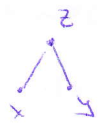
ЛП
22.II.

$p(X, Y)$ - X родител на Y

//брат и сестра $бс(X, Y) \Leftarrow \exists z(p(z,x) \wedge p(z,y)) \wedge \neg x = y$



$Внук(X, Y) \Leftarrow \exists z(p(z,x) \wedge p(y,z))$

PROLOG
Syntax

I Клаузи

1. Факти
   p(T1, ..., Tn).   // T_1 до T_n са в релацията p

2 Правила
   $\underbrace{p(*arguments*) \; \underbrace{:-}_{ако} \; \underbrace{p_1(*arg*), \ldots, p_n(*arg*)}_{условия}.}$

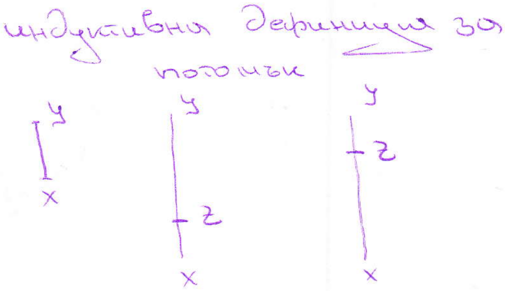
3. Цели
   ?- p1(*arg*), ..., pn(*arg*).

Деф: Термове:
- променливи - X, P (uppercase)
- константи - m, peter (lower case)
- $f(T_1, \ldots, T_n)$ T_i - термове $i \in \overline{1,n}$

$\underbrace{1}_{const} \underbrace{+}_{функционален\ символ} \underbrace{1}_{const}$

```
p(ivan, maria).
p(maria, ana).
n(X,Y) :- p(Y,X).
n(X,Y) :- p(Z,X), n(Z,Y).
    // ∃z( p(z,x) ∧ n(z,y))
```

потомък




```
?- n(maria, ivan).
?- n(X, ivan).                   // Ще изведе true
?- n(X, ivan), write(X).         // Ще изведе стойността на X

?- n(X,Y), write(X), write(Y), nl, fail.
```
nl - нов ред

Производни:

```
d(x,1).
d(x,0):-number(x).
d(x+y, Dx+Dy):-d(x,Dx), d(y,Dy).
d(x*y, Dx*y+x*Dy):-d(x,Dx), d(y,Dy).
d(sin(x), cos(x)*Dx):-d(x,Dx).

?-d(sin(x*x),D).
```

↳ = cos(x*x) 1*x+x*1 Missing ()

f - празен списък
S(H,T)
↓ ↳ tail
head

```
add(X, L, S(X, L)). // Добавя X пред L и получаваме S
remove(X, L, N):-add(X, N, L).        - правило
remove(X, S(X, N), N).                - факт

member(X, S(X, _)).                   '_' анонимна
member(X, S(_, T)):-member(X, T).         променлива

?-member(X, S(1, S(2, f)))
```

Л-П
упр. 01.III

Списъци

```
[ ]
[H|T]

[a,b,c] = [a|[b|[c|[]]]]

H=a
T=b,c

[A,B|T]
```

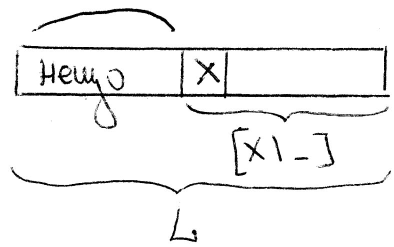
```
% first(F, L)

?- first(1, [1,2])     true
?- first(3, [1,2])     false
?- first(X, [1,2])     true X=1

first(F, [F|_]).

second(S, [_,S|_]).
second(S, [_|T]):- first(S,T).
```

```
% last(X,L)

last(X, [X]).          = last(X, [X|[]]).
last(X, [_|T]:- last(X,T).

last(X, [_|T]):- last(X,T).
last(X, [X]).
```

```
% member(X, L).
member(X, [X|_]).
member(X, [_|T]):- member(X,T)
```

```
append(A,B,AB)
append([], B, B).
append([H|A], B, [H|AB]):- append(A,B,AB).
```

```
last(X, L) :- append(_, [X], L).
member(X, L) :- append(_, [X|_], L).
```




```
% remove(X, L, N).

remove(X, [X|T], T).
remove(X, [H|T], [H|N]) :- remove(X, T, N).

member(X, L) :- remove(X, L, _).
```

```
% add(X, L, N)
add(X, L, N) :- remove(X, N, L).
```

```
perm([], []).
perm([H|P], L) :- member(H, L), remove(H, L, N), perm(P, N).
```

може и без member

```
sorted([]).
sorted([H|T]) :- first(F, T), less(H, F), sorted(T).

[A, B|T] :- less(A, B), sorted([B, T]).
```

```
not(p())

sorted(L) :- not(append(_, [A, B|_], L), not(less(A, B))).
```

X е списък от числа

Y е списък от списъци от числа.

