

Решения на задачи от писмен изпит по
Логическо програмиране

31 август 2018

Съдържание

1	Първа задача на пролог	1
1.1	Общи предикати	1
1.2	Примерно решение на I.1	1
1.3	Примерно решение на I.2	1
2	Втора задача на пролог	3
2.1	Общи предикати	3
2.2	Примерно решение на I.1	3
2.3	Примерно решение на I.2	4
3	Задача за определимост	5
3.1	Общи формули и за двата подхода на разглеждане на задачата	5
3.2	Примерно решение за случая, когато го разглеждаме като не- кореново дърво (нямат посоки ребрата)	5
3.3	Примерно решение за случая, когато го разглеждаме като ко- реново дърво (имат посоки ребрата)	6
4	Задача за изпълнимост	7
4.1	Примерно решение 1	7
4.2	Примерно решение 2	7
4.3	Примерно решение 3	8
5	Задача за резолюция	9
5.1	Примерно решение	9

1 Първа задача на пролог

Да се дефинира на пролог предикат $q(X)$, който при преудовлетворяване генерира в X всички списъци, които представляват крайни:

I.1 аритметични прогресии от съставни естествени числа.

I.2 геометрични прогресии, нито един член на които не е квадрат на естествени числа.

1.1 Общи предикати

```
nat(0).
nat(N):- nat(M), N is M + 1.

between(A, B, A):- A =:= B.
between(A, B, C):- A < B, A1 is A + 1, between(A1, B, C).

genKS(1, S, [S]).
genKS(K, S, [X|R]):- K > 1, K1 is K - 1,
    between(0, S, XI), S1 is S - XI, genKS(K1, S1, R).
```

1.2 Примерно решение на I.1

```
genArithProg(_, 0, _, []).
genArithProg(Current, N, Diff, [Current|Result]):-
    N > 0, N1 is N - 1, Next is Current + Diff,
    genArithProg(Next, N1, Diff, Result).

isPrime(X):- X > 1, N is X // 2,
    not((between(2, N, Y), X mod Y =:= 0)).

main([]).
main(L):- nat(N), genKS(3, N, [Start, NumOfElem, Diff]),
    Diff > 0, NumOfElem > 0,
    genArithProg(Start, NumOfElem, Diff, L),
    not((member(X, L), isPrime(X))).
```

1.3 Примерно решение на I.2

```
genGeomProg(_, 0, _, []).
genGeomProg(Current, N, Diff, [Current|Result]):-
    N > 0, N1 is N - 1, Next is Current * Diff,
    genGeomProg(Next, N1, Diff, Result).

isSquare(X):- N is X // 2, between(0, N, Y), Y * Y =:= X.
```

```

main ([ ] ).
main(L):- nat(N), genKS(3, N, [Start, NumOfElem, Diff]),
          Start > 0, Diff > 0, NumOfElem > 0,
          genGeomProg(Start, NumOfElem, Diff, L),
          not((member(X, L), not(isSquare(X)))).

```

2 Втора задача на пролог

I.1 Да се дефинират на пролог едноместни предикати p_1, p_2, p_3 и p_4 . такива че даден списък X :

- (1) p_1 разпознава дали празният списък е елемент на X ,
- (2) p_2 разпознава дали X съдържа елементи Y и Z , такива че не всички елементи на Y са елементи на Z ,
- (3) p_3 разпознава дали X съдържа елемент Y , който съдържа всички елементи на всички елементи на X ,
- (4) p_4 разпознава дали за всеки елемент Y на X съществува такъв елемент Z на X , че не всички елементи на Z са елементи на Y .

I.2 Да се дефинират на пролог едноместни предикати q_1, q_2, q_3 и q_4 . такива че даден списък X :

- (1) q_1 разпознава дали празният списък е елемент на X ,
- (2) q_2 разпознава дали X съдържа елементи Y и Z , които нямат общи елементи,
- (3) q_3 разпознава дали X съдържа елемент Y , чиито елементи са еленти на всички елементи на X ,
- (4) q_4 разпознава дали за всеки елемент Y на X съществува такъв елемент Z на X , че Y и Z нямат общи елементи.

