

ИЗУЧЕНИЕ ДИНАМИКИ ДИСКРЕТНЫХ СИСТЕМ МЕТОДАМИ АЛГЕБРЫ ПОДСТАНОВОЧНЫХ ПРЕДИКАТОВ

Для исследования логических (двоичных) систем обычно используется алгебра логики. С ее помощью ставится и решается большой круг задач описания, преобразования и построения логических систем. Переход от классической алгебры логики к алгебре подстановочных предикатов [1] не только расширяет круг решаемых задач, но и приводит к эффективным реализациям получения решений.

Вопросы динамики логических систем очень важны для практических приложений и им, естественно, уделяется должное внимание. Начиная с 50-х годов прошлого века появляются разные модели динамики логических систем. Наш взгляд наиболее перспективным является булево дифференциальное исчисление [2, 3]. В книге Д.Бохманна и Х.Постхофа [3] изложен аппарат булевых дифференциальных уравнений. Эти уравнения описывают связи между состояниями некоторой булевой системы. Переменные системы *принимает* значения из $B = \{0, 1\}$, а отображения упорядоченного множества переменных $X \rightarrow B$ определяют состояния системы. Не все состояния являются допустимыми с точки зрения решаемой задачи. Если состояния рассматривать как вершины некоторого графа, а связи между состояниями как его ребра, то динамика поведения системы может быть описана с помощью этого графа.

В работе под состояниями системы понимаются наборы таблицы истинности некоторого подстановочного предиката. Ребром соединяем две вершины-состояния x и x^* , которые переходят друг в друга при допустимых, с точки зрения условий задачи, изменениях переменных. Алгебра подстановочных предикатов позволяет описывать ребра (x, x^*) в случае, когда определена стратегия изменения переменной (или группы переменных).

Динамика систем, множество состояний которой - наборы некоторого подстановочного предиката (или системы предикатов) является естественным обобщением динамики булевых систем [3] и, насколько нам известно, в математической литературе не рассматривалась.

I. УСЛОВНАЯ ПРОИЗВОДНАЯ БИНАРНОГО ПРЕДИКАТА ПО ПЕРЕМЕННОЙ (ИЛИ ГРУППЕ ПЕРЕМЕННЫХ)

Определение 1 Пусть задан подстановочный предикат p с предметной областью $A = \{a_1, a_2, \dots, a_k\}$ и областью переменных $X = \{X_1, X_2, \dots, X_k\}$; наборы, на которых $p=1$, являются допустимыми с точки зрения конкретно рассматриваемой задачи. Построим новую таблицу, опуская первый столбец исходной таблицы, а затем повторяющиеся строки. Новая таблица соответствует предикату от переменных $X_2, \dots, X_n$, который мы обозначаем $p|X_1$, и называем *производной предиката p по переменной X_1* .

Аналогично мы определяем производную по любой другой переменной, а также по группе переменных. Совокупность наборов предиката p , совпадающих всюду, за исключением переменной X_1 , называется *сопряженным по X_1 множеством наборов*.

Аналогично определяется сопряженное по любой из переменных (или по группе переменных) множество наборов.

При постановке практических задач (задачи логики, задачи поведения автоматов и другие) задаются условия существования состояний - статика системы - и условия изменения переменных (или группы переменных) - динамика системы. Динамика переменных в [3] определяется условиями на дополнительную группу переменных, именуемых дифференциалами. Множество условий, содержащих переменные и их дифференциалы, можно объединить в одно условие, описывающее состояния и возможные переходы из одних состояний в другие. Это так называемое дифференциальное уравнение системы. Решению уравнения соответствует граф системы. В нашем подходе мы будем использовать понятие условной, производной предиката по переменной.

Пусть $p[X_1]$ ($X_1, X_2, \dots, X_n$) - предикат, истинный лишь на тех наборах, которые допускают изменение переменной X_1 в силу конкретно заданных условий функционирования системы.

Определение 2 Условной производной по переменной X_1 называется предикат $p[X_1] | X_1$

Аналогично определяются условная производная по любой другой переменной, а также условная производная по группе переменных. Построение графа задачи (динамической системы) мы проводим в следующем порядке:

для зафиксированного изменения переменной (группы переменных)

задается предикат, не содержащий этой переменной, истинный в силу постановки задачи. Это и будет условная производная. Она позволяет легко расширить полученное решение до предиката, определяющего наборы, на которых происходит изменение системы именно с изменением данной переменной.

1_А Случай бинарного подстановочного предиката —
задача о перевозке Волка, Козы и Капусты.