$p_1(X, Y)$ - ∃ елемент на X, който е в елемент на Y

$p_2(X, Y)$ - ∃ елемент на X, който е във всеки елемент на Y

$p_3(X, Y)$ - Всеки елемент на X е в някой елемент на Y.

$p_4(X, Y)$ - Всеки елемент на X е във всеки елемент на Y

Решение

p1(X, Y) :- member(A, X), member(B, Y), member(A, B).

I - $\exists a \exists b \varphi$

$\forall b \equiv \neg \exists b \neg$

$\neg \forall b \varphi \equiv \exists b \neg \varphi$

II - $\exists a \forall b \varphi \equiv \exists a \neg \exists b \neg \varphi$

действие

$\neg\neg \forall b \equiv \forall b$

p2(X; Y) :- member(A, X), not(member(B, Y), not(member(A, B))).

p3(X, Y) :- not(member(A, X), not(member(B, Y), member(A, B))).

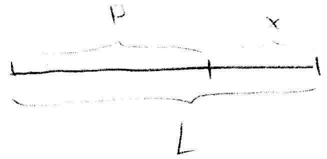
III. $\forall a \exists b \varphi \equiv \neg \exists a \neg \exists b \varphi$

IV - $\forall a \forall b \varphi \equiv \neg \exists a \neg\neg \exists b \neg \varphi \equiv \neg \exists a \exists b \neg \varphi$

p4(X, Y) :- not(member(A, X), member(B, Y), not(member(A, B))).

AM
08.III.

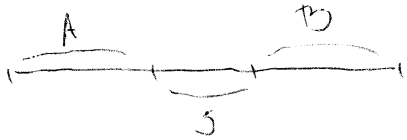
```
% prefix(P, L)
prefix(P, L) :- append(P, _, L).
```



$\exists x$ append(P, X, L)

```
suffix(S, L) :- append(_, S, L).
```

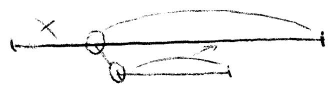
```
sublist(S, L) :-
    append(_, C, L),
    append(S, _, C).
```




```
?- sublist(S, [a, b, c]).
```

```
// sublist(S, L) :- append(S, _, C), append(_, C, L).
//   Ще има безброй решения, защото не знаем C
```

```
% subset(S, L)
subset([], _)
subset(S, [H|T]) :- subset(S, T).
subset([H|S], [H|T]) :- subset(S, T).
```




```
subset([], _)
subset([H|T], L) :- suffix([H|S], L), subset(T, S).
```

% issubset(S, L)

$\forall x (x \in S \to x \in L) \equiv \neg \exists x \neg (\neg x \in S \vee x \in L) \equiv$
$\neg \exists x (x \in S \wedge \neg x \in L)$

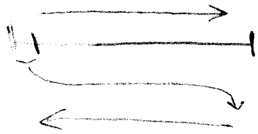
```
issubset(S, L) :- not(member(X, S), not(member(X, L))).
```

```
equal(S, L) :- issubset(S, L), issubset(L, S).
```

```
intersection(X, A, B) :- member(X, A), member(X, B).
union(X, A, B) :- member(X, A).
union(X, A, B) :- member(X, B).
```

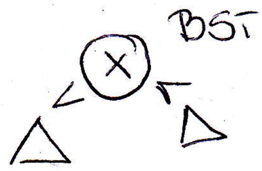
A\B

```
sub(X, A, B) :- member(X, A), not(member(X, B)).
```



```
%reverse(R,L).

reverse([],[]).
reverse(R, [H|L]):- reverse(RL,L), append(RL,[H],R).


reverse1(S,[],S).
reverse1(S,[H|T],R):- r([H|S],T,R).

reverse(R,L):- reverse1([],L,R).

//

%min(M,L).
min(M,[M]).
min(M,[H|T]):- min(N,T), min2(M,H,N).

min2(A,A,B):- less(A,B).
min2(B,A,B):- not(less(A,B)).

//

ssort([],[])
ssort([M|S],L):- min(M,L), remove(M,L,T), ssort(S,T).

//

%Quicusort
%split(L,X,A,B)


split([],X,[],[]).
split([H|T],X,[H|A],B):- H<X , split(T,X,A,B)
split([H|T],X,A,[H|B]):- not( <X), split(T,X,A,B).

qsort([],[]).
qsort(S,[X|T]):- split(T,X,A,B), qsort(SA,A),
                                 qsort(SB,B),
                                 append(SA,[X|SB],S).
```