* **Бонус задача:** да се дефинира предикат, който разпознава дали списък L е списък от списъци.

2.1 Общи предикати

```
append([], L2, L2).
append([H|T], L2, [H|R]):- append(T, L2, R).

isSubsetOf(A, B):- not((member(X, A), not(member(X, B)))).

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

is_list([]).
is_list([_|_]).

isListOfLists(L):- not((member(X, L), not(is_list(X)))).
```

2.2 Примерно решение на I.1

```
p1(L):- member([], L).
```

```

p2(L):- member(Y, L), member(Z, L),
        not(isSubsetOf(Y, Z)).

p3(L):- member(Y, L), not((member(Z, L),
        not(isSubsetOf(Z, Y)))).

p4(L):- not((member(Y, L), not((member(Z, L),
        not(isSubsetOf(Z, Y)))))).
% p4(X) :- not(p3(X)).

```

2.3 Примерно решение на I.2

```

haveInCommon(X, Y):- member(A, X), member(A, Y).

q1(L):- member([], L).

q2(L):- member(Y, L), member(Z, L),
        not((haveInCommon(Y, Z))).

q3(L):- member(Y, L), not((member(Z, L),
        not(isSubsetOf(Y, Z)))).

q4(L):- not((member(Y, L), not((member(Z, L),
        not(haveInCommon(Y, Z)))))).

```

3 Задача за определимост

Структурата S е с носител множеството от всички дървета, чиито върхове са естествени числа, и е с език с формално равенство и двуместен предикатен символ sub , който се интерпретира така:

ако $T_1 = \langle V_1, E_1 \rangle$ и $T_2 = \langle V_2, E_2 \rangle$ са дървета от универсиума на S , то:
 $sub(T_1, T_2) \leftrightarrow V_1 \subseteq V_2$ и $E_1 \subseteq E_2$.

Да се определят:

- множеството от тривиалните дървета
 $\{\langle \{n\}, \emptyset \rangle : n \in \mathbb{N}\}$;
- множеството от дърветата с два върха
 $\{\langle \{n, m\}, \{\langle n, m \rangle\} \rangle : n, m \in \mathbb{N}, n \neq m\}$;
- предикатът $leaf(T_1, T_2)$, който е верен точно тогава, когато T_1 е дърво с единствен връх и този връх е листо на T_2 ;
- множеството от онези дървета, в който никой връх не е от степен, по-голяма от 2;

3.1 Общи формули и за двата подхода на разглеждане на задачата

Първо определяме празното дърво:

$$\varphi_{\emptyset}(t_1) \equiv \forall t_2 sub(t_1, t_2).$$

След това тривиалното:

$$\varphi_{trivial_tree}(t_1) \equiv \neg \varphi_{\emptyset}(t_1) \& \forall t_2 (sub(t_2, t_1) \implies (\varphi_{\emptyset}(t_2) \vee (t_1 \doteq t_2))).$$

След това и дървото с два върха:

$$\begin{aligned} \varphi_{edge}(t_1) \equiv & \neg \varphi_{\emptyset}(t_1) \& \neg \varphi_{trivial_tree}(t_1) \& \\ & \forall t_2 (sub(t_2, t_1) \implies (\varphi_{\emptyset}(t_2) \vee \varphi_{trivial_tree}(t_2) \vee (t_1 \doteq t_2))). \end{aligned}$$

Дърво с двоичен връх е такова дърво, за което съществуват две листа (не непременно различни) и чието премахване предизвиква появата на ново листо:

$$\begin{aligned} \varphi_{binary_node}(t_1) \equiv & \exists t_2 \exists t_3 (\varphi_{leaf}(t_2, t_1) \& \varphi_{leaf}(t_3, t_1) \& \\ & \forall t_4 ((sub(t_4, t_1) \& \neg sub(t_2, t_4) \& \neg sub(t_3, t_4)) \implies \exists t_5 (\neg \varphi_{leaf}(t_5, t_1) \& \varphi_{leaf}(t_5, t_4))))). \end{aligned}$$