Рассмотрим более подробно случай бинарного подстановочного предиката, наиболее близкий к понятию булевой функции. Итак, пусть, например,

$A = \{0, 1\}$ – предметный алфавит,

$X = \{P, V, Z, K\}$ – алфавит переменных,

а рассматриваемая система задается парой подстановочных предикатов

$$p1(P, V, Z, K) = P^0 \rightarrow \neg(V^1 Z^1 \vee Z^1 K^1) = 1,$$

$$p2(P, V, Z, K) = P^1 \rightarrow \neg(V^0 Z^0 \vee Z^0 K^0) = 1;$$

здесь X^a – запись предиката узнавания буквы $a \in \{0, 1\}$, а символ $\neg$ означает отрицание предиката, перед которым он ставится.

Система рассматриваемых предикатов и введенные обозначения не случайны, они связаны с задачей о перевозке волка (V), козы (Z) и капусты (K) с берега (0) на берег(1) на лодке, вмещающей двух пассажиров, но так, чтобы ни Волк с Козой, ни Коза с Капустой не оставались без перевозчика на одном берегу. Предикат P^0 принимает значение 1, если перевозчик (P) находится на берегу (0), то есть при $P = 0$, а предикат P^1 принимает значение 1, если перевозчик (P) находится на берегу (1), то есть при $P = 1$. Теперь очевидно, что рассматриваемая система описывает статику задачи, то есть множество допустимых состояний. Решив указанную выше систему уравнений, мы получим множество допустимых наборов, которое можно уже использовать для построения графа.

Обратимся к динамике задачи :

Если перевозчик едет сам, то волк и коза, а также коза и капуста должны находиться на разных берегах реки. Средствами алгебры подстановочных

$$P^0 \vee P^1 = 1,$$

$$V^0 Z^1 K^0 \vee V^1 Z^0 K^1 = 1.$$

Решая систему, получаем 4 набора – вершины графа, а нужные ребра между вершинами находим учитывая, что изменяется только переменная P.

$$X = P^0 V^1 Z^0 K^1 \iff X^* = P^1 V^1 Z^0 K^1 \text{ (т. е. вершины 5=0101 и 13=1101)}$$

$$X = P^0 V^0 Z^1 K^0 \iff X^* = P^1 V^0 Z^1 K^0 \text{ (т. е. вершины 2=0010 и 10=1010)}$$

Если перевозчик перевозит волка, то они будут на одном берегу реки, но коза и капуста должны находиться на разных берегах:

$$P^0 V^0 \vee P^1 V^1 = 1,$$

$$Z^1 K^0 \vee Z^0 K^1 = 1.$$

Решая систему, получаем 4 набора – вершины графа, а нужные ребра между вершинами находим, учитывая, что изменяются переменные P и V одновременно:

$$X = P^0 V^0 Z^0 K^1 \iff X^* = P^1 V^1 Z^0 K^1 \text{ (т. е. вершины 1=0001 и 13=1101)}$$

$$X = P^0 V^0 Z^1 K^0 \iff X^* = P^1 V^1 Z^1 K^0 \text{ (т. е. вершины 2=0010 и 14=1110)}$$

Если перевозчик перевозит козу, то они будут на одном берегу и других ограничений нет:

$$P^0 Z^0 \vee P^1 Z^1 = 1,$$

$$V^0 \vee V^1 = 1,$$

$$K^0 \vee K^1 = 1.$$

Решая систему получаем 8 наборов – вершин графа, а нужные ребра между вершинами находим учитывая, что изменяются переменные P

и Z одновременно:

$$X = P^0 V^0 Z^0 K^0 \iff X^* = P^1 V^0 Z^1 K^0 \text{ (т. е. вершины 0=0000 и 10=1010)}$$

$$X = P^0 V^0 Z^0 K^1 \iff X^* = P^1 V^0 Z^1 K^1 \text{ (т. е. вершины 1=0010 и 11=1011)}$$

$$X = P^0 V^1 Z^0 K^0 \iff X^* = P^1 V^1 Z^1 K^0 \text{ (т. е. вершины 4=0100 и 14=1110)}$$

$$X = P^0 V^1 Z^0 K^1 \iff X^* = P^1 V^1 Z^1 K^1 \text{ (т. е. вершины 5=0101 и 15=1111)}$$

Если перевозчик перевозит капусту, то они будут на одном берегу, но коза и волк должны находиться на разных берегах:

$$P^0 K^0 \vee P^1 K^1 = 1,$$

$$V^0 Z^1 \vee V^1 Z^0 = 1.$$

Решая систему получаем 4 набора – вершины графа, а нужные ребра между вершинами находим, учитывая, что изменяются переменные P

и K одновременно.

$$X = P^0 V^0 Z^1 K^0 \iff X^* = P^1 V^0 Z^1 K^1 \text{ (т. е. вершины 2=0010 и 11=1011)}$$

$$X = P^0 V^1 Z^0 K^0 \iff X^* = P^1 V^1 Z^0 K^1 \text{ (т. е. вершины 4=0100 и 13=1101)}$$