ЛМР
упр. 15.III



% сортиране на списък, using BST

дърво: [T, L, R]
дясно поддърво
корен   ляво поддърво

```
% tsort(L, S).
% maketree(L, T).
% insert(X, T, N).
% ltr(T, S)              // ляво-корен-дясно

maketree([], []).
maketree([H|L], T) :- maketree(L, T1), insert(H, T1, T).

insert(X, [], [X, [], []]).
insert(X, [T, L, R], [T, L1, R]) :- X ≤ T, insert(X, L, L1).
insert(X, [T, L, R], [T, L, R1]) :- not(X ≤ T), insert(X, R, R1).

ltr([], []).
ltr([T, L, R], S) :- ltr(L, L1), ltr(R, R1),
                     append(L1, [T|R1], S).

tsort(L, S) :- maketree(L, T), ltr(T, S).
```

Аритметика

X is Y+Z
- ако X има стойност и
  - X == Y+Z => true
  - X \= Y+Z => false
- ако X няма стойност, приема стойността на Y+Z

X is X+1 - always false
- ако X има ~~стойност~~ стойност, X \= X+1
- ако X няма стойност, не може да се изпълни

```
% length(L, N)
length([], 0).
length([H|L], N) :- length(L, M), N is M+1     // Редът е важен
                                                  ~~M is N-1~~

% sum(L, S).
sum([], 0).
sum([H|T], N) :- sum(T, M), N is M+H
```

% member (X, N, L)
↳ nth element

```
member(X, 0, [X|_]).
member(X, N, [H|T]) :- member(X, M, T), N is M+1.

member(X, N, [H|T]) :- M is N-1, member(X, M, T).
```
↑ няма да работи в определени случаи

```
?- member(1, 0, [1,2,3]).
?- member(X, 2, [1,2,3]).
?- member(X, N, [1,2,3]).
?- member(2, N, [1,2,3]).
```

Генератори

- естествени числа

```
n(0).
n(X) :- n(X1), X is X1+1.          // 0, 1, 2, etc
```

- 1, 3, 5, 7 ....

```
n(1)
n(X) :- n(X1), X is X1+2.
```

- Фибоначи

експоненциална сложност

```
f(0, 0).
f(1, 1).
f(N, X) :- N1 is N-1, N2 is N-2, f(N1, Y), f(N2, Z), X is Y+Z
```

```
f(0, 1).
f(Y, Z) :- f(X, Y), Z is X+Y
f(X) :- f(X, _)
```

```
f(n, x, y) {
  if (n == 0) return x;
  return f(n-1, y, x+y);
}
f(n) { f(n, 0, 1); }
```

- $a_0 = a_1 = a_2 = 1$

$a_{n+3} = 3a_{n+2} + 2a_n$

```
p(1, 1, 1).
p(X, Y, Z) :- p(A, X, Y), Z is 3*Y + 2*A.
p(X) :- p(X, _, _).
```

- 0, 1, -1, 2, -2, ...   ← h.w.

ЛПР

упр. ~~28~~.III.

- Генератор на цели числа
  0, 1, -1, 2, -2, ...

```
z(Y):- n(X), s(X,Y).

s(X,X).
s(X,Y):- Y>0, Y is -X.
```

```
z(X):- n(Y), decode(Y,X).

decode(X,Y):- X mod 2 =:= 0,
              Y is -X//2.
decode(X,Y):- X mod 2 =:= 1,
              Y is X//2+1.
                   (X+1)//2
```

0	1	2	3	4	5
0	1	-1	2	-2	3

2k → -k
2k+1 → k+1

- Генериране на естествените числа ≤ 10.

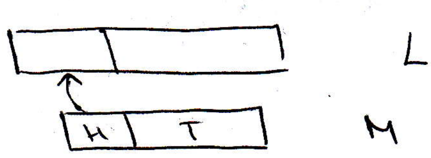
```
% n10(X).
n10(X):- n(X), X<10      // ще даде числата до 9 и ще
                         // зацикли.

% between(X,A,B).
between(A,A,B):- A=<B.
between(X,A,B):- A1 is A+1, between(X,A1,B).
               A<B
```

```
n(X,Y):- n(N), between(X,0,N), Y is N-X.
```

```
% sums(N,L).
% splits(L,M).
% concat(M)=L
   [1,1,1]
```

[[1], [1], [1]]
[[1], [1,1]]
[[1,1], [1]]
[[1,1,1]]

```
sums(0,[]).
sums(N,[X|T]):- between(X,1,N), Y is N-X, sums(Y,T).

splits([],[]).
splits(L,[H|T]):-
```



append (H, S, L), splits (S, T).