Двоичното дърво е такова дърво, че всяко негово поддърво е от предния тип:

$$\varphi_{binary}(t_1) \equiv \forall t_2 ((sub(t_2, t_1) \& \neg \varphi_{\emptyset}(t_2) \& \neg \varphi_{trivial}(t_2)) \implies \varphi_{binary_node}(t_2)).$$

3.2 Примерно решение за случая, когато го разглеждаме като некореново дърво (нямат посоки ребрата)

Листото е такъв връх, за който съществува поддърво на изходното (цялото без листото), такова че каквото и негово разширение да вземем, което не съвпада с изходното, вече няма да е поддърво на изходното.

$$\begin{aligned} \varphi_{leaf}(t_1, t_2) \equiv & sub(t_1, t_2) \& \varphi_{trivial_tree}(t_1) \& \\ & \exists t_3 \forall t_4 ((sub(t_3, t_2) \& \neg sub(t_1, t_3) \& \neg (t_3 \doteq t_4) \& sub(t_3, t_4) \& \neg (t_4 \doteq t_2))) \end{aligned}$$

$$\implies \neg sub(t_4, t_2)).$$

Друг подход, чрез определяне на това какво е вътрешен връх. Това е такъв връх, чието отстраняване предизвиква появата на поне две нови различни дървета, които са максимални поддървета на изходното:

$$\begin{aligned} \varphi_{inner}(t_1, t_2) \equiv & sub(t_1, t_2) \& \varphi_{trivial_tree}(t_1) \& \neg \varphi_{trivial_tree}(t_2) \& \\ & \exists t_3 \exists t_4 (\neg(t_3 \doteq t_4) \& \varphi_{mswi}(t_3, t_2, t_1) \& \varphi_{mswi}(t_4, t_2, t_1)). \end{aligned}$$

MSWI = Maximum subtree without inner node.

$$\begin{aligned} \varphi_{mswi}(t_1, t_2, t_3) \equiv & sub(t_1, t_2) \& \neg sub(t_3, t_1) \& \\ & \forall t_4 ((sub(t_4, t_2) \& \neg sub(t_3, t_4)) \implies sub(t_4, t_1)). \end{aligned}$$

3.3 Примерно решение за случая, когато го разглеждаме като кореново дърво (имат посоки ребрата)

$$\begin{aligned} \varphi_{three_nodes}(t_1) \equiv & \neg \varphi_{\emptyset}(t_1) \& \neg \varphi_{trivial_tree}(t_1) \& \neg \varphi_{edge}(t_1) \& \\ & \forall t_2 (sub(t_2, t_1) \implies (\varphi_{\emptyset}(t_2) \vee \varphi_{trivial_tree}(t_2) \vee \varphi_{edge}(t_2) \vee (t_1 \doteq t_2))). \end{aligned}$$

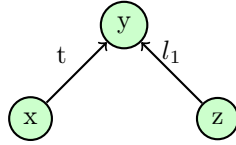
Определяме обръщане на посоката на дадено ребро:

$$\varphi_{change_direction}(t_1, t_2) \equiv \varphi_{edge}(t_1) \& \varphi_{edge}(t_2) \& sub(t_1, t_2) \& \neg(t_1 \doteq t_2).$$

Определяме посоката на дадено ребро с върхове x и y т.е.

$$edge = x \longrightarrow y :$$

$$\begin{aligned} \varphi_{direction}(x, y) \equiv & \exists z \exists t \exists l \exists l_1 (\varphi_{trivial}(x) \& \varphi_{trivial}(y) \& \varphi_{trivial}(z) \& \varphi_{edge}(t) \& \varphi_{edge}(l) \\ & \& \varphi_{change_direction}(l, l_1) \& sub(x, t) \& sub(y, t) \& sub(y, l) \& sub(z, l) \& \\ & \neg \exists k (sub(t, k) \& sub(l_1, k) \& \varphi_{three_nodes}(k))). \end{aligned}$$



Последната формула казва, че не съществува такова дърво в нашия универсиум и така определяма посоката от x към y.

