

Практическое занятие «Пролог–2»

22 сентября 2020 года

Задачи, помеченные звёздочкой, являются бонусными: стоят 5 баллов, не теряют ценности со временем и не входят в тот набор задач, которые определяют баллы на аттестацию/допуск к зачету/автомат.

1. Напишите предикат `toSquares/3(+integer,+integer,?integer)`, истинный, если третий аргумент есть количество квадратных кусков, на которые можно разрезать прямоугольник, размеры которого передаются первыми двумя аргументами, если от него каждый раз отрезается квадрат максимально большой площади. Например:

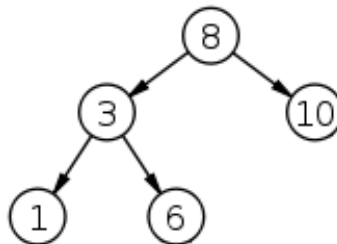
- `toSquares(2,1,2) → true`;
- `toSquares(3,5,X) → true, X = 4`;
- `toSquares(3,1,2) → false`.

2. Напишите предикат `qntDigit/4(+integer,+integer,+integer,?integer)`, истинный, если последний аргумент есть количество целых чисел, содержащих в своей десятичной записи цифру, переданную третьим аргументом, выбираемых из диапазона, заданного первыми двумя аргументами (включительно). Диапазон может включать кроме положительных и отрицательные числа, и ноль. Например:

- `qntDigit(11,23,2,5) → true`; в интервале $[11, 23]$ имеется пять целых чисел, в десятичной записи которых имеется цифра 2: 12, 20, 21, 22, 23;
- `qntDigit(-10,10,0,X) → true, X = 3`.

Разумно написать вспомогательный предикат, который истинен, если в десятичной записи данного числа содержится данная цифра.

3. Напишите предикат `fact/2(?integer,?integer)`, истинный, если второй аргумент есть факториал первого. Считаем, что факториал не определён для отрицательных чисел. Продумайте аккуратно поведение предиката в случае работы в режиме поиска по первому аргументу или по обоим аргументам: `fact(N,120)` и `fact(N,F)`.
4. Бинарное (двоичное) дерево можно хранить в виде рекурсивной структуры данных. Например, дерево



можно представить в виде терма

`tr(8, tr(3, tr(1, nil, nil), tr(6, nil, nil)), tr(10, nil, nil))`

Первый аргумент тернарного термина **tr** — это число, хранящееся в узле, второй и третий аргументы представляют собой левое и правое поддеревья в том же виде. В случае отсутствия того или иного поддерева указывается атом **nil**.

Деревом поиска называется бинарное дерево такое, что любой элемент левого поддерева некоторого узла меньше элемента, хранящегося в этом узле, а любой элемент правого поддерева — больше.

Напишите следующие предикаты:

- а) предикат **insert/2(+tree,+integer,?tree)**, истинный, если третий аргумент есть дерево поиска, полученное вставкой элемента, переданного вторым аргументом, в дерево поиска, переданное первым элементом. Если элемент уже имеется в дереве, то при его вставке дерево не изменяется.
- б) предикат **contains/2(+tree,?integer)**, истинный, если целое число, переданное вторым аргументом, содержится в дереве поиска, переданном в первом аргументе. Если второй аргумент есть свободная переменная, то предикат должен поочередно сопоставлять с этой переменной все элементы переданного дерева. Например,

```
| ?- contains(tr(2,tr(0,nil,tr(1,nil,nil)),tr(4,nil,nil)),1).  
yes  
| ?- contains(tr(2,tr(0,nil,tr(1,nil,nil)),tr(4,nil,nil)),3).  
yes  
| ?- contains(tr(2,tr(0,nil,tr(1,nil,nil)),tr(4,nil,nil)),E1).  
E1 = 0 ? ;  
E1 = 1 ? ;  
E1 = 2 ? ;  
E1 = 4 ? ;  
no
```

- в) предикат **isSearchTree/1(+tree)**, истинный, если дерево, переданное в него, является деревом поиска.
- г*) предикат **remove/2(+tree,+integer,?tree)**, истинный, если третий аргумент есть дерево поиска, полученное удалением элемента, переданного вторым аргументом, из дерева поиска, переданного первым элементом. Если элемент отсутствует в дереве, то при его удалении дерево не изменяется.

д*) предикат `printTree/1(+tree)`, «красиво» выводящий на экран переданное дерево. Например, «красиво» выведенное дерево с вышеприведённой картинки может выглядеть следующим образом:

```
8              8
|              +-3
+--3           | +-1
|  |           | +-6
|  +--1        +-10
|  |
|  +--6
|
+--10
```

Впрочем, разумный формат вывода можно придумать самому.