

SWI Prolog

Стандартные предикаты SWI Prolog

Арифметические предикаты

Для выполнения основных арифметических действий можно воспользоваться несколькими predetermined предикатами:

$X+Y$ — сложение,

$X-Y$ — вычитание,

$X*Y$ — умножение,

X/Y — деление,

$X//Y$ — целочисленное деление ,

X^Y — возведение в степень,

$X \wedge Y$ — побитовая конъюнкция (для целых чисел),

$X \vee Y$ — побитовая дизъюнкция (для целых чисел),

$X \ll Y$ — побитовый сдвиг влево на Y позиций (для целых чисел),

$X \gg Y$ — побитовый сдвиг вправо на Y позиций (для целых чисел),

$X \bmod Y$ — остаток от деления X на Y (для целых чисел),

Арифметические предикаты

abs(X) — абсолютная величина X,
acos(X) — арккосинус X,
asin(X) — арксинус X,
atan(X) — арктангенс X,
cos(X) — косинус X,
sin(X) — синус X,
exp(X) — экспонента X,
log(X) — логарифм натуральный,
log10(X) — логарифм по основанию 10,
sqrt(X) — квадратный корень X,
tan(X) — тангенс X.

Арифметические предикаты

Операндами арифметических предикатов являются арифметические термы.

Арифметические выражения строятся из атомов и переменных с помощью арифметических предикатов. Допускается инфиксная и префиксная записи арифметических выражений. Арифметические термы без переменных являются константами.

Предикаты сравнения арифметических выражений

Пусть $E1$ и $E2$ — арифметические выражения. В Прологе существуют следующие встроенные предикаты для сравнения арифметических выражений:

$E1 > E2$ истинно, если $E1$ больше $E2$;

$E1 < E2$ истинно, если $E1$ меньше $E2$;

$E1 \geq E2$ истинно, если $E1$ больше или равно $E2$;

$E1 \leq E2$ истинно, если $E1$ равно или меньше $E2$;

Предикаты сравнения арифметических выражений

$E1 ::= E2$ истинно, если $E1$ равно $E2$,
 $E1 = \backslash = E2$ истинно, если $E1$ равно $E2$,
 $E1 = E2$ истинно, если $E1$ и $E2$ сопоставимы,
 $E1 \backslash = E2$ истинно, если $E1$ и $E2$ не сопоставимы,
 $X \text{ is } E$ истинно всегда, и
неконкретизированной переменной X присваивается значение E .

Предикаты сравнения символьных термов

Пусть $E1$ и $E2$ — символьные термы. Для сравнения символьных термов используются другие встроенные предикаты:

$E1@>E2$ истинно, если терм $E1$ больше терма $E2$,

$E1@<E2$ истинно, если терм $E1$ меньше терма $E2$,

$E1@>=E2$ истинно, если терм $E1$ больше или равен терму $E2$,

Предикаты сравнения символьных термов

$E1@=<E2$ истинно, если терм $E1$ равен или меньше терма $E2$.

Символьные термы упорядочены в алфавитном порядке.

Примеры сравнения арифметических выражений

Оператор унификации “=”

? – $X=1+2$.

$X=1+2 \rightarrow$

YES

Оператор “is”

? – $X \text{ is } 1+2$.

$X=3 \rightarrow$

YES

В случае унификации сопоставляются переменная X составной терм $1+2$, и устанавливается, что X сопоставима с $1+2$ при подстановке $\{X=1+2\}$.

Оператор is заставляет систему вычислить значение выражения справа от обозначения оператора, и это значение сопоставить с переменной X .

Различие между операторами унификации “=” и арифметического сравнения “:=”

Различие между операторами унификации “=” и арифметического сравнения “:=” состоит в том, что при выполнении оператора “=” система не производит вычислений, а оператор “:=” производит вычисление выражений и сравнение и значений.

Оператор унификации “=”

? – $1+2=2+1$.

NO

Оператор сравнения “:=”

? – $1+2 := 2+1$.

YES

Предикаты определения типов термов.

integer(X) истинно, если X—целое число;

float(X) истинно, если X—вещественное
число;

number(X) истинно, если X— целое или
вещественное число;

atom(X) истинно, если X—атом;

Предикаты определения типов термов.

atomic(X) истинно, если X—атом или число;
compound(X) истинно, если X—составной терм (структура);
novar(X) истинно, если X—константа;
var(X) истинно, если X—переменная.
string(X) истинно, если X—строка.

Предикаты ввода—вывода термов и СИМВОЛОВ.

Предопределенные предикаты стандартного ввода/вывода являются внелогическими, так как выходят за рамки модели логического программирования.

Такие предикаты в процессе вычисления запросов порождают побочный эффект.

Предикаты ввода—вывода термов и СИМВОЛОВ.

Любой язык программирования должен обеспечивать средства ввода/вывода информации. Однако вычислительная модель Пролога препятствует введению операций ввода/вывода в виде чистых компонентов языка.

Предикаты ввода—вывода термов и СИМВОЛОВ.

В системах программирования на языке Пролог используются predetermined предикаты для ввода/вывода термов и СИМВОЛОВ.

Предикат ввода термов

Для ввода термов со стандартного устройства ввода ПЭВМ (клавиатуры) используется предикат `read(<терм>)`.

Предикат `read` читает терм, набираемый пользователем на клавиатуре; после вводимого терма должна обязательно следовать точка.

Предикат ввода термов

Если переменная не конкретизирована, то запрос:

?—read(X).

