

ИНФОРМАТИКА

УДК 512.5

ИНДЕКСЫ В ДИНАМИЧЕСКОЙ СИСТЕМЕ (B, δ) ДВОИЧНЫХ ВЕКТОРОВ

А. В. Власова

Саратовский государственный университет,
кафедра теоретических основ компьютерной безопасности и криптографии
E-mail: VAnastasiyaV@gmail.com

Описывается алгоритм вычисления индексов состояний динамической системы (B, δ) , состояниями которой являются двоичные векторы, а эволюционная функция δ преобразует вектор с помощью одновременного выполнения следующих действий: начальный 0 (если он есть) заменяется на 1, каждая диграмма 10 – на 01, финальная 1 (если она есть) – на 0. Доказывается корректность алгоритма.

Ключевые слова: динамическая система, эволюционная функция, аттрактор, индекс, плотность, блок.

Indices in Dynamical System (B, δ) of Binary Vectors

A. V. Vlasova

Saratov State University,
Chair of the Theoretical Foundations of Computer Security and Cryptography
E-mail: VAnastasiyaV@gmail.com

An algorithm is proposed for computation of indices of states in dynamical system (B, δ) , whose states are binary vectors and evolutionary function δ transforms vectors according to the following rules: the initial component 0 (if exists) is replaced by 1, every digram 10 by 01, and the final 1 (if exists) by 0. Correctness of the algorithm is proven.

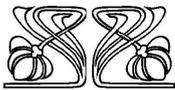
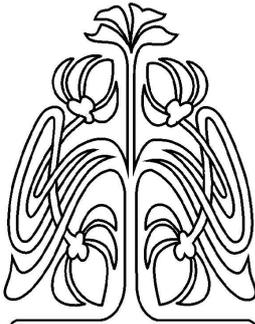
Key words: dynamical system, vector, evolutionary function, attractor, index, density, block.

ВВЕДЕНИЕ

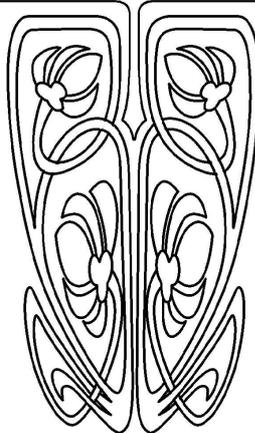
В задачах, связанных с отказоустойчивостью компьютерных сетей, заметное место занимают графовые модели, в которых отказы процессоров интерпретируются как удаление соответствующих вершин, а отказы сетевых каналов — как удаление дуг. Здесь можно выделить следующую конструкцию, получившую и самостоятельное значение в теории графов — ациклический граф с заданной структурой источников и стоков [1]. В модели [1] в качестве механизма восстановления работоспособности сети предлагается так называемая SER-динамика ациклических графов. Это позволяет использовать при изучении модельных графов идеи и методы теории конечных динамических систем, и, в частности, динамических систем двоичных векторов (см., например, [2, 3]) — когда имеется естественная двоичная кодировка графов рассматриваемого класса.

Под *конечной динамической системой* понимается пара (S, δ) , где S — конечное непустое множество, элементы которого называются *состояниями системы*, $\delta: S \rightarrow S$ — отображение множества состояний в себя, называемое *эволюционной функцией системы*.

Каждой конечной динамической системе сопоставляется *карта* — граф с множеством вершин S и дугами, проведенными из каждой



**НАУЧНЫЙ
ОТДЕЛ**





вершины $s \in S$ в вершину $\delta(s)$. Этот граф является функциональным, т.е. из каждой вершины выходит точно одна дуга. Компоненты связности графа, задающего динамическую систему, называются её бассейнами.

Каждый бассейн представляет собой контур с входящими в него деревьями. Контур называется предельным циклом, или аттрактором.

