

Paradygmaty programowania

Programowanie generyczne w C++

Dr inż. Andrzej Grosser

Częstochowa, 2014

Spis treści

1. Zadanie 10	5
1.1. Programowanie w logice	5
1.1.1. Fakty	5
1.1.2. Koniunkcje	5
1.1.3. Zapytania	5
1.1.4. Reguły	6
1.1.5. Interpreter Prolog	6

1. Zadanie 10

1.1. Programowanie w logice

1.1.1. Baza wiedzy

Przed przystąpieniem do wykonywania zapytań należy sporządzić bazę wiedzy. Są to pliki tekstowe z rozszerzeniem `pl`. Należy przy tym pamiętać, że elementy o tej samej nazwie powinny być zdefiniowane obok siebie. Kolejność definicji ma znaczenie.

1.1.2. Obiekty i zmienne

Obiekty w Prologu zapisuje się od małej litery np. `piotr`, `magda`, `ola`. Zmienne zapisuje się od wielkiej litery lub podkreślenia np. `X`, `Y`, `Kot`, `_jak`.

1.1.3. Fakty

Fakty opisują obiekty (jakąś relację obiektu), może to być relacja jedno lub wieloargumentowa. Interpretacja faktów jest zadaniem programisty. Zapisuje się je tak jak wywołanie funkcji w języku C++ i kończy znakiem kropki np.:

- `kobieta(magda).` - co oznacza, że obiekt `magda` jest kobietą,
- `matka(ola, magda).` - może to znaczyć, że `magda` jest matką `oli` (lub odwrotnie wszystko zależy od interpretacji jaką postawimy),

1.1.4. Koniunkcje

Koniunkcje zapisuje się po to, żeby zaznaczyć, że muszą być spełnione wszystkie fakty (jest to po prostu iloczyn logiczny). Np. `lubi(jan, wino)`, `lubi(jan, jedzenie)` - co oznacza że `jan` musi lubić zarówno `wino` jak i `jedzenie`.

1.1.5. Zapytania

Pozwalają na sprawdzenie faktów, na podstawie zdefiniowanej bazy wiedzy. Jeśli interpreter jest w stanie odnaleźć spełniające zapytanie cele wypisuje odnalezione fakty ze słowem `yes`, natomiast jeśli nie może na podstawie umieszczonych w bazie danych (prologowej definiowanej przez programistę za pomocą faktów i reguł) wypisuje `no`.

np. `kobieta(magda)` pozwoli sprawdzić, czy magda jest kobietą, `kobieta(X)` odnajdzie wszystkie kobiety w bazie, kolejne cele są poszukiwane po średniku (;).

1.1.6. Reguły

Używa się ich do zapisu, że fakt zależy od innych faktów, używa się ich także do zapisywania definicji. Np. `lubi(piotr, X) :- wino(X)` – pozwala zdefiniować fakt, że ip otr lubi wszystkie wina, bez względu na etykietę (nie jest szczególnie wybredny).

1.1.7. Interpreter Prolog

Na zajęciach można używać `swi-prolog` lub `gnuprolog`. Pierwszy z nich jest uruchamiany poleceniem `pl` (w nowszych wersjach `swipl`), zaś drugi komendą `gprolog`.

Po uruchomieniu interpretera powinien pow

```
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 5.6.64)
```

```
Copyright (c) 1990-2008 University of Amsterdam.
```

```
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free  
software,
```

```
and you are welcome to redistribute it under certain conditions.
```

```
Please visit http://www.swi-prolog.org for details.
```

```
For help, use ?- help(Topic). or ?- apropos(Word).
```

```
?-
```

Znaki `?-` są znakami zachęty interpretera.

Pliki z bazą wiedzy ładuje się za pomocą komendy `[nazwa_pliku_bez_rozszerzenia_pl]`.

Po załadowaniu pliku z bazą można już wykonywać zapytania na tej bazie.