Определяме кога един връх е листо. Също и кога е корен:

$$\varphi_{root}(t_1, t_2) \equiv sub(t_1, t_2) \& \varphi_{trivial}(t_1) \& \neg \exists t_3 \varphi_{direction}(t_3, t_1).$$

$$\varphi_{leaf}(t_1, t_2) \equiv sub(t_1, t_2) \& \varphi_{trivial}(t_1) \& \neg \exists t_3 \varphi_{direction}(t_1, t_3).$$

4 Задача за изпълнимост

Да се докаже, че е изпълнимо множеството, съставено от следните формули (а и b са индивидуални константи):

II.1

$$\begin{aligned} &\forall x(p(a, x) \& p(x, b)) \\ &\forall x(\forall y(p(x, y) \vee p(y, x)) \implies (x = a \vee x = b)) \\ &\forall x \forall y(\forall z(x = z \vee y = z \vee \neg p(x, z) \vee \neg p(z, y)) \implies \neg p(x, y)) \end{aligned}$$

II.2

$$\forall x(r(x, a) \& r(b, x))$$

същите като в II.1, но вместо p е r.

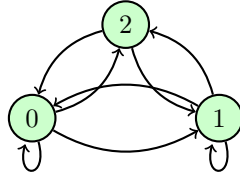
В контрапозиция втората формула казва, че за всеки обект, различен от а и b, съществува друг, с който не е свързан.

$$\forall x((x \neq a \& x \neq b) \implies \exists y(\neg p(x, y) \& \neg p(y, x)))$$

В контрапозиция третата казва, че между всеки два елемента в релацията съществува трети различен от първите два, посредством който са свързани.

$$\forall x \forall y(p(x, y) \implies \exists z(x \neq z \& y \neq z \& p(x, z) \& p(z, y)))$$

4.1 Примерно решение 1



$$\begin{aligned} S &= (\{0, 1, 2\}, p^S, a^S, b^S) \\ p^S &= \{(0, 0), (0, 1), (0, 2), (1, 1), (1, 0), (1, 2), (2, 1), (2, 0)\} \\ a^S &\doteq 0, b^S \doteq 1 \end{aligned}$$

4.2 Примерно решение 2

$$\begin{aligned} S &= (\{\clubsuit, \diamond, 1, 2, 3\}, p^S, a^S, b^S) \\ \langle x, y \rangle \in p^S &\iff x = \clubsuit \vee x = \diamond \vee y = \clubsuit \vee y = \diamond \\ a^S &\doteq \clubsuit, b^S \doteq \diamond \end{aligned}$$

4.3 Примерно решение 3

$S = (\text{множеството от всички реални интервали}, p^S, a^S, b^S)$
 $p^S \Rightarrow \subseteq$ в смисъла на подинтервал

II.1 $a^S \Rightarrow \emptyset, b^S \Rightarrow \mathbb{R}$

II.2 $a^S \Rightarrow \mathbb{R}, b^S \Rightarrow \emptyset$

5 Задача за резолюция

Нека φ_1, φ_2 и φ_3 са следните три формули:

II.1

$$\begin{aligned}\forall x \neg \forall y (q(y, x) \implies \exists z (q(y, z) \& \neg r(z, x))), \\ \forall x (\forall y (q(x, y) \implies \exists z (q(y, z) \& q(x, z))) \implies \neg \exists z q(x, z)), \\ \forall z \forall y (r(z, y) \implies \neg \exists x (q(y, x) \& \neg q(z, x))).\end{aligned}$$

II.2

$$\begin{aligned}\forall x \neg \forall y (\forall z (p(z, y) \implies r(x, z)) \implies \neg p(x, y)), \\ \forall x (\exists y p(y, x) \implies \exists y (p(y, x) \& \neg \exists z (p(z, y) \& p(z, x)))), \\ \forall x \neg \exists y (p(x, y) \& \neg \forall z (r(y, z) \implies p(x, z))).\end{aligned}$$

С метода на резолюцията да се докаже, че

$$\varphi_1, \varphi_2, \varphi_3 \models \forall x \exists y (p(x, x) \implies \exists z (r(z, y) \& \neg r(z, y))).$$

* Тъй като $(r(z, y) \& \neg r(z, y))$ е винаги лъжа, то ψ дефинираме като:

$$\neg \forall x \exists y (\neg p(x, x)) \equiv \exists x p(x, x).$$

Респективно q за вариант II.1.