Одной из основных характеристик состояний конечной динамической системы является индекс — расстояние до аттрактора того бассейна, которому принадлежит состояние. Программа [4] предназначена для исследования эволюционных параметров состояний (индекс, недостижимость, ветвление) в динамических системах, состояниями которых являются двоичные векторы, представленные такими графами, как цепи, циклы и пальмы. В настоящей работе описываются индексы динамической системы двоичных векторов, порожденных такими графами, как цепи (см. [5–7]).

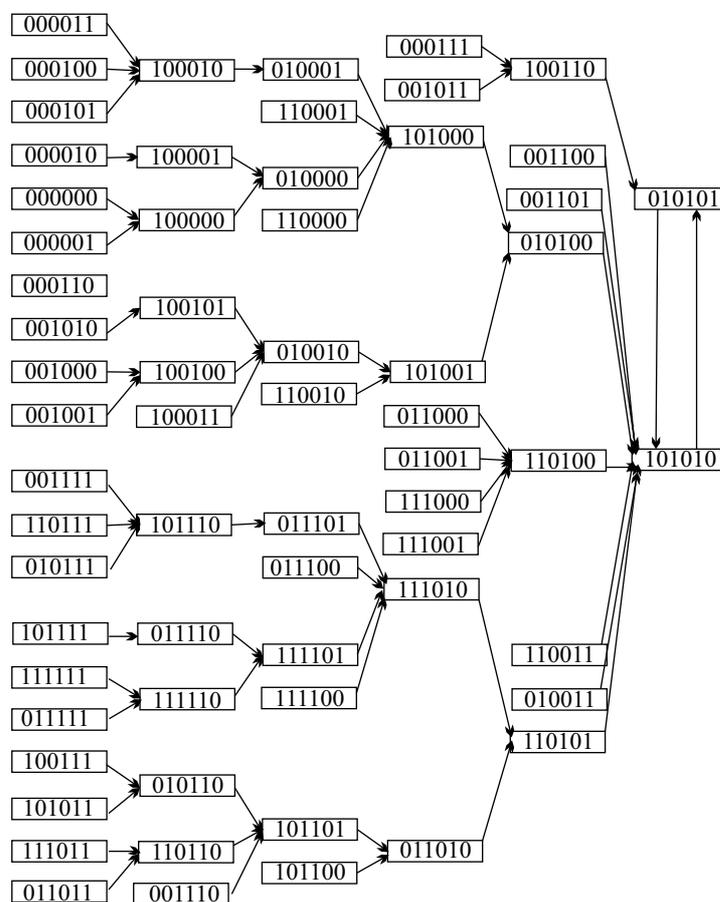
1. ОПИСАНИЕ СИСТЕМЫ

Пусть $B = \bigcup_{n=1}^{\infty} B^n$, где через B^n , $n > 0$, обозначим множество всех двоичных векторов размерности n . Пусть состоянием динамической системы в данный момент времени является вектор $v \in B$. Тогда в следующий момент времени она окажется в состоянии $\delta(v)$, описываемом следующими правилами:

- 1) если первой компонентой в v является 0, то первой компонентой в $\delta(v)$ будет 1;
- 2) если в составе v имеются диграммы (две соседние компоненты) вида 10, то в $\delta(v)$ каждая из них заменяется на 01;
- 3) если последней компонентой в v является 1, то последней компонентой в $\delta(v)$ будет 0;
- 4) других отличий между v и $\delta(v)$ нет.

Замечание. Вышеперечисленные правила применяются одновременно.

Эта динамика для двоичных векторов определена в [2]. Каждое состояние размерности n при динамике переходит в состояние также размерности n . Таким образом, система B в зависимости от n разбивается на подсистемы (B^n, δ) . На рисунке показана карта динамической системы (B^6, δ) .



Карта динамической системы (B^6, δ)

2. АТТРАКТОРЫ В СИСТЕМЕ (B, δ)

Через $p(v)$ обозначается плотность вектора v , т.е. количество пар совпадающих соседних компонент в нем. Например, $p(111111) = 5$, $p(101010) = 0$, $p(111011) = 3$, $p(\delta(111011)) = p(110110) = 2$. Очевидно, что для $v \in B^n$ будет $0 \leq p(v) \leq n - 1$.

