klasický zápis
 - zoznam (špeciálny zápis - `[jano, fero, miso]`)

Prolog (3)

Databáza obsahuje klauzuly (fakty a pravidlá):

- fakty

- sú to štruktúry (ľubovoľnej úrovni)
- definujú to čo platí

```
osoba(meno,priezvisko).  
pocasio([vietor, hmla]).  
teplota(Chladno).  
prsi.
```

Model ‘uzavretého sveta’ - čo nie je v databáze, alebo sa z nej nedá odvodiť, to neplatí

Prolog (4)

Databáza obsahuje klauzuly (fakty a pravidlá):

- pravidlá

- binárne štruktúry tvaru: `HLAVA :- TELO`
- definujú vzťahy medzi cieľmi (rozklad cieľa na podciele)

```
pocasio(zle) :- dazd(husty); teplota(chladno).  
meno([jano| OSTATNI]) :- meno(OSTATNI).  
sirota(Kto) :- not(matka(Matka, Kto)),  
not(otec(_, Kto)).
```

Premenné sú lokálne, ich platnosť je iba v rámci jednej klauzuly

Prolog (5)

Unifikácia (zhodnosť)

- neviazaná premenná sa môže unifikovať na ľubovoľný objekt, naviaže sa naň
- unifikáciou dvoch neviazaných premenných vzniknú zdieľané premenné
 - ak jedna získa hodnotu, tak ju bude mať aj druhá
- štruktúry sú unifikovateľné, ak majú rovnaký funktor a počet argumentov, a jednotlivé argumenty sú unifikovateľné

Viazaná premenná **nemôže** zmeniť svoju hodnotu

Prolog (6)

Dokazovanie (prehľadávanie)

- pokus o splnenie cieľa
 - DB sa prehľadáva zhora nadol, začína sa **od začiatku**
- pokus o opätovné splnenie (hľadanie alternatívy)
 - DB sa prehľadáva zhora nadol, začína sa **od posledne nájdenej možnosti**
- v prípade neúspešného pokusu sa systém vracia k posledne úspešnému splneniu a pokúša sa o novú alternatívu
 - zrušia sa viazania premenných, ktoré prebehli pri práve rušenej unifikácii

Prolog (7)

```
pocasio(zle) :-  
    dazd(husty);  
    teplota(chladno).  
pocasio(pekne) :-  
    slnko(svieti),  
    teplota(teplo).  
pocasio(zle) :-  
    slnko(nesvieti).  
pocasio(zle) :-  
    not(pocasio(pekne)).  
pocasio(zle).  
slnko(svieti).  
teplota(chladno).  
dazd(ziadny).
```

```
?- pocasio(zle).
```

```
yes
```

```
?- pocasio(pekne).
```

```
no
```

```
?- pocasio(X).
```

```
X = zle
```

```
yes
```

```
?- pocasio(X).
```

```
X = zle ;
```

```
X = zle ;
```

```
X = zle ;
```

```
no
```