приведет к вводу терма с клавиатуры, и этот терм будет присвоен в качестве значения переменной X.

Примеры ввода термов

?—read(X).

125.

X=125

Yes

?—read(X).

n.

X=n

Yes

Предикат вывода термов write

Для вывода термов на стандартное устройство вывода ПЭВМ (экран видеомонитора) используется предикат `write(<терм>)`.

Предикат `write` выводит терм на экран.

Предикат вывода термов write

В качестве термов могут выдавать на экран константы, значения конкретизированных переменных и составные термы. Если переменная не конкретизирована, то вместо ее значения будет выдано соответствующее переменной ссылочное число.

Предикаты вывода термов `nl` и `tab`

Предикат `nl` используется для перехода на новую строку при выдаче данных на экран. Предикат `tab(N)` используется для выдачи на экран `N` пробелов. Предикаты `nl` и `tab` применяются только совместно с предикатом `write`.

Пример программы с использованием арифметических предикатов.

Пусть требуется определить значение функции в зависимости от условия.

$$Y = \begin{cases} u^2 + a, & u < a, \\ b - u, & a \leq u \leq b \\ \frac{a + b}{u^2 + 1}, & u > b \end{cases}$$

Текст программы.

```
run:- write('Input A,B, U:'),nl,  
      read(A), number(A), read(B), number(B),  
      read(U), number(U), func(A,B,U,Y),  
      write('Y = '), write(Y),nl.
```

```
func(A,B,U,Y):-U<A,Y is U^2+A.
```

```
func(A,B,U,Y):-A=<U,U=<B,Y is B-U.
```

```
func(A,B,U,Y):- U>B,Y is (A+B)/(U^2+1).
```

Запросы к программе

?- run.

Input A, B, U:

-1.

3.

-2.

Y = 3 ->

yes

Запросы к программе

?- run.

Input A, B, U:

-1.

3.

1.

$Y = 2 \rightarrow$

yes

Запросы к программе

?- run.

Input A, B, U:

-1.

3.

4.

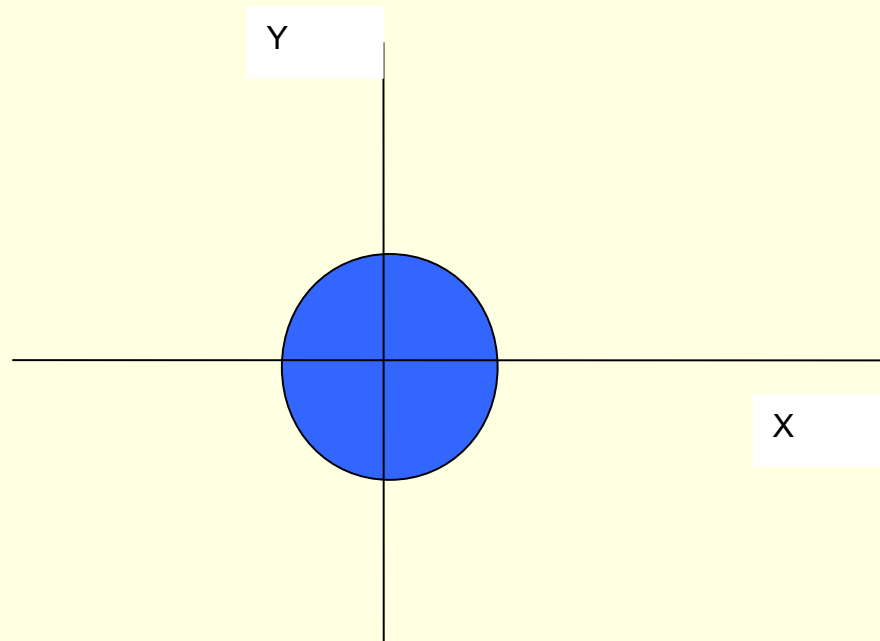
$Y = 0.11764706$

yes

Пример программ с использованием арифметических предикатов.

Пусть программа должна определять, попадает ли точка с координатами (X, Y) в область D , которая задана графически, круг с центром в точке $(0, 0)$ радиуса $r = 2$.

Заданная область



Текст программы

```
in_area(X,Y):- number(X), number(Y),(X^2+Y^2)=<4.  
out_area(X,Y):- number(X), number(Y),(X^2+Y^2)>4.  
otvet(X,Y):-in_area(X,Y),  
write('заданная точка принадлежит области D').  
otvet(X,Y):-out_area(X,Y),  
write('заданная точка не принадлежит области D').
```

Примечание. В программе используется стандартный предикат вывода на экран write (см. следующие слайды).

Текст программы, использующей предикаты ВВОДА-ВЫВОДА

```
run:- write('input X: '), read(X), write('input Y: '),  
      read(Y), number(X), number(Y), otvet(X,Y).  
run:-not(number(X)), write('Error X').  
run:-not(number(Y)), write('Error Y').
```

Текст программы, использующей предикаты ВВОДА-ВЫВОДА

```
otvet(X,Y):-in_area(X,Y),  
write('заданная точка принадлежит области D').  
otvet(X,Y):-out_area(X,Y),  
write('заданная точка не принадлежит области D').
```

Примеры запросов к программе

?- run.

input X: 1.

input Y: 1.

point in area ->

yes

?- run.

input X: 2.

input Y: 3.

point out area ->

yes

Примеры запросов к программе

?- run.

input X: q.

input Y: 2.

Error X ->

yes

?- run.

input X: 1.

input Y: a.

Error Y ->

yes

?-