Приведем программу решения задачи, записанную на языке VISUAL PROLOG

```
domains      n=integer m=n*
predicates   oc(n) p1(n,n,n,n) p2(n,n,n,n) p3(n,n,n,n) p4(n,n,n,n)
              s1(m,m,n,n) s2(m,m,n,n) s3(m,m,n,n) s4(m,m,n,n)
clauses      oc(0). oc(1).
```

```

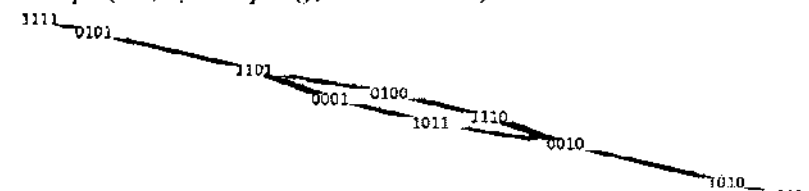
/* ДОПУСТИМЫЕ ИЗМЕНЕНИЯ ДЛЯ КАЖДОЙ ИЗ ПЕРЕМЕННЫХ */
p1(P,V,Z,K):-oc(P),oc(V),oc(Z),oc(K),V<>Z,Z<>K.
/* Если перевозчик едет сам, то волк и коза, а также коза и капуста
должны находиться на разных берегах реки */
p2(P,V,Z,K):-oc(P),oc(V),oc(Z),oc(K),P=V,Z<>K.
/* Если перевозчик перевозит волка, то они будут на одном берегу,
но коза и капуста должны находиться на разных берегах реки */
p3(P,V,Z,K):-oc(P),oc(V),oc(Z),oc(K),P=Z.
/* Если перевозчик перевозит козу, то они будут на одном берегу */
p4(P,V,Z,K):-oc(P),oc(V),oc(Z),oc(K),P=K,V<>Z.
/* Если перевозчик перевозит капусту, то они будут на одном
берегу, но коза и волк должны находиться на разных берегах реки */
/* НАХОЖДЕНИЕ РЕБЕР ГРАФА СОСТОЯНИЙ СИСТЕМЫ */
s1([P1,V1,Z1,K1],[P2,V2,Z2,K2],A,B):-
p1(P1,V1,Z1,K1),p1(P2,V2,Z2,K2),P1<P2,V1=V2,Z1=Z2,K1=K2,
A=8*P1+4*V1+2*Z1+K1,B=8*P2+4*V2+2*Z2+K2.
s2([P1,V1,Z1,K1],[P2,V2,Z2,K2],A,B):-
p2(P1,V1,Z1,K1),p2(P2,V2,Z2,K2),P1<P2,V1<V2,Z1=Z2,K1=K2,
A=8*P1+4*V1+2*Z1+K1,B=8*P2+4*V2+2*Z2+K2.
s3([P1,V1,Z1,K1],[P2,V2,Z2,K2],A,B):-
p3(P1,V1,Z1,K1),p3(P2,V2,Z2,K2),P1<P2,V1=V2,Z1<Z2,K1=K2,
A=8*P1+4*V1+2*Z1+K1,B=8*P2+4*V2+2*Z2+K2.
s4([P1,V1,Z1,K1],[P2,V2,Z2,K2],A,B):-
p4(P1,V1,Z1,K1),p4(P2,V2,Z2,K2),P1<P2,V1=V2,Z1=Z2,K1<K2,
A=8*P1+4*V1+2*Z1+K1,B=8*P2+4*V2+2*Z2+K2.
goal s1([P1,V1,Z1,K1],[P2,V2,Z2,K2],A1,B1).
/* P1=0, V1=0, Z1=1, K1=0, P2=1, V2=0, Z2=1, K2=0, A1=2, B1=10
P1=0, V1=1, Z1=0, K1=1, P2=1, V2=1, Z2=0, K2=1, A1=5, B1=13
2 Solutions */
% s2([P1,V1,Z1,K1],[P2,V2,Z2,K2],A2,B2).
/* P1=0, V1=0, Z1=0, K1=1, P2=1, V2=1, Z2=0, K2=1, A2=1, B2=13
P1=0, V1=0, Z1=1, K1=0, P2=1, V2=1, Z2=1, K2=0, A2=2, B2=14
2 Solutions */

