

PROLOG: VGRAJENI PREDIKATI

Ivan Bratko
Univerza v Ljubljani

SKUPINE VGRAJENIH PREDIKATOV

- Testiranje tipa izraza
- Konstrukcija in dekompozicija izrazov: = . . . , *functor*, *arg*, *name*
- Razne vrste enakosti in primerjanja
- “Database manipulation”: *assert*, *retract*
- Krmiljenje izvajanja
- *bagof*, *setof* and *findall*
- Input, output

V nadaljevanju sledijo primeri za ilustracijo nekaterih izbranih vgrajenih predikatov.

TESTIRANJE TIPA OBJEKTA

- **var(X)** X je neopredeljena spremenljivka
- **nonvar(X)** X ni (opredeljena) spremenljivka
- **atom(X)** X je (trenutno) atom
- **integer(X)** X je (trenutno) celo število
- **float(X)** X je (trenutno) realno število
- **number(X)** X je (trenutno) število
- **atomic(X)** X je (trenutno) atom ali število
- **compound(X)** X je (trenutno) struktura (sestavljen objekt)

PRIMER: „ŠČITENJE“ ARITMETIČNE OPERACIJE

... ,

number(X),

% Vrednost X število?

number(Y),

% Vrednost Y število?

Z is X + Y,

% Potem je možno seštevanje

...

Konstrukcija in dekompozicija izrazov: =.. , *functor*, *arg*, *name*

- Term =.. [Functor, Arg1, Arg2, Arg3, ...] % „univ“
- Primer: Povečaj geometrijski lik za faktor 1.5

?- Figure = square(3), % Kvadrat s stranico 3

...

Figure =.. [Type, Size],

NewSize is 1.5 * Size,

NewFigure =.. [Type, NewSize].

NewFigure = square(4.5).

ZAMENJAJ PODIZRAZ V IZRAZU Z NOVIM PODIZRAZOM

substitute(Subterm, Term, Subterm1, Term1):

if all occurrences of Subterm in Term are substituted with Subterm1
then we get Term1.

?- substitute(sin(x), 2*sin(x)*f(sin(x)), t, F).

F = 2*t*f(t)

% Case 1: Substitute whole term

substitute(Term, Term, Term1, Term1) :- !.

% Case 2: Nothing to substitute if Term atomic

substitute(_, Term, _, Term) :-

atomic(Term), !. % Term is a constant

% Case 3: Do substitution on arguments

substitute(Sub, Term, Sub1, Term1) :-

Term =.. [F | Args], % Get arguments

substlist(Sub, Args, Sub1, Args1), % Perform substitution on them

Term1 =.. [F | Args1]. % Construct Term1

% substlist(SubTerm, Term_List, NewSubTerm, NewTerm_List)

PRIMER UPORABE substitute/4

?- E0 = (a + b) * (a - b),
 substitute(a, E0, 6, E1),
 substitute(b, E0, 3, E2),
 Value is E2.

$$E1 = (6 + b) * (6 - b)$$

$$E2 = (6 + 3) * (6 - 3)$$

$$\text{Value} = 27$$

Razne vrste enakosti in primerjanja

- $X = Y$ prilagajanje
- $X == Y$ dobesedno enako
- $X \neq Y$ ne dobesedno enako
- $X @< Y$ X leksikografsko manjše

„DATABASE MANIPULATION“

- assert(Clause)
- asserta(Clause), assertz(Clause)
- retract(Clause)

- Primer: svet robota

% move(X, Y, Z): move block X from Y to Z

move(X, Y, Z) :-

retract(on(X,Y)), !, % Pred premikom

assertz(on(X,Z)). % Po premiku

MNOŽICE REŠITEV

- `findall(Object, Condition, List)`
List = seznam objektov Object, ki zadoščajo pogoju Condition
- `bagof(Object, Condition, List)` % Poišče objekte
- `setof(Object, Condition, List)` % Poišče in uredi
- Primeri iz robotovega sveta:
- `?- findall(B, on(B,_), L).` % L je seznam vseh kock
L = [a,b,c,d,e]
- `?- setof(Z:B, B2^(on(B,B2), z(B,Z)), L).` % Kocke urejene po Z-koord
L = [0:c, 0:d, 0:e, 1:b, 2:c]