

# Основные понятия Пролога. Рекурсия на Прологе



# Предложения



- 1) Факты
- 2) Правила
- 3) Вопросы

Общий вид:

$A :- V_1, \dots, V_n.$

# Факты и правила



Пример факта:

мама(«Натasha», «Даша»).

константа, переменная,  
составной объект

Пример правил:

бабушка(X, Y) :- мама(X, Z), мама(Z, Y).

бабушка(X, Y) :- мама(X, Z), папа(Z, Y).

процедура



# Переменные

- ❖ Неявно связаны квантором всеобщности
- ❖ Не поддерживается механизм деструктивного присваивания
- ❖ Идентификатор указывает не на адрес ячейки памяти, а на объект
- ❖ Свободные (неконкретизированные) и связанные (конкретизированные)
- ❖ Область определения – одно предложение
- ❖ Все анонимные переменные – отдельные объекты

# Вопросы. Вычисление цели



мама("Наташа", "Даша").

мама("Даша", "Маша").

goal

%мама("Наташа", "Даша").

%мама("Наташа", "Маша").

%мама(X, "Даша").

%мама("Наташа", X).

%мама(X, Y).

%мама(X, \_).

%мама(\_, \_).

Возможные результаты  
работы программы:

- 1) Цель достигнута (Yes):  
либо значения  
переменных, либо No  
solutions
- 2) Цель не достигнута  
(No): либо отношение  
не выполняется, либо  
нет достаточной  
информации

# Вычисление цели



мама("Наташа", "Даша").

мама("Даша", "Маша").

бабушка(X, Y) :- мама(X, Z),  
                  мама(Z, Y).

goal

бабушка("Наташа", X).



# Нахождение максимума из двух чисел

$\max(X, Y, X) :-$

$X > Y$ . /\* если первое число больше второго,  
то первое число - максимум \*/

$\max(X, Y, Y) :-$

$X < Y$ . /\* если первое число меньше второго,  
то второе число - максимум \*/

$\max(X, Y, Y) :-$

$X = Y$ . /\* если первое число равно второму,  
возьмем в качестве максимума  
второе число \*/



# Нахождение максимума из двух чисел - 2

$\max(X, Y, X)$ :-

$X > Y$ . /\* если первое число больше второго,  
то первое число - максимум \*/

$\max(X, Y, Y)$ :-

$X \leq Y$ . /\* если первое число меньше или равно  
второму, возьмем в качестве  
максимума второе число \*/





# Нахождение максимума из двух чисел (отсечение)

$\max2(X, Y, X)$ :-

$X > Y$ , !./ \* если первое число больше второго,  
то первое число - максимум \*/

$\max2(\_, Y, Y)$ . /\* в противном случае  
максимумом будет второе число \*/



S:-

<условие>,!,P.

S :-

P2.

if <условие> then P else P2



# Нахождение максимума из трех чисел

$\max_3 a(X, Y, Z, X):-$

$$X \geq Y, X \geq Z.$$

/\* если первое число больше или равно второму  
и третьему, то первое число - максимум \*/

$\max_3 a(X, Y, Z, Y):-$

$$Y \geq X, Y \geq Z.$$

/\* если второе число больше или равно первому  
и третьему, то второе число является  
максимумом \*/

$\max_3 a(X, Y, Z, Z):-$

$$Z \geq X, Z \geq Y.$$

/\* если третье число больше или равно первому  
и второму, то максимум - это третье число \*/



# Нахождение максимума из трех чисел (отсечение)

$\text{max3b}(X, Y, Z, X):-$

$X > Y, X > Z, !.$

/\* если первое число больше второго и третьего,  
то первое число - максимум \*/

$\text{max3b}(\_, Y, Z, Y):-$

$Y \geq Z, !.$

/\* иначе, если второе число больше третьего,  
то второе число является максимумом \*/

$\text{max3b}(\_, \_, Z, Z).$

/\* иначе максимум - это третье число \*/



# Нахождение максимума из трех чисел (с помощью max2)

$\text{max3}(X, Y, Z, M)$ :-

$\text{max2}(X, Y, XY)$ , /\* XY - максимум из X и Y \*/

$\text{max2}(XY, Z, M)$ . /\* M - максимум из XY и Z \*/

# Рекурсия на Прологе





# Программа «Родственники»

предок(Предок,Потомок):-

родитель(Предок,Потомок).

/\* предком является родитель \*/

предок(Предок,Потомок):-

родитель(Предок,Человек),

предок(Человек,Потомок).

/\* предком является родитель предка \*/



# Правило, реализующее шаг рекурсии

<имя определяемого предиката>:-  
[<подцели>],  
[<условие выхода из рекурсии>],  
[<подцели>],  
<имя определяемого предиката>,  
[<подцели>].



# Программа «Факториал»



$1! = 1$  /\* факториал единицы равен единице \*/  
 $N! = (N-1)! * N$  /\* для того, чтобы вычислить факториал некоторого числа, нужно вычислить факториал числа на единицу меньшего и умножить его на исходное число \*/

# Факториал



`fact(1,1).` /\* факториал единицы равен единице \*/

`fact(N,F):-`

`N1=N-1,`

`fact(N1,F1),` /\* F1 равен факториалу числа  
на единицу меньшего исходного  
числа \*/

`F=F1*N.` /\* факториал исходного числа равен  
произведению F1 на само число \*/

# Факториал



`fact(1,1).` /\* факториал единицы равен единице \*/

`fact(N,F):-`

`N>1,` /\* убедимся, что число больше единицы \*/

`N1=N-1,`

`fact(N1,F1),` /\* F1 равен факториалу числа,  
на единицу меньшего исходного  
числа \*/

`F=F1*N.` /\* факториал исходного числа равен  
произведению F1 на само число \*/

# Факториал

`fact(1,1):-!. /* условие останова рекурсии */`

`fact(N,F):-`

`N1=N-1,`

`fact(N1,F1), /* F1 равен факториалу числа,  
                  на единицу меньшего исходного  
                  числа */`

`F=F1*N. /* факториал исходного числа равен  
          произведению F1 на само число */`

# Факториал

## Правосторонняя рекурсия

$\text{fact2}(N,F,N,F):-!$ . /\* останавливаем рекурсию, когда третий аргумент равен первому\*/

$\text{fact2}(N,F,N1,F1):-$

$N2=N1+1$ , /\*  $N2$  - следующее натуральное число после числа  $N1$  \*/

$F2=F1*N2$ , /\*  $F2$  - факториал  $N2$  \*/

$\text{fact2}(N,F,N2,F2)$ .

/\* рекурсивный вызов с новым натуральным числом  $N2$  и соответствующим ему посчитанным факториалом  $F2$  \*/

# Факториал



factM(N,F):-

fact2(N,F,1,1). /\* вызываем предикат с уже  
заданными начальными  
значениями \*/

# Цикл с предусловием



W :-

<условие>, p, w.

w :- !.

while <условие> do P



# Программа «Родственники» левосторонняя рекурсия

предок2(Предок,Потомок):-

родитель(Предок,Потомок).

/\* предком является родитель \*/

предок2(Предок,Потомок):-

предок2(Человек,Потомок),

/\* предком является родитель предка \*/

родитель(Предок,Человек).