```

```

% s3([P1,V1,Z1,K1],[P2,V2,Z2,K2],A3,B3). /* P1=0, V1=0, Z1=0, K1=0,
P2=1, V2=0, Z2=1, K2=0, A3=0, B3=10
P1=0, V1=0, Z1=0, K1=1, P2=1, V2=0, Z2=1, K2=1, A3=1, B3=11
P1=0, V1=1, Z1=0, K1=0, P2=1, V2=1, Z2=1, K2=0, A3=4, B3=14
P1=0, V1=1, Z1=0, K1=1, P2=1, V2=1, Z2=1, K2=1, A3=5, B3=15
4 Solutions */
% s4([P1,V1,Z1,K1],[P2,V2,Z2,K2],A4,B4).
/* P1=0, V1=0, Z1=1, K1=0, P2=1, V2=0, Z2=1, K2=1, A4=2, B4=11
P1=0, V1=1, Z1=0, K1=0, P2=1, V2=1, Z2=0, K2=1, A4=4, B4=13
2 Solutions */
Записав граф в системе MAPLE и запросив его рисунок получаем
E2 := {{0, 10}, {1, 11}, {1, 13}, {2, 10}, {2, 11}, {2, 14}, {4, 13}, {4, 14},
{5, 13}, {5, 15}};
A2 := [0, 1, 2, 4, 5, 10, 11, 13, 14, 15];
DrawGraph (G2, style = spring, color = black)

```



граф к задаче о перевозке Волка, Козы и Капусты

Из рисунка видны два возможных решения задачи:

- 1) (0000) – (1010) – (0010) – (1011) – (0001) – (1101) – (0101) – (1111) (0000)
– Перевозчик, Волк, Коза и Капуста находятся на «левом (0)» берегу.
(1010) – Перевозчик и Коза переправляются на «правый - (1)» берег.
(0010) – Перевозчик сам переезжает на «левый» берег (изменилась
лишь 1-я координата).
(1011) – Перевозчик с Капустой переезжает на «правый» берег.
(0001) – Перевозчик с Козой переезжает на «левый» берег.
(1101) – Перевозчик с Волком переезжают на «правый» берег.
(0101) – Перевозчик сам переезжает на «левый» берег.
(1111) – Перевозчик с Козой едут на «правый» берег.
Вся компания на правом берегу. Конец процесса.
- 2) (0000) – (1010) – (0010) – (1110) – (0100) – (1101) – (0101) – (1111)

1_Б. Случай небинарного подстановочного предиката — задача
«Ханойская башня»

Рассмотрим знаменитую головоломку «Ханойская башня». Постановка задачи такова:

Дано n дисков разного диаметра (n может быть произвольным; в фольклоре принято $n = 64$, мы же ограничимся тремя дисками). На плоскости жестко закреплены три одинаковых вертикальных штыря. Каждый диск имеет центральное отверстие, достаточное для того, чтобы диск можно было надеть на любой штырь, но не настолько большое, чтобы какой-нибудь другой диск смог пройти сквозь него. Вначале все диски надеты на один штырь таким образом, что наибольший диск лежит в основании башни, а наименьший - на вершине, причем каждый диск лежит на диске большего размера.

Требуется перенести все диски со штыря 0 на штырь 2, используя штырь 1 как вспомогательный, перемещая каждый раз по одному диску с одного штыря на другой. Существенно, чтобы при передвижениях ни один диск не лежал на диске меньшего размера. Кроме того, можно передвигать только диск, лежащий на вершине пирамиды.

Для описания множества состояний задачи используем таблицу истинности предиката с такой областью задания:

$$A = \{0, 1, 2\}$$

$$X = \{D1, D2, D3\}$$

Цифры 0, 1, 2 мы трактуем как обозначения штырей, а D1, D2, D3 - обозначения дисков, в порядке убывания диаметра. Тогда элементарной конъюнкции $D1^2 D2^0 D3^0$ (например!) отвечает такое состояние задачи, при котором на штыре 0 находятся диски D2 и D3, разумеется, меньший на большем, а на штыре 2 диск D1. Между всеми состояниями системы и наборами таблицы устанавливается взаимно-однозначное соответствие.

Обратимся к динамике системы

Перемещение диска D1 возможно только если этот диск на некотором штыре находится в одиночестве и существует штырь, на котором диски отсутствуют. Средствами алгебры подстановочных предикатов это можно записать так:

$$D1^0 \vee D1^1 \vee D1^2 = 1,$$

$$D1^0 D2^0 \vee D1^0 D3^0 \vee D1^1 D2^1 \vee D1^1 D3^1 \vee D1^2 D2^2 \vee D1^2 D3^2 = 0,$$

$$D2^0 D3^0 \vee D2^1 D3^1 \vee D2^2 D3^2 = 1.$$

Условная производная определяется третьим уравнением и дает три набора:

$$D2 = 0, D3 = 0,$$

$$D2 = 1, D3 = 1,$$

$$D2 = 2, D3 = 2,$$

Так как допустимое перемещение всегда говорит о наличии вершины, смежной с данной, мы изменение будем представлять парой вершин X и X^* , как это уже и делалось при решении предыдущей задачи. Мы здесь лишь усовершенствуем эту технику.