5.1 Примерно решение

II.1

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

$$\begin{aligned}\varphi_1^S &\equiv \forall x \forall z (q(f(x), x) \& (\neg q(f(x), z) \vee r(z, x))), \\ \varphi_2^S &\equiv \forall x \forall z \forall t ((q(x, g(x)) \vee \neg q(x, t)) \& (\neg q(g(x), z) \vee \neg q(x, z) \vee \neg q(x, t))), \\ \varphi_3^S &\equiv \forall z \forall y \forall x (\neg r(z, y) \vee \neg q(y, x) \vee q(z, x)), \\ \psi^S &\equiv q(a, a).\end{aligned}$$

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

$$\begin{aligned}D_1 &= \{q(f(x_1), x_1)\}; \\ D_2 &= \{\neg q(f(x_2), z_2), r(z_2, x_2)\}; \\ D_3 &= \{q(x_3, g(x_3)), \neg q(x_3, t_3)\}; \\ D_4 &= \{\neg q(g(x_4), z_4), \neg q(x_4, z_4), \neg q(x_4, t_4)\}; \\ D_5 &= \{\neg r(z_5, y_5), \neg q(y_5, x_5), q(z_5, x_5)\}; \\ D_6 &= \{q(a, a)\}.\end{aligned}$$

II.2

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

$$\begin{aligned}\varphi_1^S &\equiv \forall x \forall z (p(x, f(x)) \& (\neg p(z, f(x)) \vee r(x, z))), \\ \varphi_2^S &\equiv \forall x \forall t \forall z ((p(g(x), x) \vee \neg p(t, x)) \& (\neg p(z, g(x)) \vee \neg p(z, x) \vee \neg p(t, x))), \\ \varphi_3^S &\equiv \forall x \forall y \forall z (\neg r(y, z) \vee \neg p(x, y) \vee p(x, z)), \\ \psi^S &\equiv p(a, a).\end{aligned}$$

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

$$\begin{aligned}
D_1 &= \{p(x_1, f(x_1))\}; \\
D_2 &= \{\neg p(z_2, f(x_2)), r(x_2, z_2)\}; \\
D_3 &= \{p(g(x_3), x_3), \neg p(t_3, x_3)\}; \\
D_4 &= \{\neg p(z_4, g(x_4)), \neg p(z_4, x_4), \neg p(t_4, x_4)\}; \\
D_5 &= \{\neg r(y_5, z_5), \neg p(x_5, y_5), p(x_5, z_5)\}; \\
D_6 &= \{p(a, a)\}.
\end{aligned}$$

И за двата варианта един примерен резолютивен извод на ■ е:

$$\begin{aligned}
D_7 &= Res(D_2\{x_2/y_5, z_2/z_5\}, D_5) = \\
&\quad \{\neg q(f(y_5), z_5), \neg q(y_5, x_5), q(z_5, x_5)\}; \\
D_8 &= Res(D_3\{x_3/f(y_5)\}, D_7\{z_5/g(f(y_5))\}) = \\
&\quad \{\neg q(f(y_5), t_3), \neg q(y_5, x_5), q(g(f(y_5)), x_5)\}; \\
D_9 &= Res(D_4\{x_4/f(y_5), z_4/x_5\}, D_8) = \\
&\quad \{\neg q(f(y_5), t_4), \neg q(f(y_5), x_5), \neg q(f(y_5), t_3), \neg q(y_5, x_5)\}; \\
D_{10} &= Collapse(D_9\{t_3/x_5, t_4/x_5\}) = \\
&\quad \{\neg q(f(y_5), x_5), \neg q(y_5, x_5)\}; \\
D_{11} &= Res(D_1, D_{10}\{y_5/x_1, x_5/x_1\}) = \\
&\quad \{\neg q(x_1, x_1)\}; \\
Res(D_{11}\{x_1/a\}, D_6) &= \blacksquare.
\end{aligned}$$

За вариант II.2 заменете q с p.