H1 = []

% p(x), генерира в X ∈ N; X е сума от квадратите на 4 естествени числа, X<1000.

```
p(x) :- between (X1, 0, 31),
        between (X2, 0, 31),
        between (X3, 0, 31),
        between (X4, 0, 31),
        X is X1*X1 + X2*X2 + X3*X3 + X4*X4, X<1000.
```

```
p(x): - between (X, 0, 999).
```

Теорема: Всяко естествено число може да се представи като сума от квадратите на 4 естествени числа.

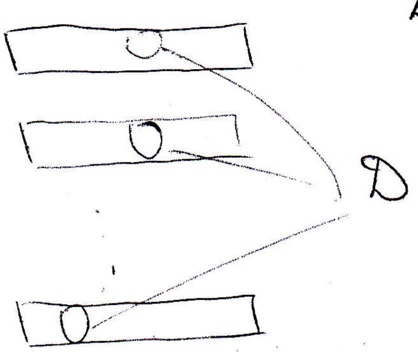
% p(X1, Y1, X2, Y2, X3, Y3) x_i, y_i - координати на точки, образуващи Δ, с прав ъгъл при X2, Y2

генерира 6-торки цели числа

```
n6(Y1, ..., Y6):-
    n(X), sum(X, [X1, X2, X3, X4, X5, X6]),
        s1(X1, Y1), s1(X2, Y2), ..., s1(X6, Y6).

s1(X, Y) :- Y is X-1.
s1(X, Y) :- X>1, Y is -(X-1).
```

% проверка дали тези 6 числа образуват Δ.

Задача   % d(D, L).   L - списък от списъци
D - декартово произведение на елементите от X.



```
d([], []).
d([X|D], [H|T]):- member(X, H), d(D, T).
```

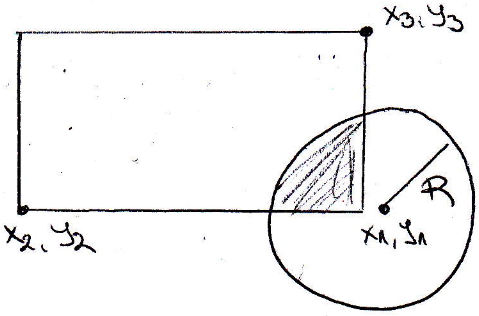
ЛПР

упр. 29.III

Задача

Да се напише предикат, генериращ всички общи точки за даден правоъгълник (X2,Y2,X3,Y3) и окръжност (X1,Y1,R).

% p(X,Y, X1,Y1, R, X2,Y2, X3,Y3)




```
inside(X,Y, X2,Y2, X3,Y3):-
    between(X, X2, Y),
    between(Y, Y2, Y3).

incircle(X,Y, X1,Y1,R):-
    (X-X1)^2 + (Y-Y1)^2 =< R^2.

p(X,Y, X1,Y1,R, X2,Y2, X3,Y3):- inside(X,Y, X2,Y2, X3,Y3),
                                 incircle(X,Y, X1,Y1,R).
```

Задача

- [] е дърво
- Ако A и B са дървета, то [A,B] е дърво. (двоични).

Да се напише предикат t(T), генериращ всички двоични дървета.

Р-ие:

```
t(T):- n(N), t(T,N).
t([],0).
t([A,B],N):- N>0, N1 is N-1,
             between(NA, 0, N1), NB is N1-NA,
             t(A,NA), t(B,NB).
```

Задача ? s(M,L) - в M генерираме всички списъци с елементи от L (може и да се повтарят)

Пример: L=[1,2]
M=[1,1,1,2].
M=[2,1,2,1,2,1].

Р-ие: 1)

```
s([],L).
s([H|T],L):- s(T,L), member(H,L).
```

2)

```
s(M,L):- n(N), s(N,M,L).
s(0,[],L).
s(N,[H|T],L):- N>0, member(H,L), N1 is N-1,
               s(N1,T,L).
```

% G - граф   G = (V, E

% edge(G, X,Y)

% path(G,X,Y,P)

% connected(G)

% stree(G,T)   //покриващо_дърво

```
edge([V,E], X, Y) :- member([X,Y], E); member([Y,X], E).

path(G,X,Y,P) :- path(G,X,Y,[],Q), reverse(Q,P).
path(G,Y,Y,V,[Y|V]).
path(G,X,Y,V,P) :- edge(X,Z),
                   not(member(Z,V)),
                   path(G,Z,Y,[X|V],P).
```

за посетени вече върхове, за да не се получи зацикляне

1)
```
connected([V,E]) :- not(member(X,V), member(Y,V),
                        not(path([V,E],X,Y,_))).
```

//Не съществуват 2 върха между, които няма път.