Аттракторы динамической системы (B, δ) были описаны в работе [5], где доказываются следующие три теоремы.

Теорема 1. Для системы (B, δ) при любом v справедливо неравенство $p(\delta(v)) \leq p(v)$.

Теорема 2. Для системы (B, δ) верно $(\forall v)(p(v) \neq 0 \implies (\exists k)(p(\delta^k(v)) < p(v))$.



Теорема 3. При любом $n \geq 2$ система (B^n, δ) имеет единственный бассейн и аттрактор, представляющий собой двухэлементный цикл, образуемый состояниями $(01)^{\lfloor n/2 \rfloor} 0$ и $(10)^{\lfloor n/2 \rfloor} 1$ при нечётном n и состояниями $(01)^{n/2}$ и $(10)^{n/2}$ при чётном n .

3. АЛГОРИТМ ВЫЧИСЛЕНИЯ ИНДЕКСА СОСТОЯНИЯ В СИСТЕМЕ (B, δ)

Блок — это подряд стоящие нули (0-блоки) или единицы (1-блоки), в количестве ≥ 2 . Длина блока — это число нулей (единиц) в нем, уменьшенное на 1.

Введем обозначения:

r — порядковый номер последней компоненты последнего 1-блока;

k — количество компонент вектора до первого блока;

f — количество компонент вектора между последним блоком и концом вектора;

i — индекс рассматриваемого состояния;

s — порядковый номер первой компоненты первого 0-блока;

p_0, p_1 — суммы длин рассматриваемых 0-блоков и 1-блоков соответственно;

УНГ — указатель начала группы;

УРС — указатель рассмотрения состояния.

Алгоритм вычисления индекса состояния системы (B, δ) .

Индекс состояния системы (B, δ) высчитывается исходя из его вида как вектора.

I. Если вектор v содержит в себе блоки и все они состоят из 1, то $i(v) = r - 1$;

II. Если вектор v содержит в себе блоки и все они состоят из 0, то $i(v) = n - s$;

III. Если вектор v содержит в себе 1-блоки, после которых идут 0-блоки, то $i(v) = n - s$ при $r - 1 < n - s$, иначе $i(v) = r - 1$;

IV. Если вектор v содержит в себе 0-блоки, после которых идут 1-блоки, то $i(v) = r - 1$ при $p_0 < p_1$, $i(v) = (r - s + 1)/2 - 1$ при $p_0 = p_1$, $i(v) = n - s$ при $p_0 > p_1$;

V. Если вектор v содержит в себе 0-блоки и 1-блоки в произвольном порядке, отличном от описанных в пунктах I — IV алгоритма, то, просматривая его слева направо, выполняем следующие действия:

1. УНГ и УРС ставятся в начале вектора. Если до первого 0-блока есть 1-блоки, то первой группой будет являться отрезок вектора от начальной компоненты до последней компоненты того 1-блока, который стоит перед первым 0-блоком включительно; УНГ и УРС ставятся между последней компонентой того 1-блока, который стоит перед первым 0-блоком, и следующей компонентой.

2. Если вектор уже полностью разбит на группы или если 1-блоков больше нет, то группа заканчивается на последней компоненте вектора и переходим в пункт 6, иначе $p_0 := 0, p_1 := 0$.

3. $p_0 := p_0 +$ сумма длин 0-блоков до первого (следующего) 1-блока от УРС; УРС ставится между последним из таких 0-блоков и следующей компонентой.

4. $p_1 := p_1 +$ сумма длин 1-блоков до первого (следующего) 0-блока от УРС; УРС ставится между последним из таких 1-блоков и следующей компонентой.