Условия допустимости перемещения диска D1 можно записать в виде

$$X2 = X3, X1 \neq X2, X1 \neq X3$$

$X = D1^a D2^b D3^c \iff D1^c D2^b D3^a = X^*$, где $a, b, c \in \{0, 1, 2\}$ и все различны. С учетом сказанного, получаем расширения найденных выше решений:

$$X = D1^1 D2^0 D3^0 \iff D1^2 D2^0 D3^0 = X^* (9 = 100 \text{ смежна } 18 = 200),$$

$$X = D1^0 D2^1 D3^1 \iff D1^2 D2^1 D3^1 = X^* (4 = 011 \text{ смежна } 22 = 211),$$

$$X = D1^0 D2^2 D3^2 \iff D1^1 D2^2 D3^2 = X^* (8 = 022 \text{ смежна } 17 = 122).$$

(Заметим, что десятичные числа здесь представлены троичным кодом).

Перемещение диска D2 возможно, только если этот диск не находится под диском D3, то есть $D2 \neq D3$. В терминах алгебры подстановочных предикатов получаем

$$D1^0 \vee D1^1 \vee D1^2 = 1,$$

$$D2^0 D3^1 \vee D2^1 D3^0 \vee D2^0 D3^2 \vee D2^2 D3^0 \vee D2^1 D3^2 \vee D2^2 D3^1 = 1,$$

Имеем 6 решений - наборов для условной производной - и $6 \cdot 3 = 18$ наборов исходного предиката, дающего нам 9 ребер графа. Расписывать их нахождение не будем, так как дальше мы приведем программу на **VISUAL PROLOG** решающую задачу, а идейная сторона уже достаточно подробно рассмотрена.

Перемещение диска D3 возможно всегда, причем на любой из двух штырей, отличных от того, на котором находится диск в данный момент. Эти перемещения нам дадут еще 27 (по числу всех вершин графа) ребер.

