Решения на задачи от писмен изпит по Логическо програмиране

24 август 2020

Ако намерите някакъв проблем с решенията, драскайте ми :)

1 Определимост

Нека $\mathcal{L}$ е езикът на предикатното смятане с формално равенство, имащ само един нелогически символ - двуместен функционален символ f. Нека $\mathbb{A}$ е крайна азбука. Да означим с $\mathcal{S}$ структурата за $\mathcal{L}$ с универсиум множеството W на всички думи над $\mathbb{A}$ и f^S конкатенацията на думи над $\mathbb{A}$, т.е. $f^S(u,v) = w \longleftrightarrow u \circ v = w$ за произволни думи u, v и w над $\mathbb{A}$.

1. Да се докаже, че в $\mathcal{S}$ са определими:

   (а) множеството $\mathrm{Pref} = \{\langle u,v\rangle \in W^2 \mid u$ е префикс на $v\}$;
   (б) множеството $\mathrm{Suff} = \{\langle u,v\rangle \in W^2 \mid u$ е суфикс на $v\}$;
   (в) множеството W_1 на еднобуквените думи от W;
   (г) множеството T_1 на думите от W с дължина, ненадминаваща 1;
   (д) за всяко $n \in \mathbb{N}, n > 1$, множеството W_n на n-буквените думи от W;
   (е) за всяко $n \in \mathbb{N}, n > 1$, множеството T_n на думите с дължина, ненадминаваща n;
   (ж) множеството $O = \{\langle u,v,w\rangle \in W^3|$ най-дългият общ префикс на u и v е суфикс на $w\}$.
   (з) множеството $P = \{\langle u,v,w\rangle \in W^3 \mid$ най-дългият общ суфикс на u и на v е префикс на $w\}$.

2. Да се изрази броят на автоморфизмите в $\mathcal{S}$ чрез броя на буквите от $\mathbb{A}$.

1.1 Примерно решение

$$Pref(u,v) \leftrightharpoons \exists w(f(u,w) \doteq v).$$
$$Suff(u,v) \leftrightharpoons \exists w(f(w,u) \doteq v).$$
$$\varphi_\epsilon(u) \leftrightharpoons \forall v(f(u,v) \doteq v).$$
$$W_1(u) \leftrightharpoons \forall v(Pref(v,u) \implies \varphi_\epsilon(v) \vee v \doteq u) \,\&\, \neg\varphi_\epsilon(u).$$
$$T_1(u) \leftrightharpoons \varphi_\epsilon(u) \vee W_1(u).$$

Оттам нататък с индукция по n доказваме, че W_n и T_n са определими като в базата за W е: W_1, индукционната хипотеза е за $W_1, ..., W_n$ и
$W_{n+1}(u) \leftrightharpoons \exists v \exists w(W_1(v) \,\&\, W_n(w) \,\&\, f(v,w) \doteq u)$.
Базата за T е: $_1$, индукционната хипотеза е за T_n и $T_{n+1}(u) \leftrightharpoons T_n(u) \vee W_{n+1}(u)$.
Може още: $T_{n+1}(u) \leftrightharpoons \forall v(Pref(v,u) \implies \varphi_\epsilon(v) \vee W_1(v) \vee ... \vee W_n(v) \vee v \doteq u)$.

Сега за O и P ще имаме помощни ф-ли $\varphi_*(u,v,z)$ и $\phi_*(u,v,z)$, чиито семантики са:

- $\varphi_*(u,v,z)$ is true $\iff$ z е най-дълъг общ префикс на u и v.
- $\psi_*(u,v,z)$ is true $\iff$ z е най-дълъг общ суфикс на u и v.

$$\begin{gathered}\varphi_*(u,v,z) \leftrightharpoons Pref(z,u) \,\&\, Pref(z,v) \,\&\, \forall (Pref(y,u) \,\&\, Pref(y,v) \implies Pref(y,z)).\\ \psi_*(u,v,z) \leftrightharpoons Suff(z,u) \,\&\, Suff(z,v) \,\&\, \forall (Suff(y,u) \,\&\, Suff(y,v) \implies Suff(y,z)).\\ O(u,v,w) \leftrightharpoons \exists z(\varphi_*(u,v,z) \,\&\, Suff(z,w)).\\ P(u,v,w) \leftrightharpoons \exists z(\psi_*(u,v,z) \,\&\, Pref(z,w)).\end{gathered}$$

Сега нека $|\mathbb{A}| = n$ за някое естествено число n. Тогава $|\mathcal{A}\mathrm{ut}(\mathcal{S})| = n!$, тъй като това е броя на всички пермутации над тази азбука (не са повече, т.к изискваме да се имаме биективност и фунционалност на релацията). Сега нека вземем една пермутация над $\mathbb{A}$ примерно h и да покажем, можем да я надградим тази биекция, така че да действа върху всички думи над азбуката $\mathbb{A}$ (бележим го това множество с $\mathbb{A}^*$) и да е автоморфизъм от $\mathbb{A}^*$ в $\mathbb{A}^*$.