5. Если $p_0 \leq p_1$, то группа заканчивается на последней компоненте того 1-блока, который стоит перед УРС, и идём в пункт 2, УНГ ставится между последней компонентой этой группы и следующей компонентой; а если 0-блоков больше нет, то группа заканчивается на последней компоненте состояния, и идём в пункт 6. Если $p_0 > p_1$ и ещё остались 1-блоки, то идём в пункт 3, а если 1-блоков больше нет, то группа заканчивается на последней компоненте состояния и идём в пункт 6.

6. В каждой группе подсчитываем p_0 и p_1 и помечаем группы знаками «-», «=», «+», если в них $p_0 > p_1, p_0 = p_1, p_0 < p_1$ соответственно. Объединяем рядом стоящие группы с одинаковыми пометками «-» или «+» в одну, при этом если между «+»-группами встречаются «=»-группы, то их также объединяем в «+»-группу. Пусть h — общее количество групп. Считаем групповые индексы $i_j, 1 \leq j \leq h$, согласно следующим правилам: в «-»-группе $i_j = n - s_j$, в «=»-группе $i_j = (r_j - s_j + 1)/2 - 1$, в «+»-группе $i_j = r_j - 1$. Тогда $i(v) = \max_{1 \leq j \leq h} i_j$.

Теорема 4 (Корректность алгоритма вычисления индекса состояния). Алгоритм вычисления индекса состояния корректен.



Доказательство. I. Если вектор v содержит в себе блоки и все они состоят из 1, то $i(v) = r - 1$. Рассмотрим эволюцию таких состояний в зависимости от количества 1-блоков.

I.1. Вектор имеет единственный 1-блок. Рассмотрим эволюцию таких состояний в зависимости от расположения 1-блока.

а) вектор содержит единственный 1-блок, который расположен в начале вектора.

В табл. 1 приведена схематичная эволюция таких состояний.

Таблица 1

Таким образом, $i(v) = r - 1$.

Заметим, что длина 1-блока на каждом очередном шаге уменьшается на единицу за счёт поглощения самой левой компоненты и за ним следуют чередующиеся нули и единицы, пока все его компоненты не поглотятся.

б) вектор содержит единственный 1-блок, который расположен не в начале вектора.

В табл. 2 приведена схематичная эволюция таких состояний.

Таким образом, $i(v) = k + p_1 = r - 1$.

Заметим, что 1-блок на каждом очередном шаге смещается влево на одну компоненту, пока не встанет в начало состояния, затем длина 1-блока на каждом очередном шаге уменьшается на единицу за счёт поглощения самой левой компоненты и за ним следуют чередующиеся нули и единицы, пока все его компоненты не поглотятся.

I.2. Вектор имеет несколько блоков, и все они состоят из 1.

Из рассмотрения пункта I.1 можно заключить, что 1-блоки на каждом очередном шаге смещаются влево на одну компоненту, пока не встанут в начало состояния, затем длина 1-блока на каждом очередном шаге уменьшается на единицу за счёт поглощения самой левой компоненты и за ним следуют чередующиеся нули и единицы, пока все компоненты 1-блока не поглотятся. Очевидно, что в данном случае последним поглотится самый правый 1-блок и тогда $i = r - 1$.

Таким образом, рассмотрев все возможные ситуации, делаем вывод, что пункт I алгоритма корректен.

II. Если вектор v содержит в себе блоки, и все они состоят из 0, то $i(v) = n - s$. Рассмотрим эволюцию таких состояний в зависимости от количества 0-блоков.

II.1. Вектор имеет единственный 0-блок. Рассмотрим эволюцию таких состояний в зависимости от расположения 0-блока.

а) вектор содержит единственный 0-блок, который расположен в конце вектора.

Схематичная эволюция таких состояний: $101000...000 \rightarrow_1 010100...000 \rightarrow_2 101010...000 \rightarrow_3 \dots \rightarrow_{p_0-1} 01010...100 \rightarrow_{p_0} 10101...010$.