Приводим программу решения задачи на **VISUAL PROLOG**.

```
domains n=integer m=n*
predicates oc(n) pl(n,n,n) s1(m,m,n,n) p2(n,n,n) s2(m,m,n,n) p3(n,n,n) s3(m,m,n,n)
clauses oc(0).oc(1).oc(2).
pl(D1,D2,D3):-      oc(D1),oc(D2),oc(D3), D1oD2, D2=D3.
s1([X1,X2,X3],[Y1,Y2,Y3],Z,T):- oc(X1),oc(X2),oc(X3),oc(Y1),oc(Y2),oc(Y3),
pl(X1,X2,X3),pl(Y1,Y2,Y3),X1<Y1,X2=Y2,X3=Y3,Z=9*X1+3*X2+X3,T=9*Y1+3*
Y2+Y3.

p2(D1,D2,D3):-      oc(D1),oc(D2),oc(D3), D2oD3.
s2([X1,X2,X3],[Y1,Y2,Y3],Z,T):- oc(X1),oc(X2),oc(X3),oc(Y1),oc(Y2),oc(Y3),
p2(X1,X2,X3),p2(Y1,Y2,Y3),X1=Y1,X2<Y2,X3=Y3,Z=9*X1+3*X2+X3,T=9*Y1+3*
Y2+Y3.

p3(D1,D2,D3):-      oc(D1),oc(D2),oc(D3).
s3([X1,X2,X3],[Y1,Y2,Y3],Z,T):-oc{X1},oc(X2),oc(X3),oc(Y1),oc(Y2),oc(Y3),
p3(X1,X2,X3),p3(Y1,Y2,Y3),X1=Y1,X2=Y2,X3<Y3,Z=9*X1+3*X2+X3,T=9*Y1+3*
Y2+Y3.

goal      s1([X1,X2,X3],[Y1,Y2,Y3],Z,T).
/* s2([X1,X2,X3],[Y1,Y2,Y3],Z,T). */
/* s3([X1,X2,X3],[Y1,Y2,Y3],Z,T). */
/*
X1=0, X2=1, X3=1,    Y1=2,Y2=1,Y3=1,    Z=4,T=22
X1=0, X2=2, X3=2,    Y1=1,Y2=2, Y3=2,    Z=8,T=17
X1=1, X2=0, X3=0,    Y1=2,Y2=0,Y3=0,    Z=9,T=18
3 Solutions */
/* X1=0, X2=0, X3=1,    Y1=0,Y2=2,Y3=1,    Z=1,T=7
X1=0, X2=0, X3=2,    Y1=0,Y2=1,Y3=2,    Z=2, T=5
X1=0, X2=1, X3=0,    Y1=0,Y2=2,Y3=0,    Z=3, T=6
X1=1, X2=0, X3=1,    Y1=1,Y2=2, Y3=1,    Z=10,T=16
X1=1, X2=0, X3=2,    Y1=1,Y2=1, Y3=2,    Z=11,T=14
X1=1, X2=1, X3=0,    Y1=1,Y2=2, Y3=0,    Z=12, T=15
```

```
/* X1=0, X2=0, X3=0,    Y1=0, Y2=0, Y3=1,    Z=0, T=1
X1=0, X2=0, X3=0,    Y1=0,Y2=0, Y3=2,    Z=0, T=2
X1=0, X2=0, X3=1,    Y1=0,Y2=0,Y3=2,    Z=1, T=2
X1=0, X2=1, X3=0,    Y1=0,Y2=1,Y3=1,    Z=3, T=4
X1=0, X2=1, X3=0,    Y1=0,Y2=1,Y3=2,    Z=3, T=5
X1=0, X2=1, X3=1,    Y1=0,Y2=1,Y3=2,    Z=4, T=5
X1=0, X2=2, X3=0,    Y1=0,Y2=2,Y3=1,    Z=6, T=7
X1=0, X2=2, X3=0,    Y1=0,Y2=2,Y3=2,    Z=6, T=8
X1=0, X2=2, X3=1,    Y1=0,Y2=2,Y3=2,    Z=7, T=8
X1=1, X2=0, X3=0,    Y1=1, Y2=0, Y3=1,    Z=9, T=10
X1=1, X2=0, X3=0,    Y1=1, Y2=0, Y3=2,    Z=9, T=11
X1=1, X2=0, X3=1,    Y1=1, Y2=0, Y3=2,    Z=10, T=11
X1=1, X2=1, X3=0,    Y1=1, Y2=1, Y3=1,    Z=12,T=13
X1=1, X2=1, X3=0,    Y1=1, Y2=1, Y3=2,    Z=12, T=14
X1=1, X2=1, X3=1,    Y1=1, Y2=1, Y3=2,    Z=13, T=14
X1=1, X2=2, X3=0,    Y1=1, Y2=2, Y3=1,    Z=15, T=16
X1=1, X2=2, X3=0,    Y1=1, Y2=2, Y3=2,    Z=15,T=17
X1=1, X2=2, X3=1,    Y1=1, Y2=2, Y3=2,    Z=16, T=17
X1=2, X2=0, X3=0,    Y1=2,Y2=0,Y3=1,    Z=18, T=19
X1=2, X2=0, X3=0,    Y1=2, Y2=0, Y3=2,    Z=18, T=20
X1=2, X2=0, X3=1,    Y1=2, Y2=0, Y3=2,    Z=19, T=20
X1=2, X2=1, X3=0,    Y1=2, Y2=1, Y3=1,    Z=21,T=22
X1=2, X2=1, X3=0,    Y1=2, Y2=1, Y3=2,    Z=21, T=23
X1=2, X2=1, X3=1,    Y1=2, Y2=1, Y3=2,    Z=22, T=23
X1=2, X2=2, X3=0,    Y1=2, Y2=2, Y3=1,    Z=24, T=25
X1=2, X2=2, X3=0,    Y1=2,Y2=2,Y3=2,    Z=24, T=26
X1=2, X2=2, X3=1,    Y1=2, Y2=2, Y3=2,    Z=25, T=26
27 Solutions */
```

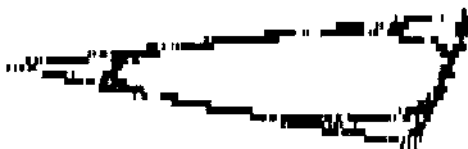
В системе **MAPLE** с учетом вершин и полученных ребер построен граф

H=Digraph([10, 11, [1,0], [0,2], [2,0], [1,2], [1,7], [7, 1], [2, 1], [2, 5], [5,2], [3,4], [4,3], [3, 5], [5,3], [3,6], [6,3], [4,5], [5,4], [4,22], [22,4], [6,7], [7, 6], [6, 8], [8,6], [7, 8], [8, 7], [8,17], [17, 8], [9, 10], [10, 9], [11, 9], [9, 11], [9, 18], [18, 9], [10,11], [11, 10], [10, 16], [16, 10], [11,14], [14, 11], [12, 13],

H:=

Graph 2: a directed unweighted graph with 27 vertices and 78 arc(s)

DrawGraph (G1, style = spring, color = black)

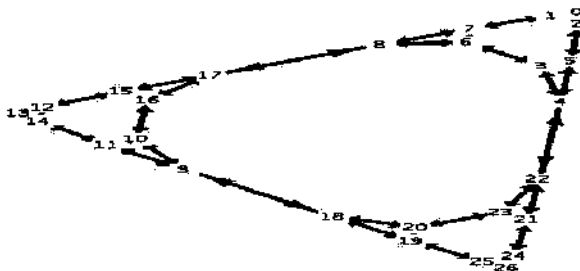


Последовательность переходов со штыря 0 на штырь 2 можно получить, применяя алгоритм Дейкстры:

> *Dijkstras.Algorithm (H, 0, 26)*

[[0,2, 5,4,22,21,24,26], 7]

Чтобы проследить перемещение дисков, нужно записать каждую вершину как состояние задачи. Мы приведем еще один граф, которым более удобно пользоваться:



Граф задачи «Ханойская башня»

Переход от состояния **011** к **211** означает, что большой диск перемещается со штыря 0 на штырь 2. Средний и малый диски остаются на штыре 1. Имея в распоряжении граф игры, мы можем делать и другие выводы. Например, найти последовательность перемещений из любого «правильного» состояния в любое другое «правильное» состояние; заметить, что семи перемещений достаточно, чтобы из любого «правильного» состояния попасть в любое другое «правильное» состояние, и др.

2. ПРОИЗВОДНАЯ ПРЕДИКАТА ПО ПЕРЕМЕННОЙ В ЗАДАННОМ НАПРАВЛЕНИИ

Когда предметное множество содержит **n** элементов, таблицу истинности следует разбить на **n** равных частей, каждая из которых не содержит два одинаковых набора при переменных, отличных от X1 (для переменных X2, X3 ... мы поступаем аналогично).

Пусть, для определенности задан подстановочный предикат:

$$A = \{ 0, 1, 2 \},$$

$$X = \{ X1, X2, X3 \},$$

$$p = (\neg[X1^0])(\neg[X2^1])[X3^2] (+) [X1^0][X2^2][X3^0].$$

Таблица истинности предиката содержит $3^3 = 27$ элементов, а трехэлементные наборы состоят из цифр $\in \{0, 1, 2\}$. К понятию производной предиката по данному направлению (а здесь возможных направлений - три: $n1 = 0_1$, $n2 = 0_2$, $n3 = 1_2$) можно подойти так же, как мы это сделали при определении производной от предиката. Но мы введем здесь ее аналитически:

$$p[X1]_{a_b} = p(a, X2, X3) (+) p(b, X2, X3),$$

($a, b \in \{0, 1, 2\}$, $a < b$, $(+)$ - операция строгой дизъюнкции.) Аналогично вводятся производные по направлению для других переменных.

На языке **VISUAL PROLOG** была составлена программа построения таблицы истинности подстановочного предиката и таблиц истинности производных предикатов по направлению. Это позволило показать использование введенного понятия для решения задачи нахождения оптимального порядка исключения переменных для построения переключательной цепи (заметим, что предлагаемый алгоритм содержит эвристики и потому говорить, что получаем наилучшее из возможных решений, все же нельзя).

Идея алгоритма заключается в том, чтобы найти переменную и направление такие, чтобы производная имела наибольший вес, а вес определяется числом конститuent единицы в соответствующем производном предикате. Суть эвристики в том, что при таком выборе переменной и направления число изменений предиката максимально, а значит, он наиболее сильно зависит от такой производной.

Приведем на языке **VISUAL PROLOG** алгоритм нахождения значений предиката и производной.

```

domains      n = integer
predicates   o(n) oc(n)
             ol(n,n) xor(n,n,n)
             uzn(n,n,n)
             p(n,n,n,n) dpx1(n,n,n,n,n) dpx2(n,n,n,n,n) dpx3(n,n,n,n,n)

clauses
o(0). o(1).
oc(0).oc(1).oc(2)
uzn(X,A,1):- oc(A),X=A.
uzn(X,A,0):- oc(A),X<>A
ol(A, NA):- o(A),NA=1-A.
xor(A,B,Res):- o(A),o(B),Res=A+B-2*A*B.
p(X1,X2,X3,R):- oc(X1),oc(X2),oc(X3),
uzn(X1,0,X10), ol(X10,NX10),
uzn(X2,1,X21), ol(X21,NX21), uzn(X2,2,X22),
uzn(X3,0,X30),uzn(X3,2,X32),
D1= NX10*NX21*X32, D2=X10*X22*X30, xor(D1,D2,R).
dpx1(A,B,X2,X3,Res):- oc(X2),oc(X3),oc(A),oc(B),
A<B,p(A,X2,X3,R1),p(B,X2,X3,R2),xor(R1,R2,Res).

dpx2(A,B,X1,X3,Res):- oc(X1),oc(X3),oc(A),oc(B),
A<B,p(X1,A,X3,R1),p(X1,B,X3,R2),xor(R1,R2,Res).

dpx3(A,B,X1,X2,Res):- oc(X1),oc(X2),oc(A),oc(B),
A<B,p(X1,X2,A,R1),p(X1,X2,B,R2),xor(R1,R2,Res).

goal /*dpx1(A,B,X2,X3,1).*/ /* dpx2(A,B,X1,X3,1). */ dpx3(A,B,X1,X2,1).
/* Производная по переменной X1 в направлении 0-1 так же, как
и в 0-2 приводят к одинаковому результату: предикату  $X2^0X3^2 \vee X2^2X3^0 \vee X2^2X3^2$ . Производная по 1-2 даст тождественно ложный
предикат.
A=0, B=1, X2=0, X3=2
A=0, B=2, X2=0, X3=2
A=0, B=1, X2=2, X3=0
A=0, B=2, X2=2, X3=0
A=0, B=1, X2=2, X3=2
A=0, B=2, X2=2, X3=2
6 Solutions */

