

Programowanie w logice

PROLOG

- PROLOG – język wysokiego poziomu
- Powstał w 1972 na Uniwersytecie w Marsylii (Francja) w zespole A.Colmerauer'a i F.Roussel'a
- PROgrammation en LOGique, PROgramming in LOGic, PROramowanie w LOGice

LPI-2012/2013 MKG

- Podstawa utworzenia PROLOGu: logika predykatów pierwszego rzędu oraz zasada rezolucji.
- Program napisany w PROLOGu zawiera zbiór asercji (faktów), które mogą być traktowane jako aksjomaty pewnej teorii oraz zbiór reguł wnioskowania (dedukcji) dla tej teorii. Problem do rozwiązania jest twierdzeniem teorii, które należy udowodnić.

LPI-2012/2013 MKG

- Języki proceduralne (imperatywne) – program zawiera szczegółowy opis kolejnych kroków postępowania zgodnie z przyjętym algorytmem tak, aby z wprowadzonych danych otrzymać żądane wyniki (PASCAL, FORTRAN, C)
- Język deklaratywny – wymaga określenia związków (relacji) między danymi a wynikiem (PROLOG)

LPI-2012/2013 MKG

Zastosowania:

- systemy komputerowe wykorzystujące metody sztucznej inteligencji (Artificial Intelligence),
- szczególnie w systemach ekspertowych (Expert System)
- i systemach z bazą wiedzy (Knowledge-based Systems)

LPI-2012/2013 MKG

AI obejmuje:

- Systemy ekspertowe (doradcze)
- Uczenie się przez komputer
- Automatyczne dowodzenie twierdzeń
- Przetwarzanie języka naturalnego
- Robotyka
- Rozpoznawanie obrazów
- Rozpoznawanie mowy
- Gry komputerowe

LPI-2012/2013 MKG

Fakty

- Opisują związki między obiektami, opisują obiekty. Przedstawia się je za pomocą predykatów.
- predykat(obiekt1, obiekt2, obiekt3,... obiektn).
- lubi(anna, piotr).
lubi(piotr, anna).
kobieta(anna).
ojciec(jan, marek).
odleglosc(poznan, warszawa, 300).

LPI-2012/2013 MKG

Fakty

- Nazwy obiektów występujące w nawiasach nazywamy argumentami.
- Zbiór faktów nazywamy bazą danych.

LPI-2012/2013 MKG

Cel

- Bazę danych można wykorzystać poprzez zadawanie pytań lub inaczej celów do realizacji.
- Celem, w zależności od formy, jest:
 - pytanie o prawdziwość faktów,
 - polecenie znalezienia obiektów będących w relacji z innymi obiektami.

LPI-2012/2013 MKG

Zapytania (cele)

lubi(jan, ryby).
lubi(jan, książki).
lubi(jan, kino).
lubi(ewa, książki).
lubi(ewa, kwiaty).

?- lubi(jan, kino).

yes

?- lubi(jan, konie).

no

LPI-2012/2013 MKG

Zapytania

?- lubi(jan,X).

X=ryby;

X=książki;

X=kino.

?- lubi(X,książki).

X=jan;

X=ewa.

?- lubi(Y,kwiaty).

Y=ewa.

?- lubi(ewa,Z).

Z=książki;

Z=kwiaty.

LPI-2012/2013 MKG

Reguły

- stwierdzenia dotyczące obiektów i ich powiązań, opisują zależności między obiektami

predykat(obiekt1, obiekt2, obiekt3,..., obiektn) **if** predykat1(obiekt11, obiekt12, obiekt13,..., obiekt1m1) **and** predykat2(obiekt21, obiekt22, obiekt23, ..., obiekt2m2) **and** ... predykat(obiektk1, obiektk2, obiektk3,..., obiektkmk).

LPI-2012/2013 MKG

Reguły

- Np.

siostra(X,Y) :- kobieta(X),rodzice(M,O,X),
rodzice(M,O,Y), X\=Y.

X jest siostrą Y, jeśli X jest kobietą i
X i Y mają takich samych rodziców

- Predykat **siostra** jest tutaj **głową reguły** (głowa składa się tylko z jednego predykatu), zaś warunki: **kobieta(X)**, **rodzice(M,O,X)**, **rodzice(M,O,Y)**, **X\=Y** tworzą **ciało reguły**.

