

Логическое программирование

Кевролетин В.В. группа с8403а(246)

17 May 2012

Содержание

1	Задание 19	1
1.1	Условие	1
1.2	Решение	1
1.2.1	Исходный код	1
1.2.2	Тесты	2

1 Задание 19

1.1 Условие

Отношение для упорядоченного дерева.

1.2 Решение

Бинарное дерево является упорядоченным, если для каждого узла верно, что значение в нём больше любого значения в узлах из левого поддерева и больше либо равно любому значению из правого поддерева.

Предикат `is_ordered_tree/1` проверяет, является ли дерево упорядоченным, используя предикат `check_ordering/3`. `check_ordering/3` проверяет, выполняется ли условие упорядоченности для текущего узла и его правого и левого поддерева.

1.2.1 Исходный код

```
is_data(X) :-
    number(X).

get_data(node(Data, _, _), Data) :-
    is_data(Data).

is_ordered_tree(leaf(Data)) :-
    is_data(Data).

is_ordered_tree(node(Data, Left, Right)) :-
    is_data(Data),
    check_ordering(Data, Left, Right),
    is_ordered_tree(Left),
    is_ordered_tree(Right).
```

```

is_ordered_tree(null).

check_ordering(Data, Left, Right) :-
    check_less_eq_or_null(Data, Right),
    check_greater_or_null(Data, Left).

check_less_eq_or_null(Data, null) :-
    is_data(Data).

check_less_eq_or_null(Data, node(DataNode, _, _)) :-
    is_data(Data),
    =<(Data, DataNode).

check_greater_or_null(Data, null) :-
    is_data(Data).
check_greater_or_null(Data, node(DataNode, _, _)) :-
    is_data(Data),
    >(Data, DataNode).

```

1.2.2 Тесты

Для написания тестов использовалось расширение языка Prolog в реализации SWI-Prolog.

```

:- begin_tests(is_ordered_tree).

test(test01) :-
    is_ordered_tree(node(10, null, null)), !.
test(test02) :-
    is_ordered_tree(null), !.
test(test03) :-
    is_ordered_tree(null), !.
test(test04) :-
    is_ordered_tree(node(10,
                        node(5, null, null),
                        node(15, null, null))), !.
test(test04) :-
    is_ordered_tree(node(10,
                        node(5,
                            node(0, null, null),
                            null),
                        node(15,
                            null,
                            node(20, null, null)))))), !.
test(test04 [false]) :-
    is_ordered_tree(node(10,
                        node(5,
                            node(0, null, null),
                            node(0, null, null)),
                        node(15,
                            null,
                            node(20, null, null)))))), !.

```

```
:- end_tests(is_ordered_tree).
```