СТРУКТУРА ОТЧЕТА по лабораторной работе

1. Представление программы в виде дерева.

Задана программа в виде 2-х правил.

P : - Q, R.

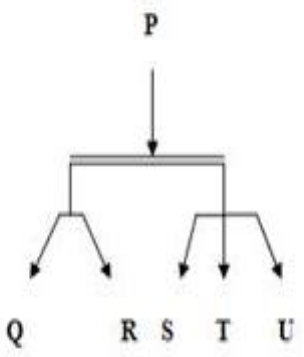
P : - S, T, U.

Напомним, что запятая между целями обозначает *конъюнкцию* целей.

Вершина типа ИЛИ обозначается двумя линиями.

Вершина типа И обозначается одной линией.

Структура дерева будет выглядеть так:



Дизъюнкция обозначается точкой с запятой. Например:

P :- Q; R.

читается так: P - истинно, если истинно Q *или* истинно R. То есть смысл такого предложения тот же, что и смысл следующей пары предложений:

P :- Q.
P :- R.

2. Структура предикатов

domains

list = integer* - список содержит целые числа

предикаты

member(integer, list) - 1-ый аргумент «число», 2-ой – «список»

intersect(list, list, list) – все три аргумента – списки.

текст программы

```
member(X, [X|_]).
member(X, [_|Y]) :- member(X, Y).
intersect([], [], _).
intersect([X|Y], [X|L1], L2) :- member(X, L2),!, intersect(Y, L1, L2).
intersect(Y, [_|L1], L2) :-  intersect(Y, L1, L2).
```

3. Пример выполнения программы в режиме трассировки.

```
listlength([],A,A).
listlength([_|T],A,O):-
                    A1 is A+1,
                    listlength(T,A1,O).
```

?- listlength([a,b,c],0,N).

```
    1    1 Call: listlength([a,b,c],0,_501)
    2    2 Call: _1096 is 0+1
    2    2 Exit: 1 is 0+1
    3    2 Call: listlength([b,c],1,_501)
    4    3 Call: _2817 is 1+1
    4    3 Exit: 2 is 1+1
    5    3 Call: listlength([c],2,_501)
    6    4 Call: _4538 is 2+1
    6    4 Exit: 3 is 2+1
    7    4 Call: listlength([],3,_501)
    7    4 Exit: listlength([],3,3)
    5    3 Exit: listlength([c],2,3)
    3    2 Exit: listlength([b,c],1,3)
    1    1 Exit: listlength([a,b,c],0,3)
N = 3
yes
```

4. Рекурсивная схема обработки подцелей

Обработка подцелей в правилах

В качестве подцелей могут быть:

- факты,
- другие правила,
- тоже самое правило = рекурсивный вызов

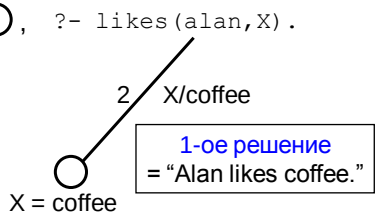
```
1) drinks(alan,water).
2) likes(alan,coffee).
3) likes(heather,coffee).

4) likes(Person,Drink):-
        drinks(Person,Drink).   ← другая подцель
5) likes(Person,Somebody):-
        likes(Person,Drink),    ← рекурсивная
        likes(Somebody,Drink).  ← подцель
```

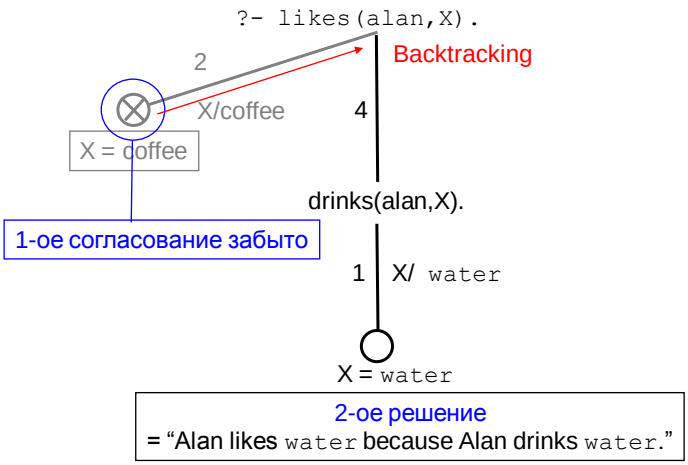
Представление доказательства цели в виде дерева

Будем использовать AND/OR деревья.