2)
```
connected([[],[]]).
connected([[X|V],E]) :- not(member(Y,V),
                            not(path([[X|V],E],X,Y,_))).
```

```
% stree(G,T)      G = [V,E] , T = [V,E1]

st(G,V,[],[]).
st(G,V,N,[[X,Y]|R]) :- edge(G,X,Y), member(X,V), member(Y,N),
                       remove(Y,N,N1),
                       st(G,[Y|V],N1,R).
```

посетени

Непосетени

```
stree([V,E],[V,E1]) :- V=[X|V1],
                       st([V,E],[X],V1,E1).
```

ЛПР
05.IV

Задача: [[[1, [2]], 1], [2]] → [1, 2, 1, 2]

```
flat([], []).
flat([H|T], F) :- list(H), flat(H, FH), flat(T, FT), append(FH, FT, F).

list([]).
list([_|_]).
```

Задача: c(L, C) [1, 2, 2, 3, 3, 4, 4, 4, 5]
[[1], [2, 2], [3, 3], [4, 4, 4], [5]].

```
c([], []).
c([A], [[A]]).
c([A, A|T], [[A|C]|R]) :- c([A|T], [C|R]).
c([B, A|T], [[B]|R]) :- A\=B, c([A|T], R).
```

Задача: p(A, B) - дали A и B имат 1 и същи прости делители

$\forall x (pd(x, A) \to d(x, B))$
$\neg \exists x \neg(\neg pd(x, A) \vee d(x, B))$.
$\neg \exists x (pd(x, A) \wedge \neg d(x, B))$.

```
d(A, B) :- B mod A =:= 0.
prime(X) :- X>1, X1 is X-1, not(between(Y, 2, X1), d(Y, X)).
pd(X, A) :- between(X, 2, A), prime(X), d(X, A).
psubset(A, B) :- not(pd(X, A), not(d(X, B))).

p(A, B) :- psubset(A, B), psubset(B, A).
```

Задача: p(L) генерира всички списъци от списъци $[X_1, \ldots X_n]$, където X_i е списък с елементи от 0 до 99 и X_i е префикс на X_{i+1}.

```
%q(N, L) - N-елементен списък с елементи от 0 до 99
q(0, []).
q(N, [H|T]) :- N>0, N1 is N-1, between(H, 0, 99), q(N1, T).

% s(X, N, L) - N-елементен списък L, с последен елемент X.

s(X, 1, [X]).
s(X, N, [H|T]) :- N>1, N1 is N-1, s(X, N1, T), T=[Y|_], prefix(H, Y).

p(L) :- n(M, N), q(M, X), s(X, N, L).
```

ЛПР
Р. 12.IV

% q(N, X, K)

q(0, X, 0)
q(N, X, K) :- member(Y, X), Y<N, M is N-Y, q(M, X, K1), K is K1+1.

q(N, X) :- member(N, X), q(N, X, K), K>1.

% q(N, X) - намира елементите N на X, които са сума на няколко, не задължително различни, елемента на X.

$Var = \langle x_1, \dots, x_n, y, z, \dots \rangle$
$\mathbb{F}unc$
$\mathbb{P}red$
Const

def: Терм
- променлива
- $f \in \mathbb{F}unc$ (n-местен), а $\tau_1, \dots, \tau_n$ са термове, то $f(\tau_1, \dots, \tau_n)$ е терм.

def: Атомарна ф-ла
- $\tau_1 = \tau_2$ τ_i - терм, $i = 1, 2$
- $p \in \mathbb{P}red$ - n-местен, $\tau_1, \dots, \tau_n$ - термове, то $p(\tau_1, \dots, \tau_n)$ е атомарна ф-ла.

def: Формули
- атомарни ф-ли
- φ и ψ - ф-ли, тогава:
  - $\neg\varphi$
  - $\varphi \vee \psi$
  - $\varphi \wedge \psi$
  - $\varphi \Rightarrow \psi$
  - $\varphi \Leftrightarrow \psi$
  - $\forall x \varphi$
  - $\exists x \varphi$

  също са формули.