Таким образом, $i(v) = p_0 = n - s$. Можно заметить, что длина 0-блока на каждом очередном шаге уменьшается на единицу за счёт поглощения самой правой компоненты и перед ним идут чередующиеся нули и единицы, пока все его компоненты не поглотятся.

б) вектор содержит единственный 0-блок, который расположен не в конце вектора.

Схематичная эволюция таких состояний: $10...1010000...00010...101 \rightarrow_1 01...0101000...00001...010 \rightarrow_2 10...101010...00000...101 \rightarrow_3 \dots \rightarrow_{f-1} 10...10101...01000...001 \rightarrow_f 01...01010...10100...000 \rightarrow_{f+1} 10...10101...01010...000 \rightarrow_{f+2} \dots \rightarrow_{f+p_0-1} 10...10101...01010...100 \rightarrow_{f+p_0} 01...01010...10101...010$.

Таким образом, $i(v) = f + p_0 = n - s$.

Заметим, что 0-блок на каждом очередном шаге смещается вправо на одну компоненту, пока не встанет в конец вектора, затем длина 0-блока на каждом очередном шаге уменьшается на единицу за счёт поглощения самой правой компоненты и перед ним стоят чередующиеся нули и единицы, пока все его компоненты не поглотятся.

II.2. Вектор имеет несколько блоков, и все они состоят из 0.

№ шага	Состояние
0	111...11101...010
1	111...11010...101
2	111...10101...010
...	...
$r - 2$	110...10101...010
$r - 1$	101...01010...101

Таблица 2

№ шага	Состояние
0	01...01011...111101...010
1	10...10111...1111010...101
2	01...01111...1110101...010
3	10...11111...1101010...101
...	...
$k - 1$	01...11110...10101...010
k	11...11101...01010...101
$k + 1$	11...11010...10101...010
$k + 2$	11...10101...01010...101
...	...
$k + p_1 - 1$	11...01010...10101...010
$k + p_1$	10...10101...01010...101



Из рассмотрения пункта II.1 можно заключить, что каждый 0-блок на очередном шаге смещается вправо на одну компоненту, пока не достигает конца вектора, затем длина 0-блока на каждом очередном шаге уменьшается на единицу за счёт поглощения самой правой компоненты и перед ним следуют чередующиеся нули и единицы, пока все компоненты 0-блока не поглотятся. Очевидно, что в данном случае последним поглотится самый левый 0-блок, и тогда индекс будет равен уменьшенному на единицу порядковому номеру с конца вектора первой компоненты самого левого 0-блока, или $i = n - s$.

Таким образом, рассмотрев все возможные ситуации, делаем вывод, что пункт II алгоритма корректен.

III. Если вектор содержит в себе 1-блоки, после которых идут 0-блоки, то $i = n - s$ при $r - 1 < n - s$, иначе $i = r - 1$.

Исходя из доказательств предыдущих пунктов, можно заключить, что в данном случае на каждом шаге эволюции одновременно 0- и 1-блоки начнут движение вправо или влево соответственно, пока не достигнут своих концов вектора и не начнут поглощаться, т.е. самым последним поглотится тот блок, который дальше всего будет стоять от своего края. Таким образом, нужно определить порядковый номер от конца состояния первой компоненты первого 0-блока, определить порядковый номер последней компоненты последнего 1-блока, выбрать максимальный из них и считать согласно пунктам I или II, т.е. если $r - 1 < n - s$, то $i = n - s$, иначе $i = r - 1$. Таким образом, пункт III алгоритма корректен.

IV. Если вектор содержит в себе 0-блоки, после которых идут 1-блоки, то $i = r - 1$ при $p_0 < p_1$, $i = (r - s + 1)/2 - 1$ при $p_0 = p_1$, $i = n - s$ при $p_0 > p_1$.

Обозначим через g количество компонент вектора между последним 0-блоком и первым 1-блоком.

IV.1. Рассмотрим сначала эволюцию таких состояний, включающих в себя один 0-блок, после которого есть 1-блок, и в зависимости от их значений p_0 и p_1 :

а) состояния, для которых $p_0 < p_1$.