1. Запрос представляется корнем дерева.
2. Дерево растет вниз.
3. Каждая ветвь обозначает подцель.
   1. Каждая ветвь помечается номером согласующейся клаузы и
   2. Значениями переменных, полученных при унификации.
4. Каждый конец ветви помечаем:
   1. Успешное согласование ◯,
   2. Неудача ⊗,
   3. Другая подцель.

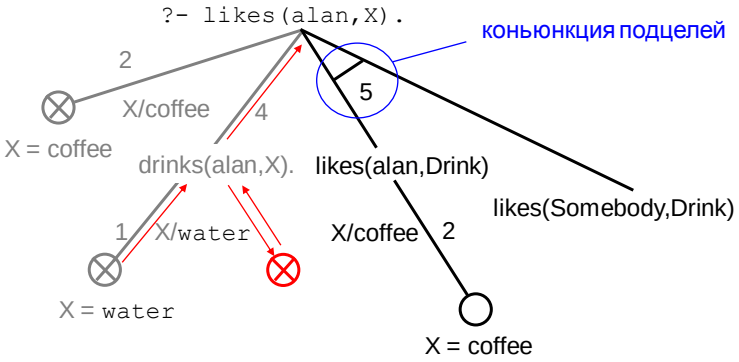


Представление доказательства цели в виде дерева

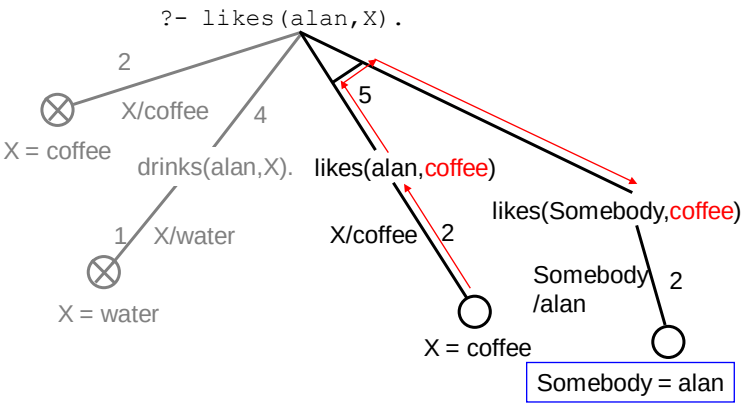


Рекурсия

Когда предикат вызывает самого себя внутри правила, мы называем такой вызов рекурсивным.



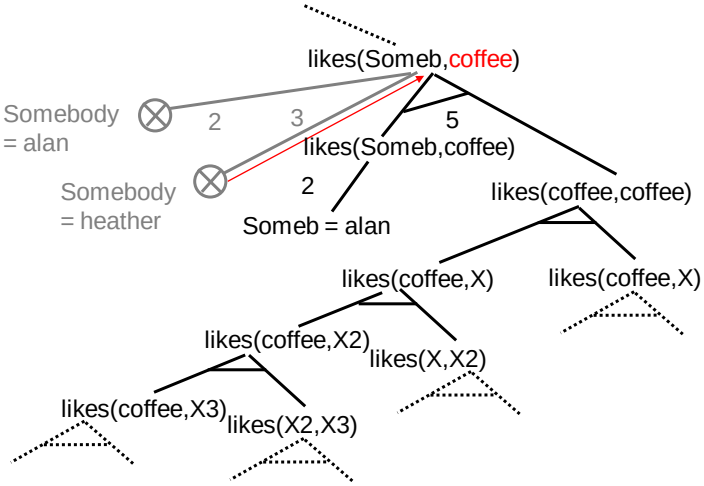
Рекурсия



3е решение = “Alan likes Alan because Alan likes coffee.”

Бесконечная рекурсия

- Если рекурсивный вызов некорректен – возможно зацикливание.



5. Выводы по работе.