Унарните $(\neg\varphi, \exists x\varphi, \forall x\varphi)$ са с най-голям приоритет

$[(p(x) \wedge p(y)) \rightarrow p(z)]$

def: Структура
$A = \langle U, I \rangle$
$U \neq \emptyset$ (!), $v: Var \rightarrow U$

- $\mathcal{I}(c) = c^{\mathcal{A}} \in \mathcal{U}$
- $\mathcal{I}(f) = f^{\mathcal{A}} : \mathcal{U}^n \to \mathcal{U}$
- $\mathcal{I}(p) : p^{\mathcal{A}} \subseteq \mathcal{U}^n$

$\|x\|_{\mathcal{A}}^{v}$ // структура $\mathcal{A}$ и оценка v

- $\|x\|_{\mathcal{A}}^{v} = v(x)$
- $\|c\|_{\mathcal{A}}^{v} = c^{\mathcal{A}}$
- $\|f(\tau_1, \dots, \tau_n)\|_{\mathcal{A}}^{v} = f^{\mathcal{A}}(\|\tau_1\|_{\mathcal{A}}^{v}, \dots, \|\tau_n\|_{\mathcal{A}}^{v})$
- $\|\varphi\|_{\mathcal{A}}^{v} = \mathcal{U} \quad \equiv \quad \mathcal{A} \models_v \varphi \equiv \mathcal{A}, v \models \varphi$
  модел за φ
- $\mathcal{A} \models_v \tau_1 \doteq \tau_2$ ТСТК $\|\tau_1\|_{\mathcal{A}}^{v} = \|\tau_2\|_{\mathcal{A}}^{v}$

- $\mathcal{A} \models_v p(\tau_1, \dots, \tau_n)$ ТСТК $\langle \|\tau_1\|_{\mathcal{A}}^{v}, \dots, \|\tau_n\|_{\mathcal{A}}^{v} \rangle \in p^{\mathcal{A}}$
- $\mathcal{A} \models_v \neg\varphi$ ТСТК $\mathcal{A} \not\models_v \varphi$
- $\mathcal{A} \models_v \varphi \wedge \psi$ ТСТК $\mathcal{A} \models_v \varphi$ и $\mathcal{A} \models_v \psi$
- $\mathcal{A} \models_v \varphi \vee \psi$ $\mathcal{A} \models_v \varphi$ или $\mathcal{A} \models_v \psi$
- $\mathcal{A} \models_v \varphi \Rightarrow \psi$ ТСТК $\mathcal{A} \not\models_v \varphi$ или $\mathcal{A} \models_v \psi$
- $\mathcal{A} \models_v \varphi \Leftrightarrow \psi$ ТСТК $\|\varphi\|_{\mathcal{A}}^{v} = \|\psi\|_{\mathcal{A}}^{v}$ (и двете са истина или и двете са лъжа)

- $\mathcal{A} \models_v \forall x \varphi$

$$v_a^x(y) = \begin{cases} v(y), & y \neq x \\ a, & y = x \end{cases}, \quad a \in \mathcal{U}$$

- $\mathcal{A} \models_v \forall x \varphi \quad \equiv \quad \forall a \in \mathcal{U} \quad \mathcal{A} \models_{v_a^x} \varphi$

$\mathcal{U} \equiv |A|$

- $\mathcal{A} \models_v \exists x \varphi \quad \equiv \quad \exists a \in \mathcal{U} \quad \mathcal{A} \models_{v_a^x} \varphi$

Задача $\forall x\, p(x) \to p(x)$. Да се докаже, че е винаги вярно.

Р-ие: Фиксираме $\mathcal{A}, v$.

Допускаме, че $\mathcal{A}, v \not\models \forall x\, p(x) \Rightarrow p(x) \Rightarrow$

$$\underbrace{\mathcal{A}, v \models \forall x\, p(x)} \quad \text{и} \quad \underbrace{\mathcal{A}, v \not\models p(x)} \quad \Rightarrow$$

$\forall a \in |\mathcal{A}|$

$\mathcal{A}, v^x_a \models p(x)$

Нека $a = v(x)$. тогава $v^v_{v(x)} = v \Rightarrow \mathcal{A}, v^x_{v(x)} \models p(x) \equiv$

$\mathcal{A}, v \models p(x) \Rightarrow$ противоречие

Задача Да се докаже, че е винаги вярно $p(x) \to \exists x\, p(x)$

Р-ие $\mathcal{A}, v \models p(x)$ и $\underbrace{\mathcal{A}, v \not\models \exists x\, p(x)}$

не съществува $a \in |\mathcal{A}| : \mathcal{A}, v^x_a \models p(x)$

При $a = v(x)$

$\mathcal{A}, v \models p(x)$

$\equiv$

$\mathcal{A}, v^x_{v(x)}\; p(x)$, $v(x) = a$

$\Rightarrow$ противоречие

За упр: $\forall x\, (\varphi \wedge \psi) \Leftrightarrow (\forall x\, \varphi \wedge \forall x\, \psi)$