LPI-2012/2013 MKG

Poszukiwanie odpowiedzi

- Zapisane w bazie danych fakty i reguły analizowane są **od góry do dołu w kolejności wprowadzenia**. Szukany jest fakt potwierdzający zapytanie.
- Jeżeli w pytaniu jest zmienna, to w trakcie wyszukiwania odpowiedzi jest **ukonkretniana** (podstawiane są pod nią stałe wartości).
- Jeżeli zapytanie jest złożone, to zawsze poszukuje się potwierdzenia predykatów od lewego do skrajnie prawego. Powrót do wcześniejszych predykatów celem sprawdzenia wszystkich kombinacji nazywa się **nawracaniem** (backtracking).

LPI-2012/2013 MKG

- Fakty i reguły stanowią tzw. **klauzule**.

- Fakt to klauzula składająca się tylko z nagłówka (nie posiada treści).

- Zbiór klauzul, w których predykaty tworzące nagłówki mają tę samą nazwę i liczbę argumentów tworzą **procedurę**.

LPI-2012/2013 MKG

Proceduralna i deklaratywna interpretacja klauzul

Każdą klauzulę o ogólnej postaci:

A :- B₁, B₂,..., B_n.

można interpretować w sposób:

- deklaratywny
- proceduralny

LPI-2012/2013 MKG

Proceduralna i deklaratywna interpretacja klauzul

- **Deklaratywna interpretacja** klauzuli:
A zachodzi, jeśli zachodzą (są prawdziwe) **B₁, B₂,..., B_n.**
- **Proceduralna interpretacja** klauzuli:
w celu obliczenia **A** należy najpierw obliczyć **B₁, B₂,..., B_n.**

Obliczenie B₁, B₂,..., B_n jest równoznaczne z wywołaniem odpowiednich procedur B₁, B₂,..., B_n.

LPI-2012/2013 MKG

Unifikacja

Term to stała, zmienna lub struktura.

- **Stałe** (obiekty proste, struktury atomowe, atomy) to symboliczne nazwy obiektów występujących w programie.

Przykłady:

j23, 1, 2, 007, jan, aula_A,

jan_kowalski, 'Jan Kowalski', '125,00'

LPI-2012/2013 MKG

Unifikacja

- **Zmienne** symbolicznie przedstawiają nazwy obiektów, które nie są w tej chwili znane.

Przykłady: X, A, Ogon_listy, _miasto

- **Struktura** – obiekt składający się z innych obiektów, określona jest przez funktor oraz nazwy obiektów składowych (argumentów funktora)

Przykłady:

ksiazka(adam, mickiewicz, pan_tadeusz, 2012)

ksiazka(autor(adam, mickiewicz),
tytul(pan_tadeusz), wydanie(100, 2012))

LPI-2012/2013 MKG

Unifikacja

Unifikacja termów T1 i T2 polega na szukaniu wyrażeń, jakie trzeba podstawić pod zmienne występujące w T1 i T2, by po ich podstawieniu termy stały się identyczne. Jeśli takiego postawienia nie ma, to unifikacja zawodzi.

T1=T2

LPI-2012/2013 MKG

Unifikacja

Jeżeli termy T1 i T2 są zmiennymi, np. X i Y to przy próbie uzgodnienia tych zmiennych możliwe są przypadki:

- Zmienna X jest ukonkretniona, czyli związana z pewną stałą (strukturą), a Y jest wolna – wtedy Y zostanie ukonkretniona przez wartość zmiennej X.
np.

stolica(berlin, niemcy)=Y.

LPI-2012/2013 MKG

Unifikacja

- Zmienna X jest wolna, a Y ukonkretniona, wtedy X zostanie ukonkretniona przez wartość zmiennej Y.

np.

X=madryt.

LPI-2012/2013 MKG

Unifikacja

- Jeśli obie zmienne są wolne, to wtedy następuje ich powiązanie, czyli jeśli w pewnym momencie działania programu jedna z nich zostanie ukonkretniona, to druga automatycznie przyjmie tę samą wartość.

np.

X=Y.

LPI-2012/2013 MKG

Unifikacja

- Jeśli T1 i T2 są **stałymi** (atomami lub liczbami) to równość zachodzi, gdy ta sama stała występuje po obu stronach predykatu =.

np.

praga=praga.

2010=2010.

LPI-2012/2013 MKG

Unifikacja

Dwie struktury są sobie równe, jeśli

- a) są opisane przez ten sam funktor,
- b) funktory mają tę samą liczbę argumentów,
- c) odpowiednie argumenty są sobie równe.

np.

kolor(niebieski, auto)=kolor(niebieski, auto).

Uzgadnianie:

staw(morskie_oko, tatry)=staw(X, tatry).

LPI-2012/2013 MKG

Unifikacja

- $a(X, Y, Z) = a(s, t, v)$.
- $X = 1$.
- $X = \text{uczenia}(\text{uam})$.
- $\text{uczelnia}(\text{uam}, \text{poznan}) = \text{uczelnia}(Y, \text{poznan})$.
- $\text{stolica}(X, \text{polska}) = \text{stolica}(\text{warszawa}, P)$.