Нека $w \in \mathbb{A}^*$. Тогава w е крайна редичка от букви от $\mathbb{A}$:

$$(\exists n \in \mathbb{N})[w = a_1 \circ a_2 \circ ... \circ a_n \,\&\, a_1 \in \mathbb{A} \,\&\, a_2 \in \mathbb{A} \,\&\, ... \,\&\, a_n \in \mathbb{A}].$$

Нека дефинираме $H : \mathbb{A}^* \to \mathbb{A}^*$ така:

$$H(w) = H(a_1 \circ a_2 \circ ... \circ a_n) = h(a_1) \circ h(a_2) \circ ... \circ h(a_n).$$

Искаме H да е биекция и хомоморфизъм. Това че H е биекция се вижда от $H^{HOK(\text{дължини на всички цикли в } h)}(w) = Id_{\mathbb{A}^*}$. Сега тук нелогическите символи са само f и за него ще се погрижим да проверим, че $H(f^S(u,v)) = f^S(H(u),H(v))$ за $u,v \in \mathbb{A}^*$. Т.к. $u,v \in \mathbb{A}^*$, то значи

$$(\exists n \in \mathbb{N})[u = a_1 \circ a_2 \circ ... \circ a_n \,\&\, a_1 \in \mathbb{A} \,\&\, a_2 \in \mathbb{A} \,\&\, ... \,\&\, a_n \in \mathbb{A}]$$

и

$$(\exists m \in \mathbb{N})[v = b_1 \circ b_2 \circ ... \circ b_m \,\&\, b_1 \in \mathbb{A} \,\&\, b_2 \in \mathbb{A} \,\&\, ... \,\&\, b_m \in \mathbb{A}].$$

Ще използваме дефинициите на H, f^S и асоциативност на $\circ$:

$$\begin{gathered}H(f^S(u,v)) = H(f^S(a_1 \circ a_2 \circ ... \circ a_n, b_1 \circ b_2 \circ ... \circ b_m)) =\\ H(a_1 \circ a_2 \circ ... \circ a_n \circ b_1 \circ b_2 \circ ... \circ b_m) =\\ h(a_1) \circ h(a_2) \circ ... \circ h(a_n) \circ h(b_1) \circ h(b_2) \circ ... \circ h(b_m) =\\ (h(a_1) \circ h(a_2) \circ ... \circ h(a_n)) \circ (h(b_1) \circ h(b_2) \circ ... \circ h(b_m)) =\\ f^S(h(a_1) \circ h(a_2) \circ ... \circ h(a_n), (b_1) \circ h(b_2) \circ ... \circ h(b_m)) =\\ f^S(H(u), H(v)).\end{gathered}$$

2 Изпълнимост

Нека a и b са различни индивидни константи, f е триместен функционален символ, а x, y и z са различни индивидни променливи. Да означим с Γ_1 Множеството от следните три формули:

2.1 Вариант 1

$$f(f(x,y,a),z,a) \doteq f(x,f(y,z,a),a),$$
$$f(f(x,y,b),z,b) \doteq f(a,f(y,z,b),b),$$
$$f(f(x,y,a),z,b) \doteq f(f(x,z,b),f(y,z,b),a).$$

2.2 Вариант 2

$$f(a,f(a,x,y),z) \doteq f(a,x,f(a,y,z)),$$
$$f(b,f(b,x,y),z) \doteq f(b,x,f(b,y,z)),$$
$$f(a,x,f(b,y,z)) \doteq f(b,f(a,x,y),f(a,x,z)).$$

Нека:

1. $\Gamma_2 = \Gamma_1 \cup \{(a \doteq b)\}$;
2. $\Gamma_3 = \Gamma_1 \cup \{\neg(a \doteq b)\}$;
3. $\Gamma_4 = \Gamma_2 \cup \{\forall x \forall y \exists z \neg(f(x,y,z) \doteq y)\}$.