ЛПР
19.IV

- $\mathcal{A}, v \models \{\varphi\}$
- $\mathcal{A}, v \models \mathcal{E}$ - $\mathcal{E}$ е изпълнимо при структура $\mathcal{A}$ и оценка v $\mathcal{E}$ - множество от ф-ли
- $\mathcal{A} \models \mathcal{E}$ - $\mathcal{E}$ е изпълнимо при структура $\mathcal{A}$ и каквато и да е оценка
- $\models \mathcal{E}$ - $\mathcal{E}$ е изпълнимо при каквато и да е структура и и каквато и да е оценка

$p(x) \to \exists x\, p(x)$

$\mathcal{A}, v \models p(x) \to \exists x\, p(x)$

Допускаме противното:

$\underbrace{\mathcal{A}, v \models p(x)}$ и $\underbrace{\mathcal{A}, v \not\models \exists x\, p(x)}$

фиксираме $v(x) = a$ Не съществува a от A : $\mathcal{A}, v^x_a \models p(x)$

$\mathcal{A}, v^x_{v(x)} \models p(x)$

$\Rightarrow \lightning$

Задача
1) $\forall x \neg p(x,x)$
2) $\forall x \exists y\ p(x,y)$
3) $\forall x \forall y \forall z\ (p(x,y) \wedge p(y,z) \to p(x,z))$

Има ли краен модел?

Допускаме, че $\exists$ краен модел : $|A| = n$, $A = \{x_1, \dots, x_n\}$

от 2) $\Rightarrow$ $x_{i1} \to x_{i2} \to x_{i3} \to \dots x_{in} \to x_{in+1}$ // Дирихле

$x_i \dots x_j = x_i$

от 3) $\Rightarrow$ $x_i \to x_j = x_i$, което е противоречие с $\underline{1)}$

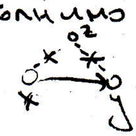
Задача $p_i(x)$ - предикати $i = \overline{1,n}$

	$p_1^{\mathcal{A}}$	$p_n^{\mathcal{A}}$
т. a	1 .. 0 ..	1
т. b	1	1

- характеристичен вектор $(\# = 2^n)$
$\Rightarrow$ краен модел (няма безкраен)

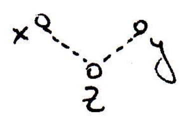
Задача Да се докаже, че м-вото от ф-ли е изпълнимо

1) $\forall x \forall y \forall z\ (p(x,y) \to \neg p(x,z) \vee \neg p(z,y))$



$\forall x \forall y \forall z \forall t\ (p(x,y) \wedge p(y,z) \wedge p(z,t) \to p(x,t))$

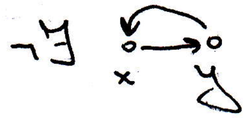
$\exists x \exists y \forall z\ (p(x,z) \vee p(y,z))$

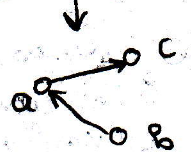


$A = \{a, b\}$
$p^A = \{\langle a, b\rangle\}$

2) $\exists x(\exists y\, p(x,y) \wedge \exists y\, p(y,x)) \equiv \exists x(\exists y\, p(x,y) \wedge \exists z\, p(x,z))$

$\neg \exists x \exists y\, (p(x,y) \wedge p(y,x))$





$A = \{a, b, c\}$
$p^A = \{\langle a, c\rangle, \langle b, a\rangle\}$

3) $\forall x \forall y\, (p(x, f(f(y))) \to \neg p(y, f(x)))$
$\forall x \forall y\, (x = y \vee p(x, f(y)) \vee p(y, f(f(x))))$
$\forall x\, (x = f(f(x)))$

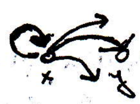
$\forall x \forall y\, (p(x,y) \to \neg p(y,x))$ $\quad f^A(a) = a$
$\forall x \forall y\, (x = y \vee p(x,y) \vee p(y,x))$

1 модел.
друг модел.

