

## Лекция 6

# Логические задачи

# Задача 1 (Зебра)

Пять человек различной национальности живут в пяти первых домах одной из улиц. Они являются представителями пяти различных профессий, и у каждого из них есть свои любимые напитки и животные, среди которых нет повторяющихся. Дома, в которых они живут, покрашены в различные цвета. Известно, что:

- Англичанин живет в красном доме.
- У испанца есть собака.
- Японец является художником.
- Итальянец пьет чай.
- Норвежец живет в первом доме слева.
- Владелец зеленого дома пьет кофе.
- Зеленый дом находится справа от белого.
- Скульптор разводит улиток.
- Дипломат живет в желтом доме.
- В доме посередине пьют молоко.
- Норвежец живет рядом с голубым домом.
- Скрипач пьет фруктовые соки.
- Лису держат в доме, соседнем с домом врача.
- Лошадь держат по соседству с домом дипломата.

Кто держит зебру, и кто пьет воду?

# Решение

Улица - список домов:

*[нац, проф, цвет, животное, напиток],[...],[...],\_,\_]*

*Предикат «справа»:*

*next([X1,X2|\_],X1,X2).*

*next([\_|T],X1,X2):- next(T,X1,X2).*

*Предикат «по соседству»:*

*neighbour(Street,X1,X2):-next(Street,X1,X2).*

*neighbour(Street,X1,X2):-next(Street,X2,X1).*

# Предикат solve

`solve(Street,Z,W):-`

```
Street = [[norwegian,_,_,_],[_,_,blue,_,_],[_,_,_,_,milk],_,_],  
member([english,_,red,_,_],Street),  
member([spanish,_,_,dog,_,_],Street),  
member([japanese,painter,_,_,_],Street),  
member([italian,_,_,_,tea],Street),  
member([_,_,green,_,_,coffee],Street),  
next(Street,[_,_,white,_,_],[_,_,green,_,_]),  
member([_,_,sculptor,_,_,snails,_,_],Street),  
member([_,_,diplomat,yellow,_,_],Street),  
member([_,_,musician,_,_,_,juice],Street),  
neighbour(Street,[_,_,_,fox,_,_],[_,_,doctor,_,_,_]),  
neighbour(Street,[_,_,_,horse,_,_],[_,_,diplomat,_,_,_]),  
member([Z,_,_,zebra,_,_],Street),  
member([W,_,_,_,water],Street).
```

# Запросы

?- solve(X,Z,W).

X = [ [norwegian,diplomat,yellow,fox,water],  
[italian,doctor,blue,horse,tea],  
[english,sculptor,red,snails,milk],  
[spanish,musician,white,dog,juice],  
[japanese,painter,green,zebra,coffee] ]

Z = japanese

W = norwegian

## Пример 2 (Слесарь)

На одном заводе работали три друга: слесарь, токарь и сварщик. Их фамилии Борисов, Иванов и Семенов. У слесаря нет ни братьев, ни сестер. Он - самый младший из друзей. Семенов, женатый на сестре Борисова, старше токаря. Назовите фамилии слесаря, токаря и сварщика.

# Решение (1)

```
solve(B,I,S) :- permute([B,I,S],[slesar,tokar,svarshik]),not(contradiction([B,I,S])).
```

```
contradiction(V):- data(V,F,A,TF), logicalnot(TF,FT), data(V,F,A,FT).
```

```
logicalnot(true,false).
```

```
logicalnot(false,true).
```

```
data(_sister,[slesar],false).
```

```
data(_old,[tokar,slesar],true).
```

```
data(_old,[slesar,tokar],false).
```

```
data(_old,[svarshik,slesar],true).
```

```
data(_old,[slesar,svarshik],false).
```

```
data([B,_],sister,[B],true).
```

```
data([_,_S],old,[S,tokar],true).
```

```
data([_,_S],old,[tokar,S],false).
```

# Решение (2)

```
solve(B,I,S) :- permute([B,I,S],[slesar,tokar,svarshik]),not(contradiction([B,I,S])).
```

```
contradiction(V):- data(V,F,A,fact,TF), logicalnot(TF,FT), data(V,F,A,_,FT).
```

```
logicalnot(true,false).
```

```
logicalnot(false,true).
```

```
data(_,sister,[slesar],fact,false).
```

```
data(_,old,[tokar,slesar],fact,true).
```

```
data(_,old,[svarshik,slesar],fact,true).
```

```
data([B,_,_],sister,[B],fact,true).
```

```
data([_,_,S],old,[S,tokar],fact,true).
```

```
data(V,old,[X,Y],rule,TF):-
```

```
    data(V,old,[Y,X],fact,FT),
```

```
    logicalnot(FT,TF).
```



# Запросы

```
?-solve(B,I,S), write('Борисов-'), write(B), nl,  
    write('Иванов-'), write(I), nl,  
    write('Семенов-'), write(S).
```

Борисов- tokar

Иванов- slesar

Семенов- svarshik

### Пример 3 (Домино)

Алла, Галя, Лена и Марина играли в домино. Марина младше, чем Галя. Лена старше, чем любая из ее противниц. Марина старше, чем ее партнерша. Алле и Гале вдвоем больше лет, чем Лене и Марине вместе.

Кто с кем играл, как распределить девушек по возрасту?

# Решение

Пары: Лена – X  
Y – Z

Список возрастов (от старшего к младшему): Age = [X1,X2,X3,X4]

Предикат «старше»:

more(X,Y,[X|T]):-member(Y,T).

more(X,Y,[\_ | T]):-member(Y,T),member(X,T),more(X,Y,T).

```
solve :- Age = [X1,X2,X3,X4], permute(Age,[alla,galia,lena,marina]),
remove(X,[alla,galia,marina],[Y,Z]),
more(galia,marina,Age),
more(lena,Y,Age), more(lena,Z,Age),
member(Pair,[[lena,X],[Y,Z]]), remove(marina,Pair,[MP]),
more(marina,MP,Age),
not(permute([alla,galia],[X3,X4])), not(permute([alla,galia],[X2,X4])),
write('lena-'),write(X),nl,
write(Y),write('-'),write(Z),nl,
write(Age).
```

# Запросы

?- solve.

lena-galia

alla-marina

[galia,lena,marina,alla]

# Рекурсия

Примеры:

*Печать списка:*

1) `writelist([]).`

`writelist([H|T]) :- write(H), nl, writelist(T).`

2) через `member` (`remove`, `permute`, `for`)

`print :- member(X,[1,2,3,4,5]),write(X),fail.`

`print:- for(X,1,5),write(X),fail.`

# Циклы

Цикл **for**:

for(A,A,\_).

for(X,A,B):-A<B, A1 is A+1, for (X, A1,B).

Цикл **repeat**:

repeat.

repeat:-repeat.

?-repeat, read(X), 0 is X mod 5, write(X).

*Возврат к repeat до тех пор, пока условие не станет истинным*