Да се докаже кои от множествата $\Gamma_1, \Gamma_2, \Gamma_3$ и Γ_4 са изпълними.

2.3 Примерни решения вариант 1

За Γ_1, Γ_2**:**
$S_1 = (\{0,1\}; f^{S_1}; a^{S_1}, b^{S_1})$
$f^{S_1}(x,y,z) \leftrightharpoons y$
$a^{S_1} \leftrightharpoons 0, b^{S_1} \leftrightharpoons 0$

За Γ_3:
$S_2 = (\{0,1\}; f^{S_2}; a^{S_2}, b^{S_2})$
$f^{S_2}(x,y,z) \leftrightharpoons y$
$a^{S_2} \leftrightharpoons 0, b^{S_2} \leftrightharpoons 1$

За $\Gamma_1, \Gamma_2, \Gamma_4$:
$S_3 = (\{0,1\}; f^{S_3}; a^{S_3}, b^{S_3})$
$f^{S_3}(x,y,z) \leftrightharpoons z$
$a^{S_3} \leftrightharpoons 0, b^{S_3} \leftrightharpoons 0$

За $\Gamma_1, \Gamma_2, \Gamma_4$:
$S_4 = (\mathbb{N}; f^{S_4}; a^{S_4}, b^{S_4})$
$f^{S_4}(x,y,z) \leftrightharpoons max\{x,y,z\}$
$a^{S_4} \leftrightharpoons 0, b^{S_4} \leftrightharpoons 0$

За $\Gamma_1, \Gamma_2, \Gamma_4$:
$S_5 = (\mathbb{Z}^- \cup \{0\}; f^{S_5}; a^{S_5}, b^{S_5})$
$f^{S_5}(x,y,z) \leftrightharpoons min\{x,y,z\}$
$a^{S_5} \leftrightharpoons 0, b^{S_5} \leftrightharpoons 0$

Помислете какво трябва да промените в структурите, за да получите модели за вариант 2.

3 Резолюция

3.1 Вариант 1

Нека $\varphi_1, \varphi_2, \varphi_3$ и φ_4 са следните четири формули:

$$\varphi_1 \leftrightharpoons \forall x \exists y((q(x,y) \Rightarrow p(x,y)) \,\&\, \forall z(p(z,y) \Rightarrow r(x,z))).$$
$$\varphi_2 \leftrightharpoons \forall x(\exists y p(y,x) \Rightarrow \exists y(p(y,x) \,\&\, \neg\exists z(p(z,y) \,\&\, p(z,x)))).$$
$$\varphi_3 \leftrightharpoons \forall z \left(\exists x \exists y(\neg q(x,y) \,\&\, \neg p(x,y)) \Rightarrow \forall z_1 q\left(z_1, z\right)\right).$$
$$\varphi_4 \leftrightharpoons \neg\exists x \exists y \exists z((p(x,y) \,\&\, r(y,z)) \,\&\, \neg p(x,z)).$$

С метода на резолюцията докажете, че:
$\varphi_1, \varphi_2, \varphi_3, \varphi_4 \models \exists y \forall x \exists z((p(x,x) \vee r(y,z)) \Rightarrow (\neg p(x,x) \,\&\, r(y,z)))$.

3.2 Вариант 2

Нека $\varphi_1, \varphi_2, \varphi_3$ и φ_4 са следните четири формули:

$$\varphi_1 \leftrightharpoons \forall x \exists y(q(y,x) \,\&\, \forall z(q(y,z) \Rightarrow (r(z,x) \vee p(y,z)))).$$
$$\varphi_2 \leftrightharpoons \forall x(\exists y q(x,y) \Rightarrow \exists y(q(x,y) \,\&\, \neg\exists z(q(y,z) \,\&\, q(x,z)))).$$
$$\varphi_3 \leftrightharpoons \forall z_1 \left(\exists z \exists x \exists y(p(x,y) \,\&\, q(x,z)) \Rightarrow \forall z_2 \neg p\left(z_1, z_2\right)\right).$$
$$\varphi_4 \leftrightharpoons \neg\exists x \exists y \exists z((q(y,x) \,\&\, r(z,y)) \,\&\, \neg q(z,x)).$$

С метода на резолюцията докажете, че:
$\varphi_1, \varphi_2, \varphi_3, \varphi_4 \models \exists z \forall x \exists y((q(x,x) \vee r(y,z)) \Rightarrow (\neg q(x,x) \,\&\, r(y,z)))$.