$A = \{a\}$
$p^A = \varnothing$
$f^A(a) = a$

4) $\forall x\, (p(x) \to \forall y\, p(y))$ [или всичките, или нито една] ($p^A = A$, $p^A = \varnothing$)

$\forall x \forall y\, (q(x,y) \vee \neg q(x,x))$



$\equiv \forall x \forall y\, (q(x,x) \to q(x,y))$

$\exists x\; q(x,x)$

$\exists x \exists y\; \neg q(x,y)$

$\forall x \forall y \forall z\, ((p(z) \Leftrightarrow q(x,y)) \Leftrightarrow r(x,y,z))$

$r(x,y,z)$ е И тстк $p(z) \leftrightarrow q(x,y)$

$A = \{a, b\}$

$p^A = \{a, b\}$

$q^A = \{\langle a, a\rangle, \langle a, b\rangle\}$

$r^A = \{\langle a, a, \_\rangle, \langle a, b, \_\rangle\}$

5) $\exists u \forall x \, (f(x,u) = x \wedge f(u,x) = x)$

$\forall x \forall y \forall z \, (f(x, f(z,y)) = f(f(x,z), y))$

$\exists x \exists y \, (\neg f(x,y) = f(y,x))$

$ux = xu = x$

$x(zy) = (xz)y$

$xy \neq yx$

матрици

P

ЛПР 26.IV

Определимост

$var^{free}(\varphi) \subseteq \{x_1, \dots, x_n\}$,
$a_1, \dots, a_n \in A$

$\mathcal{A} \models \varphi [\![a_1, \dots, a_n]\!]$ тстк съществува оценка v $(\forall v)$ $\mathcal{A}, v^{x_1 \dots x_n}_{a_1 \dots a_n} \models \varphi$

φ определя $\{\langle a_1, \dots, a_n \rangle \mid \mathcal{A} \models \varphi [\![a_1, \dots, a_n]\!]\}$

Пример 1) $\{0\}$
$\langle \mathbb{N}, + \rangle$

Формула, която да определя

- $\forall x\ (x + y) \doteq x$
  $v(y) = 0$ $\Rightarrow$ формулата е вярна при $y = 0$
- $x + x \doteq x$

2) $\{\langle x, y \rangle \mid x \leq y\}$ $\langle \mathbb{N}, + \rangle$

$\exists z\ x + z = y$

3) $\{\langle x, y \rangle \mid x < y\}$ $\langle \mathbb{N}, + \rangle$

$\neg \exists z\ y + z = x$

4) $\{1\}$ $\langle \mathbb{N}, + \rangle$

$\varphi_1(x) \rightarrow$ $\underbrace{\neg x = x + x}_{x \neq 0}$ $\wedge\ \forall z\ (z < x \rightarrow z = 0)$

трябва да се махнат

5) $\{2\}$ $\langle \mathbb{N}, + \rangle$

$\varphi_1(x) \wedge y = x + x$ ← грешно, има 2 свободни променливи, определя м-вото $\{\langle 1, 2 \rangle\} \subset \mathbb{N}^2$

$\forall x\ (\underbrace{\varphi_1(x)}_{\text{истина само при } x=1} \rightarrow y \doteq x + x)$ $\Rightarrow y = 2$

- При дадена структура $\langle \mathbb{N}, +, \cdot \rangle$ можем винаги да и определим м-вото $\{n\}$, $\forall n \in \mathbb{N}$

6) $\langle \mathbb{N}, \cdot \rangle$, $\{0\}$

$\forall x \quad xy = y$

7) $\langle \mathbb{N}, \cdot \rangle$, $\{1\}$

$x . x = x \wedge \underbrace{\neg \forall z \quad xz = x}_{\text{изключва нулата}}$

- Хомоморфизъм $h: \quad \to$

f - n-местен функционален символ

$f^{B}(h(a_1) \dots h(a_n)) = h(f^{A}(a_1, \dots, a_n))$

- Идентитетът е автоморфизъм (изоморфизъм в себе си)

- $R \subseteq A^n$, R - определимо и h - автоморфизъм.
Тогава $\langle a_1, \dots, a_n \rangle \in R \iff \langle h(a_1) \dots h(a_n) \rangle \in R$

Пример $\langle \mathbb{Z}, + \rangle$

- $\{0\}$ $\quad x + x = x$
- $\{1\}$ определимо ?

$h(x+y) = h(x) + h(y)$ - трябва да намерим h, което да е биекция

$h(x) = -x$

$h(h(x)) = x$

$h(\{1\}) = \{-1\}$ $\qquad \{1\} \neq \{-1\}$

$\Rightarrow$ това не е определимо м-во

Пример $\langle \mathbb{R}, \cdot \rangle$

- $\{0\}$ $\quad \forall x \quad xy = y$ $\qquad$ ✓ определимо
- $\{1\}$ $\quad \forall x \quad xy = x$

- $\{-1\}$ $\quad x . x = 1 \wedge x \neq 1 \quad (\exists y (\varphi_1(y) \wedge x . x = y \wedge \neg x = y))$

$$h(x) = \begin{cases} 0 & x = 0 \\ \frac{1}{x} & \text{, иначе} \end{cases}$$