

Логическое программирование

Кевролетин В.В. группа с8403а(246)

17 May 2012

Содержание

1	Задание 22	1
1.1	Условие	1
1.2	Решение	1
1.2.1	Исходный код	1
1.2.2	Тесты	2

1 Задание 22

1.1 Условие

Написать программу для отношения `substitute` для термов с использованием `univ`.

1.2 Решение

Воодится вспомогательный терм `substituteeach` для итерации по списку, полученному при помощи предиката `univ`:

1.2.1 Исходный код

```
substitute(Old, New, Old, New).
```

```
substitute(Old, _, Term, Term) :-  
    atomic(Term),  
    \+ compare(=, Old, Term).
```

```
substitute(Old, New, Term, Term1) :-  
    compound(Term),  
    Term =.. [F | Args],  
    substitute_each(Old, New, Args, NewArgs),  
    Term1 =.. [F | NewArgs].
```

```
substitute_each(Old, New, [Old | Xs], [New | Ys]) :-  
    substitute_each(Old, New, Xs, Ys).
```

```
substitute_each(Old, New, [X | Xs], [X | Ys]) :-
```

```
substitute_each(Old, New, Xs, Ys).
```

```
substitute_each(_, _, [], []).
```

1.2.2 Тесты

Для тестирования используется расширения языка SWI-Prolog

```
:- begin_tests(substitute).
```

```
test(t01) :-
    substitute(a, b, a, b).
test(t02) :-
    substitute(a, b, c, c).
test(t03) :-
    substitute(a, b, node(a), node(b)).
test(t04) :-
    substitute(a, b, node(a, a), node(b, b)).
test(t05) :-
    substitute(node(a, b), node(c, d),
               node(node(a, b),
                     node(c, d)),
               node(node(c, d),
                     node(c, d))).
```

```
:- end_tests(substitute).
```