3.3 Обработка на някои от формулите (задачата е същата като тази на 6 юли 2020)

Нека:
$\chi' \rightleftharpoons \exists y \forall x \exists z((p(x,x) \vee r(y,z)) \Rightarrow (\neg p(x,x) \,\&\, r(y,z)))$;
$\chi'' \rightleftharpoons \exists z \forall x \exists y((q(x,x) \vee r(y,z)) \Rightarrow (\neg q(x,x) \,\&\, r(y,z)))$.
Сега:

$$\begin{aligned}
\chi' &\,\|\!\models\!\|\, \exists y \forall x \exists z((\neg p(x,x) \,\&\, \neg r(y,z)) \vee (\neg p(x,x) \,\&\, r(y,z))) \\
&\,\|\!\models\!\|\, \exists y \forall x \exists z(\neg p(x,x) \,\&\, (\neg r(y,z) \vee r(y,z))) \\
&\,\|\!\models\!\|\, \exists y \forall x \exists z(\neg p(x,x)) \\
&\,\|\!\models\!\|\, \forall x \neg p(x,x).
\end{aligned}$$

Аналогично $\chi'' \mathrel{|}\!\!\models\!\!\mathrel{|} \forall x \neg q(x,x)$. Остава само φ_4 да преобразуваме и получаваме задачата от 6ти юли:

$$\varphi_4 \mathrel{|}\!\!\models\!\!\mathrel{|} \forall x \forall y \forall z((p(x,y) \,\&\, r(y,z)) \Rightarrow p(x,z))$$

Същото и за другия вариант.

3.4 Примерно решение за вариант 1 (вариант 2 е аналогичен)

Получаваме следните формули като приведем в ПНФ, СНФ и КНФ:

$$\varphi_1^{final} \rightleftharpoons \forall x \forall z((p(x,f(x)) \vee \neg q(x,f(x))) \,\&\, (\neg p(z,f(x)) \vee r(x,z))).$$
$$\varphi_2^{final} \rightleftharpoons \forall x \forall t \forall z((p(g(x),x) \vee \neg p(t,x)) \,\&\, (\neg p(z,g(x)) \vee \neg p(z,x) \vee \neg p(t,x)).$$
$$\varphi_3^{final} \rightleftharpoons \forall z \forall x \forall y \forall z_1(q(x,y) \vee p(a,y), \vee q(z_1,z)).$$
$$\varphi_4^{final} \rightleftharpoons \forall x \forall y \forall z(\neg r(y,z) \vee \neg p(x,y) \vee p(x,z)).$$
$$\psi^{final} \rightleftharpoons p(a,a).\ \textbf{(Ползваме } \chi' \textbf{ за база.)}$$

Дизюнктите са (нека ги номерираме променливите по принадлежност към дизюнкт):

$$D_1 = \{\neg q(x_1, f(x_1)), p(x_1, f(x_1))\};$$
$$D_2 = \{\neg p(z_2, f(x_2)), r(x_2, z_2)\};$$
$$D_3 = \{p(g(x_3), x_3), \neg p(t_3, x_3)\};$$
$$D_4 = \{\neg p(z_4, g(x_4)), \neg p(z_4, x_4), \neg p(t_4, x_4)\};$$
$$D_5 = \{q(x_5, y_5), p(x_5, y_5), q(z_1, z_5)\};$$
$$D_6 = \{\neg r(y_6, z_6), \neg p(x_6, y_6), p(x_6, z_6)\};$$
$$D_7 = \{p(a,a)\}.$$

Примерен резолютивен извод на $\blacksquare$ е:

$$D_8 = Collapse(D_5\{z_1/x_5, z_5, y_5\}) = \{q(x_5, y_5), p(x_5, y_5)\};$$

$$D_9 = Res(D_1, D_8\{x_5/x_1, y_5/f(x_1)\}) = \{p(x_1, f(x_1))\};$$

$$D_{10} = Res(D_2\{x_2, y_6, z_2/z_6\}, D_6) = \{\neg p(z_6, f(y_6)), \neg p(x_6, y_6), p(x_6, z_6)\};$$

$$D_{11} = Res(D_{10}\{z_6/g(f(y_6))\}, D3\{x_3/f(y_6)\}) = \{\neg p(t_3, f(y_6)), \neg p(x_6, y_6), p(x_6, g(f(y_6)))\};$$

