

Практическое занятие «Пролог–1»

10 сентября 2019 года

Стандартным методом описания предикатов Пролога является следующий: `имя_предиката/арность(+аргумент, -аргумент, ?аргумент, ...)`. В таком описании **арность** означает количество аргументов предиката; префикс **+** при описании аргумента говорит, что данный аргумент должен быть сопоставлен при вызове доказательства предиката; префикс **-** означает, что аргумент должен быть несопоставленной переменной; наконец, **?** соответствует ситуации, когда оба варианта допустимы.

Полезные предикаты для управления средой:

- `change_directory/1(+путь)`: сменить текущую папку; в случае успеха — папка существует и переход в нее не запрещен — предикат истинен и происходит смена текущей папки; в случае ошибки происходит прерывание процесса доказательства;
- `directory_files/2(+папка, ?файлы)`: предикат истинен, если **файлы** есть список имён файлов в папке **папка**; если второй аргумент — свободная переменная, то она сопоставляется со списком файлов. В случае успеха предикат истинен, в случае ошибки происходит прерывание процесса доказательства;
- `consult/1(+файл_список-файлов)`: загружает и обрабатывает один файл (в случае, когда аргумент — строка) или набор файлов, когда аргумент — список строк. Истинен в случае существования всех указанных файлов при условии корректности записанных в них листингов; ложен в противном случае.
- `listing/0`: выводит на экран имеющуюся в настоящий момент теорию.

Задания:

1. (незачётная) Загрузите и установите GNU Prolog. Запустите его: команда `gprolog`. Смените папку на ту, в которой будете работать.
2. (незачётная) Напишите программу с двумя предикатами:

`p1(X,Y,Z) :- Z=X+Y.`

`p2(X,Y,Z) :- Z is X+Y.`

Сравните поведение этих предикатов, задавая различные значения входных данных.

3. Используя пример с лекции, напишите бинарный предикат, истинный, если второй аргумент есть двойной факториал первого. Напомним, что двойной факториал числа N вычисляется как

$$N!! = \begin{cases} 1 \cdot 3 \cdot \dots \cdot N, & \text{если } N \text{ нечётно,} \\ 2 \cdot 4 \cdot \dots \cdot N, & \text{если } N \text{ чётно.} \end{cases}$$

4. Напишите предикат, принимающий истинные значения на неотрицательных целых числах, двоичная запись которых содержит чётное количество единиц. Операция взятия остатка от деления $X \bmod Y$, проверка равенства двух чисел — операция $=$, неравенства — операция \neq .
5. Напишите предикат, имеющий истинные значения на простых числах.
6. Напишите предикат `sirakuz(N,A0)`, который при своем доказательстве выводит N членов «сиракузской последовательности» с начальным членом $A0$. Предикат вывода объекта X на экран `write/1(X)`, перевод строки — предикат `nl/0`. Эти два предиката всегда истинны.

Сиракузская последовательность задается рекуррентным соотношением

$$a_{n+1} = \begin{cases} a_n/2, & \text{если } a_n \text{ — чётное число,} \\ 3a_n + 1, & \text{если } a_n \text{ — нечётное число.} \end{cases}$$

7. Напишите предикат `nextDate(D1,D2)`, истинный, если $D2$ есть запись даты дня, следующего за тем, запись которого дана в $D1$. Дату разумно записывать в виде терма `date(день,месяц)`. Информацию о количестве дней можно задать набором фактов `numOfDays(месяц,число)`; имена месяцев можно представить в виде атомов `jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec`. Информацию о следовании месяцев можно представить в виде набора фактов `nextMonth(месяц1,месяц2)`. Считаем, что формат даты выдержан: первый аргумент терма `date` всегда один из 12 атомов, представляющих имена месяцев, а второй аргумент всегда целое число. Однако дата может быть неверно заданной, например, `date(feb,31)` или `date(jun,-10)`. В таком случае доказательство предиката должно заканчиваться неуспехом. Предусмотрите работу предиката `nextDate` в режиме проверки и в режиме поиска второго аргумента, то есть формуляр предиката есть `nextDate(+date,?date)`. Например,
- `nextDate(date(sep,10),date(sep,11)) → true,`
`nextDate(date(jul,31),date(aug,1)) → true,`
`nextDate(date(apr,19),date(apr,21)) → false,`
`nextDate(date(may,29),date(jun,1)) → false,`
`nextDate(date(dec,31),X) → true, X = date(jan,1).`