В табл. 3 приведена схематичная эволюция таких состояний. Таким образом, $i(v) = r - 1$.

Заметим, что в этом конкретном случае сначала на каждом шаге эволюции 0-блок начинает двигаться вправо, 1-блок начинает двигаться влево за счёт поглощения компонент, стоящих между блоками; когда блоки оказываются стоящими рядом, то их длины начинают уменьшаться на единицу за счёт поглощения компонент друг друга, пока 0-блок полностью не поглотится. Таким образом, в результате остаётся 1-блок и чередующиеся нули и единицы, дальнейшая эволюция была описана в пункте I, т.е. получается, что в процессе эволюции для подсчёта индекса состояния имеет значение только 1-блок, а точнее порядковый номер

его последней компоненты, так как при эволюции его последняя компонента просто смещается влево на каждом шаге эволюции, пока из всего 1-блока не останется она одна, т.е. $i(v) = r - 1$.

б) состояния, для которых $p_0 = p_1$.

Из рассуждений пункта IV.1.а получаем, что при эволюции 0-блок и 1-блок начнут движение навстречу друг другу, пока не окажутся рядом, а затем начнут поглощать друг друга с каждым следующим шагом эволюции, пока от них не останется по одной компоненте, причём это уже будет вектор, полностью состоящий из чередующихся нулей и единиц. Таким образом, в данном случае индекс будет равен количеству компонент блока и половине количества компонент между блоками.

в) состояния, для которых $p_0 > p_1$.

Здесь ситуация будет аналогична пункту IV.1.а, только в результате остаётся один 0-блок и чередующиеся нули и единицы, дальнейшая эволюция описана в пункте II, т.е. $i(v) = n - s$.

Таблица 3

№ шага	Состояние
0	01...01000...00101...01011...111101...01
1	10...10100...00010...10111...111010...10
2	01...01010...00001...01111...110101...01
...	...
$g/2$	01...0101000...00001111...11110101...01
$g/2 + 1$	10...1010100...00001111...11101010...10
$g/2 + 2$	01...0101010...00001111...11010101...10
...	...
$g/2 + p_0 - 1$	01...0101010011...11010101...10
$g/2 + p_0$	10...1010101011...10101010...01
...	...
$r - 2$	1101...01010...101
$r - 1$	1010...10101...010



IV.2. Рассмотрим теперь состояния, включающие в себя несколько 0-блоков, после которых идут 1-блоки, также в зависимости от значений p_0 и p_1 в векторе.

а) состояния, для которых $p_0 < p_1$.

Из пункта IV.1.а получаем, что здесь на каждом шаге эволюции 0-блоки начинают двигаться вправо, 1-блоки начинают двигаться влево за счёт поглощения компонент, стоящих между блоками; когда они оказываются стоящими рядом, то блоки начинают уменьшаться на единицу за счёт поглощения компонент друг друга, пока один из блоков полностью не поглотится, после чего опять продолжают сдвиги блоков навстречу друг другу, пока не поглотится самый последний 0-блок. Таким образом, останутся только 1-блоки и чередующиеся нули и единицы, дальнейшая эволюция описана в пункте I. Таким образом, получается, что в процессе эволюции для подсчёта индекса состояния имеет значение только последний 1-блок, а точнее порядковый номер его последней компоненты, так как при эволюции его последняя компонента просто смещается влево на каждом шаге эволюции, пока из всего этого 1-блока не останется она одна, т.е. $i(v) = r - 1$.

б) состояния, для которых $p_0 = p_1$.