$$D_{12} = Res(D_{11}, D_4\{x_4/f(y_6), z_4/x_6\}) = \{\neg p(t_3, f(y_6)), \neg p(x_6, y_6), \neg p(t_4, f(y_6)), \neg p(x_6, f(y_6))\};$$

$$D_{13} = Collapse(D_{12}\{t_3/a, y_6/a, t_4/a, x_6/a\}) = \{\neg p(a, f(a)), \neg p(a, a)\}.;$$

$$D_{14} = Res(D_{13}, D_9\{x_1/a\}) = \{\neg p(a, a)\};$$

$$\blacksquare = Res(D_{14}, D_7).$$

4 Пролог: задача за дървета

Дърво се нарича краен неориентиран свързан и ацикличен граф. За един списък от списъци $[V, E]$ ще казваме, че представя неориентирания граф G, ако V е списък от всички върхове на G и $\{v, w\}$ е ребро в G тогава и само тогава, когато $[v, w]$ или $[w, v]$ е елемент на E.
Да се дефинира на пролог предикат $art\_tree(V, E)/arc\_tree(V, E)$, който по дадено представяне $[V, E]$ на краен неориентиран граф разпознава дали има такава двойка върхове v и w, че $[V, E + [v, w]]/[V, E - [v, w]]$ да е представяне на дърво, където $E + [v, w]/E - [v, w]$ е списъкът, получен от E с премахването на всички срещания на елемента $[v, w]$/добавянето на нов елемент $[v, w]$.

4.1 Общи предикати

```prolog
% Helper predicates: member, append, length.

takeNeighbourVertex(E, U, W):-
    member([U, W], E); member([W, U], E).

removeAll([], _, []).
removeAll([H|T], H, R):- removeAll(T, H, R).
removeAll([H|T], X, [H|R]):- H \= X, removeAll(T, H, R).

% acyclicPath(Egdes, Start, [End], Path).
acyclicPath(_, U, [U|P], [U|P]).
acyclicPath(E, U, [W|P], Result):-
    U \= W,
    takeNeighbourVertex(E, Prev, W),
    not(member(Prev, [W|P])),
    acyclicPath(E, U, [Prev, W|P], Result).

isConnected([V, E]):- not(( member(U, V), member(W, V),
            not(acyclicPath(E, U, [W], _)) )).

isAcyclic([V, E]):-
    not(( member(U, V), acyclicPath(E, U, [U], P), P \= [_] )).
    clearRepeatedEdges([], []).

% clearRepeatedEdges(Edges, EdgesWithNoDuplicates).
clearRepeatedEdges([[U, W]|Rest], [[U, W]|Result]):-
    clearRepeatedEdges(Rest, Result), not(member([W, U],
    ↪ Result)).
```

```
clearRepeatedEdges([[U, W]|Rest], Result):-
    clearRepeatedEdges(Rest, Result), member([W, U], Result).
```

4.2 Примерно решение 1

```
% G is connected and acyclic (contains no cycles).
addEdgeIfNeeded([U, W], E, [[U, W]|E]):- not(member([U, W], E)).
addEdgeIfNeeded([U, W], E, E):- member([U, W], E).

removeEdgeIfNeeded([U, W], E, NewE):- removeAll(E, [U, W], NewE).

isTree([V, E]):- isConnected([V, E]), isAcyclic([V, E]).

art_tree([V, E], [U, W]):-
    member(U, V), member(W, V),
    addEdgeIfNeeded([U, W], E, CandidateE),
    isTree([V, CandidateE]).

arc_tree([V, E], [U, W]):-
    member(U, V), member(W, V),
    removeEdgeIfNeeded([U, W], E, CandidateE),
    isTree([V, CandidateE]).
```

4.3 Примерно решение 2

```
% Идея с покриващо дърво.

% stree(V, E, Vis, NotVis, Result).
stree(_, _, [], []).
stree(E, Vis, NotVis, [[X,Y]|R]):-
    member(X, Vis), member(Y, NotVis),
    takeNeighbourVertex(X, Y, E),
    removeAll(Y, NotVis, NotVisNew),
    stree(E, [Y|Vis], NotVisNew, R).

% spanTree(Graph, Tree).
spanTree([[], []], [[], []]).
spanTree([[X|V], E], [[X|V], T]):- stree(E, [X], V, T).

acyclic([V, E]):-
    spanTree([V, E], [V, T]),
    not(( member([X, Y], E),
```

```
        not(takeNeighbourVertex(X, Y, T)) )).

arc_tree1([V, E]):-
    member([X, Y], E),
    removeAll(E, [X, Y], E1),
    acyclic([V, E1]).
```