LPI-2012/2013 MKG

Porównywanie wartości

- **X=Y**

Porównanie kończy się sukcesem, gdy oba wyrażenia są identyczne lub da się je uzgodnić

- **X\=Y**

Porównanie kończy się sukcesem, gdy wyrażen nie daje się uzgodnić

LPI-2012/2013 MKG

Porównywanie wartości

□ $X=Y$

Predykat $X=Y$ również oznacza równość, ale w węższym znaczeniu niż $X=Y$.

Jeśli X lub Y w wyrażeniu $X=Y$ jest zmienną, to następuje uzgodnienie.

W przypadku $X=Y$ uzgodnienie nie nastąpi, jeśli jedna ze zmiennych ma przypisaną wartość, a druga nie.

Predykat $=$ traktuje zmienną nieukonkretnioną jako równą dowolnej wartości,

dla predykatu $==$ zmienna nieukonkretniona jest równa jedynie zmiennej z nią związanej

□ $X\backslash==Y$

LPI-2012/2013 MKG

Lista operatorów arytmetycznych i porównania

- $+$ dodawanie
- $-$ odejmowanie
- $/$ dzielenie
- $//$ dzielenie całkowite
- $*$ mnożenie
- $**$ potęga
- **mod** reszta z dzielenia
- **is** znak równości (wynik obliczeń arytmetycznych) np. (X is 1 mod 3.)
- $==$ czy wartości równe
- $=\backslash$ czy wartości różne
- $>$ większe
- $<$ mniejsze
- $>=$ większe równe
- $<=$ mniejsze równe

LPI-2012/2013 MKG

Operator "is"

- „is” służy do ukonkretniania występującej po lewej stronie zmiennej przez *wyrażenie arytmetyczne* znajdującą się po prawej stronie.

- $?- X$ is $2+3$.

$X = 5$

Yes

LPI-2012/2013 MKG

Operacja równości ($=$) a unifikacja ($=$)

Operator $=$ sprawdza dopasowanie dwóch obiektów i jeśli jest ono możliwe, prowadzi do wiązania zmiennych w tych obiektach (bez obliczania wartości wyrażeń!).

Operator $==$ powoduje obliczenie wartości argumentów bez wiązania zmiennych (muszą być one już związane).

Przykład

$?- 1+2==2+1.$
Yes

$?- 1+A=B+2.$
 $A=2$
 $B=1$

$?- 1+2=2+1.$
No

$?- 1+A:=B+2.$
ERROR:Arguments are not
sufficiently instantiated

LPI-2012/2013 MKG

Zastosowanie operacji arytmetycznych

Największy wspólny dzielnik dwóch liczb: dla dwóch liczb całkowitych, dodatnich X i Y , największy wspólny dzielnik D :

- równa się X , jeżeli X i Y są równe,
- równa się największemu wspólnemu dzielnikowi X i $Y-X$, jeżeli $X < Y$,
- równa się największemu wspólnemu dzielnikowi Y i $X-Y$, jeżeli $Y < X$.

Zapis w Prologu:

```
nwd(X,X,X).  
nwd(X,Y,D):- X<Y, Y1 is Y-X, nwd(X,Y1,D).  
nwd(X,Y,D):- Y<X, nwd(Y,X,D). lub  
nwd(X,Y,D):- Y<X, X1 is X-Y, nwd(X1,Y,D).
```

LPI-2012/2013 MKG

SWI-PROLOG

- www.swi-prolog.org
- Programy prologowe zapisuje się używając dowolnego edytora np. write, pico, vim, emacs, notatnik (Windows).
- Plik zapisujemy z rozszerzeniem **pl**.
- Aby wczytać do pamięci plik prologowy, należy wydać interpreterowi następujące polecenie:
 $?- [nazwa_pliku]$.
- **Uwaga:** Wczytywane pliki muszą być w tym samym katalogu, spod którego uruchamiany jest program pl.

LPI-2012/2013 MKG

Literatura

- W.F.Cloksin, C.S.Mellish, Prolog.Programowanie, Helion.
- E.Gatnar, K.Stapor, Prolog, Wyd. PLJ.
- G.Brzykcy, A.Meissner, Programowanie w Prologu i programowanie funkcyjne, Wyd.PP.
- F.Kluźniak, S.Szpakowicz, Prolog, WNT.
- R.Kowalski, Logika w rozwiązywaniu zadań, WNT.

LPI-2012/2013 MKG