## Синтаксис ПРОЛОГА.

Программа на прологе состоит из **предложений** . Предложения трех видов: **факты, правила, вопросы.** Все предложения строятся из термов.

**Терм** является синтаксической единицей.

### Константы.

**Константы** - это поименованные конкретные объекты или отношения.  
**Атомы** - аналогичны атомам или символам в лиспе.  
**Атомы** могут задаваться:

1) Цепочкой букв, цифр и символом подчеркивания '**\_**', начиная со строчной буквы.

**a  
sister  
x\_23**

2) специальными символами\*: **====>**

**Числа**Целые ( диапазон -32768 32767)  
Действительные ( диапазон 1Е-307 1Е+308)

### Переменные.

**Переменные** служат для обозначения объектов, значения которых меняются в ходе выполнения программы.

Имена **переменных** могут начинаться:  
— или с прописной буквы  
— или с символа подчеркивания

**X   
Y   
Result  
\_result**

Eсли значение переменной не интересует, то можно использовать **анонимные** переменные в виде символа подчеркивания '**\_**'.

Например,

**haschild(X):-parent(X, Y).**

Здесь значение **Y** не интересует, можно записaть

**haschald(X):-parent(X, \_).**

Значение анонимной переменной не выводится на печать. Если несколько анонимных переменных, то они все разные. Использование анонимных переменных позволяет не выдумывать имена переменных, когда не надо.

Пусть задано отношение parents для двух родителей.

**parents(ann, tom, bob).**

Тогда в правиле:

**child(X):-parents(\_, \_, X).**

**Y Z**

обе анонимные переменные разные.

! Внимание. Область действия переменных - одно предложение.

Одноименные переменные в разных предложениях могут иметь разные значения.

### Структуры.

**Структура** — это единый объект состоящий из совокупности других объектов, называемых компонентами.  
Компоненты в свою очередь могут быть также структурами.

|  |  |
| --- | --- |
|  | Структура **data**:  Название структуры стоит перед скобками,а компоненты внутри скобок, через запятую.  Название структуры — **ФУНКТОР**. |

Структуры можно изображать в виде деревьев:

**data**

**/ | \**

**27 april 1992**

Структура с компонентами - структурами **owns(bob, book(moby\_dick, mell\_will)).**

Структуры можно использовать для представления геометрических фигур.

|  |  |
| --- | --- |
| **Объекты:**  **точка Р1=point(1, 1) точка Р2=point(3, 3) отрезок** seg(P1, P2)  или  **seg(point(1, 1), point(3, 3))**  треугольник  **triangle(point(2, 5), point(2, 8), point(5, 8))** |  |

Cтруктуру можно представить в виде дерева

|  |  |
| --- | --- |
|  | Корень дерева — главный функтор. |

Если точка трехмерного пространства: **point3(X, Y, Z)**Можно записать : **point(X, Y, Z)**Получается: **point(X, Y, Z) и point(X, Y)**

Это разные термы, т.к. каждый функтор различается двумя параметрами:

* именем
* арностью - т.е. числом атомов

Поэтому пишут:

**point/2  
point/3**

Например,если написано: Отношение **point/2** задано ....  
Это означает, что задано **point(\_,\_)** ,  
а не **point/2(\_,\_)**.

!!! Это типичная ошибка.

### Операторы — тоже функторы.

Некоторые функторы удобнее записывать, как операторы.  
Например, можно записать

**+(1, 2)**

или

**+**

**/ \**

**1 2**

Удобнее записать **1+2**, т.е. в виде оператора. Причем надо понимать, что это не операция сложения, а операторная запись структуры. Такие операторы называются **инфиксными.**

Аналогично операторная запись

**2\*a+b\*c**

может быть представлена в виде структуры:

**+( \*(2, a), \*(b, c))**

Это и производит пролог при трансляции операторных выражений. Надо четко понимать, что операторы - это другая форма записи структуры.

## Арифметика.

В прологе выполняются следующие операции:

+  
-  
\*  
/  
mod— остаток от целочисленного деления.

Если записать:

|  |  |
| --- | --- |
|  | **?-X = 2+1.  X=2+1** |

т.к. это просто сопоставление переменной и структуры.

Чтобы арифметическое выражение рассчитывалось, необходимо использовать встроенный оператор **is**, который заставляет выполнять арифметические операции.

|  |  |
| --- | --- |
|  | **?-X is 2+1. X=3**  **?-Y is 2\*(5+6). Y=22** |

Могут быть переменные, но они должны иметь значение к моменту вычисления.