4.4 Примерно решение 3

```
% G is connected and has n — 1 edges.
art_tree2([V, E], [U, W]):-
    clearRepeatedEdges(E, NewE),
    member(U, V), member(W, V),
    addEdgeIfNeeded([U, W], E, CandidateE),
    length(V, N), N1 is N - 1,
    length(CandidateE, N1),
    isConnected([V, CandidateE]).
```

4.5 Примерно решение 4

```
% G has no simple cycles and has n — 1 edges.
art_tree3([V, E], [U, W]):-
    clearRepeatedEdges(E, NewE),
    member(U, V), member(W, V),
    addEdgeIfNeeded([U, W], E, CandidateE),
    length(V, N), N1 is N - 1,
    length(CandidateE, N1),
    isAcyclic([V, CandidateE]).
```

5 Пролог: задача за списъци

Казваме, че списъкът X е екстерзала/екстерзана за списъка от списъци Y, ако X има поне един общ елемент с всички елементи на Y и поне два общи елемента с нечетен/четен брой елементи на Y Да се дефинира на пролог двуместен предикат екстерзала (X, Y), който по даден списък от списъци Y при презадоволяване генерира всички екстерзали/екстервали X за Y с възможно най-малка дължина и спира.

5.1 Превод на условията за екстерзала/ексервала

Условие 1: $(\forall A \in Y)(\exists B \in X)[B \in A]$.
Условие 2:
$|\{A|A \in Y, [append(\_, [B|L], X), member(C, L)[B \in A \,\&\, C \in A]]\}_M| \equiv 1/0 \mod 2$, където долен индекс M значи мултимножество.

5.2 Примерно решение

```
% Helper predicates: member, append, length, permutation.

generateIntersection([], _, []).
generateIntersection([H|T], B, [H|R]):- member(H, B),
↪  generateIntersection(T, B, R).
generateIntersection([H|T], B, R):- not(member(H, B)),
↪  generateIntersection(T, B, R).

subsequence([], []).
subsequence([H|T], [H|R]):- subsequence(T, R).
subsequence([_|T], R):- subsequence(T, R).

conditionExterzala(X, Y):-
    haveCommonAtLeastNCommonElements(X, Y, 1, N),
    length(Y, N),
    haveCommonAtLeastNCommonElements(X, Y, 2, M),
    isOdd(M).

conditionExtervala(X, Y):-
    haveCommonAtLeastNCommonElements(X, Y, 1, N),
    length(Y, N),
    haveCommonAtLeastNCommonElements(X, Y, 2, M),
    isEven(M).
```

```
isOdd(M):- M mod 2 =:= 1.

isEven(M):- M mod 2 =:= 0.

isLowerBoundOkey(Int, LowerBound, 1):- length(Int, LenInt),
 ↪  LenInt >= LowerBound.
isLowerBoundOkey(Int, LowerBound, 0):- length(Int, LenInt),
 ↪  LenInt < LowerBound.

haveCommonAtLeastNCommonElements(_, [], _, 0).
haveCommonAtLeastNCommonElements(X, [H|T], LowerBound, N):-
    haveCommonAtLeastNCommonElements(X, T, LowerBound, M),
    generateIntersection(X, H, Int),
    isLowerBoundOkey(Int, LowerBound, Bit),
    N is M + Bit.


generateCandidateX([], []).
generateCandidateX([H|T], ResultX):-
    generateCandidateX(T, CurrentX),
    subsequence(H, H1),
    append(H1, CurrentX, ResultX).

exterzala(X, Y):-
    generateCandidateX(Y, X1), permutation(X1, X),
     ↪  conditionExterzala(X, Y), length(X, N),
    not(( generateCandidateX(Y, X1), conditionExterzala(X1, Y),
     ↪  length(X1, M), M < N )).

extervala(X, Y):-
    generateCandidateX(Y, X1), permutation(X1, X),
     ↪  conditionExtervala(X, Y), length(X, N),
    not(( generateCandidateX(Y, X1), conditionExtervala(X1, Y),
     ↪  length(X1, M), M < N )).
```