```

/* Для переменной X2 имеем:

```

A=0, B=2, X1=0, X3=0
A=1, B=2, X1=0, X3=0
A=0, B=1, X1=1, X3=2
A=1, B=2, X1=1, X3=2
A=0, B=1, X1=2, X3=2
A=1, B=2, X1=2, X3=2
6 Solutions

```

Производная по X2 в направлении 0-1 будет $X1^1X3^2 \vee X1^2X3^2$
в направлении 0-2 - $X1^0X3^0$
в направлении 1-2 - $X1^0X3^0 \vee X1^1X3^2$ */

/* Для переменной X3 :

```

A=0, B=1, X1=0, X2=2
A=0, B=2, X1=0, X2=2
A=0, B=2, X1=1, X2=0
A=1, B=2, X1=1, X2=0
A=0, B=2, X1=1, X2=2
A=1, B=2, X1=1, X2=2
A=0, B=2, X1=2, X2=0
A=1, B=2, X1=2, X2=0
A=0, B=2, X1=2, X2=2
A=1, B=2, X1=2, X2=2
10 Solutions

```

Производная по X3 в направлении 0-1 будет $X1^0X2^2$
в направлении 0-2 будет $X1^0X2^2 \vee X1^1X2^0 \vee X1^1X2^2 \vee X1^2X2^0 \vee X1^2X2^2 =$
 $= X2^2 \vee (\sim X1^0)(X2^0)$

в направлении 1-2 будет $X1^1X2^0 \vee X1^1X2^2 \vee X1^2X2^0 \vee X1^2X2^2$.

Заметим, что наибольшее число конституент единицы содержит производная по переменной X3 в направлении 0-2. Для остаточного предиката

$g(X1,X2) = X2^2 \vee X1^1(\sim X2^1)$ мы повторим процедуру вычисления производных с фиксацией переменной и направления по которым число конституент будет наибольшим.

После исключения переменной X3 задача настолько упрощается, что дальнейшего использования программы не понадобится.

Переключательная цепь кроме обычных в схемах булевых элементов “and”, “or”, “not” содержит еще и элементы узнавания букв (см. [1]).

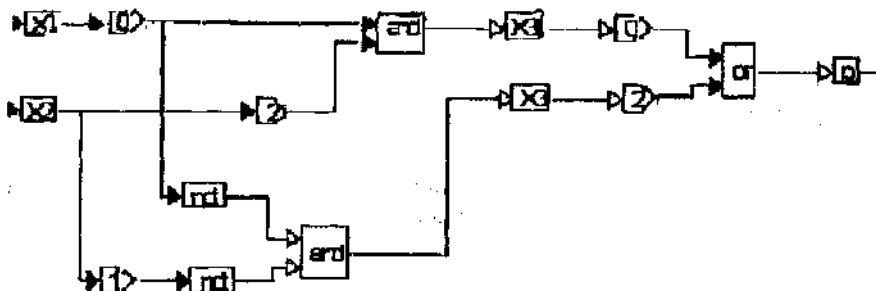


Рисунок переключательной цепи предиката

$$p = (\neg[X1^0])(\neg[X2^1][X3^2] (+) [X1^0][X2^2][X3^0],$$

полученной последовательным исключением переменных.

Литература

1. Шабанов-Кушнarenко Ю. П. Теория интеллекта. Математические средства. //Харьков. ХГУ: «Вища школа», 1984.
2. Бохманн Д. Булево дифференциальное исчисление. // Изв.АН СССР, Техническая кибернетика, 1977.
3. Бохманн Д.,Постхоф Х. Двоичные динамические системы. // М: Энергоатом-Издат, 1986.
4. Адаменко А. Н., Кучуков А. М. Логическое программирование и Visual Prolog. //СПб.: БХВ-Петербург, 2003.
5. Горбатов В.А., Горбатов А.В.,Горбатова М.В. Дискретная математика. // М: АСТ* Астрель, 2003.
6. Любота В. Н. Использование пакета GraphTheory системы MAPLE при минимизации булевых функций. //ДОСДМ, №9 (январь 2010), 40 - 47.