**f(X, Y, Z):-Z is X\*X + Y\*Y.**

и задать вопрос

|  |  |
| --- | --- |
|  | **?- f(2, 3, R). R=13** |

Если определить:

**sum(X1, X2, X):-X is X1 + X2.**

|  |  |
| --- | --- |
|  | **?-sum(2, 3, X). X=5**  **?- sum(1, 1, X1), sum(X1, X1, X). 4.** |

## 2.3 Операции сравнения.

Используются при сравнении чисел  
Есть операции

**X > Y  
X < Y  
X >= Y  
X =< Y**

равенство для любых термов

X = Y  
X \= Y

*(с побочным эффектом)*

|  |  |
| --- | --- |
|  | **?- 100 4. yes** |

**age(mary, 20).  
age(ann , 23).  
age(bob , 25).**

|  |  |
| --- | --- |
|  | **?- age(X , Y), Y>21. X=ann; X=bob;**  **?- age(X , Y), Y>21, Y<=23. X=ann ;** |

Определим отношение "выше" **higher/2**

**higher(point(X1, Y1), point(X2, Y2):-Y1>Y2.**

|  |  |
| --- | --- |
|  | **?- higher(point( 1, 1), point( 2, 2). no** |

Если наоборот - то будет **yes**.

## Сопоставление.

Главной операцией в процессе выполнения пролог -программы, является **сопоставление** (согласование, унификация) термов.

Например,

**рarent(pam, bob).**

|  |  |
| --- | --- |
|  | **?- parent(pam, bob). Yes** |

Цель согласуется.

|  |  |
| --- | --- |
|  | **?-parent(pam, X). X=bob** |

Tоже согласуются, но **X** конкретизируется и принимает значение **bob**

Но если взять два составных терма -структуры.

**a(b, C, d(e, F, g(h, i, J)))  
a(B, c, d(E, f, g(H, I, j)))**

*cогласуются ли они? Что будет с переменными после согласования? Как они будут конкретизированны?*

**Сопоставление** -это процесс, на вход которого подаются два терма, а он проверяет, соответствуют ли эти термы друг другу.

Если термы не сопоставимы, значит сопоставление терпит неудачу.  
Если термы сопоставимы, тогда пpоцесс сопоставления находит конкретизацию переменных, делающих эти термы тождественными, и завершается удачей.

Какие же правила определяют сопоставимость двух термов **S** и **Т** ?

1. Если **S** и **Т** константы , то **S** и **Т** cопоставимы, только, если они являются одним и тем же объектом.  
   т.е.:  
   **2** сопоставляется с **2  
   bob** сопоставляется с **bob**
2. Если **S** переменная, а **Т** -произвольный объект, то ни сопоставимы и **S** приписывается значение **T**. Наоборот, если **Т** -переменная, а **S** -произвольный объект, то **T** приписывается значение **S**.  
   Говорят,что **T** конкретизируется значением **S**.  
   т.е.:   
    **data(M, D, 1992).** **data(may, 3, Y).**Сопоставимы, переменные конкретизируются:  
   **М=may  
   D=3  
   Y=1992**в обоих термах.
3. Если **S** и **Т** — структуры, то они сопоставимы, если  
   (а) **S** и **Т** имеют одинаковый главный функтор  
   и  
   (b) все их соответствующие компоненты сопоставимы.  
   Результирующая конкретизация определяется сопоставлением компонент.

Если сопоставить структуры

**a(b, C, d(e, F, g(h, i, J)))  
a(B, c, d(E, f, g(H, I, j)))**

то получим

**B=b  
C=c  
E=e  
F=f  
H=h  
I=i  
J=j**

Рассмотрим более сложный пример:

**triangle(point(2, 5), A, point(B, 8))**

**triangle(X, point(2, 8), point(5, 8))**

После конкретизации

**X=point(2, 5)  
A=point(2, 8)  
B=5**

Еще один пример:

**конкретизированная переменная Х=2**

**!**

**triangle(point(X, 5), point(X, 8), point(5, Z))**

**неконкретизированная переменная Y=2**

**!**

**triangle(point(2, 5), point(Y, 8), point(5, A))**

Если **Y** представляет собой неконкретизированную переменную, а переменная **X** конкретизированную, то**X** и **Y** согласуются и принимает значение **Х**.  
т.е.

**X=2  
Y=2**

Переменные **Z** и **A** обе не конкретизированы. Они согласуются и становятся **сцепленными**.  
Если две переменные сцеплены, то при конкретизации одной из них, второй переменной автоматически будет присвоено тоже самое конкретное значение, что и первой. Как это было с **X** и **Y**.

## Второе значение операции = в Прологе.

В прологе операция **=** , кроме сравнения выполняет сопоставление двух термов, с конкретизацией переменных.

Если термы согласуются, то результат истина.

|  |  |
| --- | --- |
|  | **?-a(B,c(x,D))=a(d,c(X,3)).**  **Yes B=d X=x D=3** |

Аналогично, **/=** дает истину, если термы не согласуются.

|  |  |
| --- | --- |
|  | **?-a(x)/=a(d). Yes** |

## 2.6 Примеры сопоставления структур.

Рассмотрим структуры, описывающие отрезки

|  |  |  |
| --- | --- | --- |
|  | Два факта ,описывающие свойство вертикальность и горизонтальность.  **vertical(seg(point(X,Y),point(X,Y1))). horiz(seg(point(X,Y),point(X1,Y))).** | |
|  | **?-vertical(seg(point(2,2),point(2,5))). Yes**  **?- horiz(seg(point(1,4),point(2,4))). Yes**  **?-horiz(seg(point(1,4),point(2,X))). X=4** |