Из рассуждений пунктов IV.1.б и IV.2.а получаем, что при эволюции 0-блоки и 1-блоки начнут движение навстречу друг другу, пока не окажутся рядом, а затем начнут поглощать друг друга с каждым следующим шагом эволюции, пока не поглотится один из блоков, после чего продолжится аналогичное движение, пока рядом не окажутся последние оставшиеся 0-блок и 1-блок, которые начнут поглощать друг друга с каждым следующим шагом эволюции, пока от них не останутся по одной компоненте, причём это уже будет вектор полностью состоящий из чередующихся нулей и единиц. Таким образом, $i = (r - s + 1)/2 - 1$.

Покажем, что количество компонент от первой компоненты первого 0-блока до последней компоненты последнего 1 блока, т.е. $r - s + 1$, всегда будет чётно. Докажем это индукцией по p_0 .

Базис индукции: $p_0 = 1$. Здесь получается, что длины 0- и 1-блоков равны 1 и в сумме дают чётное число, количество компонент между этими блоками также чётно, так как это чередующиеся нули и единицы, а после 0-блока обязательно будет стоять 1 и перед 1-блоком будет стоять 0, в итоге получаем чётное количество компонент от первой компоненты первого 0-блока до последней компоненты последнего 1-блока.

Шаг индукции. Предположим, что при любом $p_0 \leq m \in \mathbf{N}$ количество компонент от первой компоненты первого 0-блока до последней компоненты последнего 1 блока, т.е. $r - s + 1$, будет чётно. Покажем, что данное условие выполняется при $p_0 = m + 1$. Рассмотрим здесь несколько ситуаций.

1) длины первого 0-блока и последнего 1-блока не равны 1. Тогда отсекаем у этих блоков по одному элементу, получаем случай $p_0 = m$, для которого требуемое условие выполняется, а отсеки мы две компоненты, т.е. в итоге получаем чётное число и выполнение нужного условия.

2) длина первого 0-блока или длина последнего 1-блока равна 1. Например, длина первого 0-блока равна 1. Тогда отсекаем у этих блоков по одной компоненте, получаем случай $p_0 = m$, для которого требуемое условие выполняется. От 0-блока осталась одна компонента, до следующего 0-блока идёт нечётное количество компонент, а отсеки мы две компоненты, т.е. в итоге получаем чётное число и выполнение нужного условия.

3) длины первого 0-блока и последнего 1-блока равны 1. Тогда отсекаем у этих блоков по одному элементу, получаем случай $p_0 = m$, для которого требуемое условие выполняется. От 0-блока и 1-блока осталось по одной компоненте, до следующего 0-блока идёт нечётное количество компонент, от предыдущего 1-блока идёт нечётное количество компонент, а отсеки мы две компоненты, т.е. в итоге получаем чётное число и выполнение нужного условия.

в) состояния, для которых $p_0 > p_1$.

Здесь ситуация будет аналогична пункту IV.2.а, только в состоянии в конце концов остаётся один 0-блок и чередующиеся нули и единицы, дальнейшая эволюция описана в пункте II, т.е. $i(v) = n - s$.

V. Вектор состоит из 0-блоков и 1-блоков в произвольном порядке.

Из доказательств предыдущих пунктов можно заключить, что при эволюции такого состояния 0-блоки будут сдвигаться вправо, при этом если они будут встречаться с 1-блоками, то длина 0-блока будет уменьшаться с очередным шагом эволюции, т.е. если есть подряд стоящие 0-блоки, то они или все поглотятся, если следующие за ними подряд стоящие 1-блоки в сумме имеют равную



