

Prolog (Pro-Logic) Programowanie w Logice

Dr inż. Piotr Urbanek

Do czego służy ProLog?

Używany w wielu systemach informatycznych związanych z:

- logiką matematyczną (automatyczne dowodzenie twierdzeń);
- przetwarzaniem języka naturalnego;
- symbolicznym rozwiązywaniem równań;
- sztuczną inteligencją;
- przechowywaniem i przetwarzaniem danych.

Pisanie programu w Prologu nie polega na opisywaniu algorytmu!

Prolog jest językiem **opisowym** i **deklaratywnym**.

Różnice między ProLogiem a OOP

Rozwiązanie jakiegoś problemu w ProLogu nie polega na stworzeniu odpowiedniego algorytmu (jak to ma miejsce w imperatywnych językach programowania tj. C lub Java) ale określeniu **czego on dotyczy**. Używane są do tego **fakty** i **reguły** ich przetwarzania.

Rolą Prologu jest **wywnioskowanie** rozwiązania na podstawie podanych informacji.

Obiekt „klasyczny”

Posiada dwie istotne cechy:

- **stan** w jakim w danej chwili się znajduje
- **zachowanie** jakie dla nich jest typowe.

Stan przechowuje w zmiennych zwanych polami

Zachowanie jest definiowane za pomocą funkcji zwanych metodami.

Każdy utworzony przez programistę obiekt jest instancją pewnej **klasy**.

```
class Punkt
{
private:
double x;
double y;
public:
Punkt(double x, double y);
void Przesun(Wektor w);
};
```

Różnice między ProLogiem a OOP

Obiekt w Prologu **bytem**. Nie ma definicji **z czego** się on składa i co można z nim zrobić (co ma miejsce w OOP), ale **jaki jest**.

Dla każdego obiektu definiowane są **relacje** jakim obiekt ten podlega.

Poniższe fakty stwierdzają, że:

ciezszy(pomarańcza,jablko). – pomarańcza jest cięższa od jabłka,

ciezszy(jablko,mandarynka). – jabłko jest cięższe od mandarynki.

*Pomarańcza, jabłko, mandarynka – **obiekty***

***Ciezszy** – relacja między obiektami.*

Programowanie w ProLogu

ciezszy(arbuz,pomarancza).

ciezszy(pomarancza,jablko).

ciezszy(jablko,winogrono).

arbuz > pomarancza > jablko > winogrono

Jak zapisać w ProLogu przechodność relacji?

ciezszy(X,Y) :- ciezszy(X,Z),ciezszy(Z,Y).

Odpowiedź

?- ciezszy(arbuz,X).

X = pomarancza ;

X = jablko ;

X = winogrono ;

ERROR: Out of local stack

Praca z Prologiem składa się z następujących etapów:

1. Definiowanie obiektów poprzez definiowanie **faktów** dotyczących obiektów i związków między nimi.
2. Definiowanie **reguł** dotyczących obiektów i związków między nimi.
3. Zapytania o obiekty i związki między nimi.

Programowanie w ProLogu

Konwencja pisania programu w ProLogu:

- Nazwy relacji i obiektów muszą zaczynać się małymi literami.
 - Nazwy rozpoczynające się od dużej litery oznaczają zmienne.
 - Najpierw zapisujemy relację, a potem, rozdzielone przecinkami i ujęte w nawias okrągły, obiekty których ona dotyczy.
 - Nazwy obiektów występujących w nawiasach nazywamy **argumentami**.
 - Nazwę relacji znajdującej się przed nawiasem nazywamy **predykatem**.
 - Nie można jako predykatu użyć zmiennej. Innymi słowy, nie można się dowiedzieć jaka relacja łączy obiekty jas i malgosia
- X(jas,malgosia).
- Fakt i regułę kończymy znakiem kropki.

- Kolejność obiektów umieszczonych w nawiasie jest dowolna, ale trzeba stosować ją konsekwentnie. O ile bowiem dobrze znanym faktem jest to, że Ala lubi swojego kota, to nie oznacza to, że kot ten lubi Alę.
- Zbiór faktów i reguł nazywamy **bazą danych**.
- Składnia reguły jest następująca:

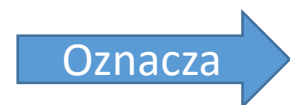
<głowa reguły> :- <treść reguły>.

Praca z programem – zapytania

Wczytywanie programu:

? [plikBezRozszerzenia].

?- posiada(piotr,ksiazka)



Czy Piotr posiada książkę?

Czy istnieje fakt mówiący, że Piotr ma książkę?

?- posiada(piotr,X).



Co ma Piotr?

Jeśli Piotr ma X, to X jest tym czego szukam.

lubi(jas,piernik).

lubi(jas,malgosia).

lubi(malgosia,cukierek).

lubi(malgosia,piernik).

?- lubi(jas,malgosia), lubi(malgosia,jas).

Czy prawdą jest, że Jaś lubi Małgosię i Małgosia lubi Jasia?

Praca z programem – zapytania

?- lubi(jas,X), lubi(malgosia,X).

Szukam tego wszystkiego co lubi zarówno Jas jak i Małgosia.

?- lubi(jas,X); lubi(malgosia,X).

Szukam tego wszystkiego co lubi Jas lub co lubi Małgosia.

Podsumowanie

Program Prologu składa się z **termów**. Wyróżniamy cztery rodzaje termów:

- **atomy** (ang. *atoms*),
- **liczby** (ang. *numbers*),
- **zmienne** (ang. *variables*)
- **termy złożone** (ang. *compound terms*).

Każdy term zapisywany jest jako ciąg znaków pochodzących z następujących czterech

kategorii:

- **duże litery**: A-Z
- **małe litery**: a-z
- **cyfry**: 0-9
- **znaki specjalne**: % + - * / \ ~ ^ < > : . ? @ # \$ &

Zbiór ten uzupełnia **znak podkreślenia** (_), który zwykle traktowany jest jak litera.

Podsumowanie

Atomy

Atom jest ciągiem znaków utworzonym z

- małych i dużych liter, cyfr i znaku podkreślenia z zastrzeżeniem, że pierwszym znakiem musi być mała litera, np. jas, a, aLA, x_y_z, abc
- dowolnego ciągu znaków ujętego w apostrofy, np. 'To też jest atom'
- symboli, np. ?- lub :- .

Liczby

W SWI-Prologu dostępne są zarówno liczby całkowite jak i rzeczywiste

-17, 23, 99.9, 123e-3

Podsumowanie

Zmienne

Zmienna jest ciągiem znaków utworzonym z małych i dużych liter, cyfr i znaku podkreślenia z zastrzeżeniem, że pierwszym znakiem musi być duża litera lub znak podkreślenia,

np. X, Kto, _123, X_1_2, _

Ostatni z wymienionych przykładów, pojedynczy znak podkreślenia, to tak zwana **zmienna anonimowa**. Korzystamy z niej zawsze wtedy, gdy interesuje nas tylko czy coś jest prawdą, ale zupełnie nie interesuje nas co, np.

Czy ktoś lubi Jasia?

?- lubi(_,jas).

?- a(1,2)=a(X,Y).

X = 1,

Y = 2

?- a(1,2)=a(X,X).

No

?- a(1,2)=a(_,_).

Yes

Podsumowanie

Termy złożone

```
posiada(piotr,auto(nissan,almera)).  
posiada(marcin,auto(fiat,punto)).  
maAuto(X) :- posiada(X,auto(_,_)).
```