или большую длину, или поглотят сами следующие за ними подряд стоящие 1-блоки и продолжают сдвиг вправо, встречая очередные 1-блоки, если сумма длин 0-блоков будет больше суммы длин 1-блоков. С точки зрения 1-блоков ситуация получается аналогичная, но так как на очередном шаге эволюции движение 0- и 1-блоков происходит одновременно, то, не теряя общности, будем рассматривать этот случай с точки зрения движения 0-блоков, т.е. ситуаций, когда 0-блоки будут полностью поглощаться 1-блоками, что возможно в ситуациях, когда суммы длин 1-блоков больше или равны сумме длин 0-блоков, или 0-блоки поглотят 1-блоки, что возможно, когда суммы длин 1-блоков меньше суммы длин 0-блоков. В пунктах V.1–V.5 алгоритма как раз описывается процедура деления состояния на группы, в которых суммы длин 0-блоков или меньше сумм длин 1-блоков, или равны, или больше (в этом случае это может быть только крайняя справа группа). В итоге получаем вектор, разделенный на группы, причём слева будет «+»-группа, справа «-»-группа, а между ними могут быть ещё «=>»-группы. Так как при динамике эти группы (блоки) начинают свои движения одновременно, то индекс будет равен максимальному из групповых индексов. А это описано в пункте V.6 алгоритма подсчёта индекса, т.е. алгоритм верен. \square

Сложность алгоритма равна $O(n)$, где n — количество компонент в состоянии.

Пример. 1011101001101000111010001

Вектор подходит только под V пункт алгоритма. Сначала получаем деление вектора на группы (в результате выполнения переходов $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 6$): [10111][010011][01000111][010001]. Подсчитываем групповые индексы: $i_1 = r_1 - 1 = 5 - 1 = 4$ («+»-группа); $i_2 = (r_2 - s_2 + 1)/2 - 1 = (11 - 8 + 1)/2 - 1 = 1$ («=>»-группа); $i_3 = (r_3 - s_3 + 1)/2 - 1 = (19 - 14 + 1)/2 - 1 = 2$ («=>»-группа); $i_4 = n - s_4 = 25 - 22 = 3$ («-»-группа). Таким образом, $i(v) = 4$.

Эволюция состояния:

1011101001101000111010001 \rightarrow 0111010101010100110101000 \rightarrow 1110101010101010101010100 \rightarrow 1101010101010101010101010 \rightarrow 1010101010101010101010101 (аттрактор), т.е. $i(v) = 4$.

Одним из параметров бассейна динамической системы является его *глубина* — наибольший из индексов состояний.

Следствие (глубина бассейна). При любом $n \geq 2$ система (B^n, δ) имеет глубину бассейна, равную $n - 1$.

Доказательство. Рассмотрим формулы вычисления индекса состояния в каждом пункте алгоритма.

I. $i(v) = r - 1$. Здесь значение индекса будет максимальным при $r = n$, например для состояния $(1)^n$.

II. $i(v) = n - s$. Здесь значение индекса будет максимальным при $s = 1$, например для состояния $(0)^n$.

В пунктах III, IV, V ситуация будет аналогична, при этом очевидно, что значение индекса в состоянии, для которого $p_0 = p_1$ (или в «=>»-группе для V пункта), всегда будет меньше $n - 1$.

Таким образом, максимальный индекс в системе (B^n, δ) равен $n - 1$. \square

Библиографический список

1. *Barbosa V. C.* An atlas of edge-reversal dynamics. L.: Chapman&Hall/CRC, 2001. 372 с.
2. *Салий В. Н.* Об одном классе конечных динамических систем // Вестн. Томск. гос. ун-та. Приложение. 2005. № 14. С. 23–26.
3. *Colon-Reyes O., Laubenbacher R., Pareigis B.* Boolean monomial dynamical systems // Ann. Comb. 2004. Vol. 8. P. 425–439.
4. *Власова А. В.* Исследование эволюционных параметров в динамических системах двоичных векторов // Свидетельство о государственной регистрации программы для ЭВМ № 2009614409, выданное Роспатентом. Зарегистрировано в Реестре программ для ЭВМ 20 августа 2009 г.
5. Аттракторы в динамических системах двоичных векторов / А. В. Власова; Саратов. гос. ун-т. Саратов, 2010. 19 с. Деп. в ВИНТИ 23.06.2010, № 392–В2010.
6. *Власова А. В.* Ветвления в динамической системе n -мерных двоичных векторов // Инновационные технологии XXI века в управлении, информатике и образовании. Нальчик, 2008. С. 109–112.
7. Об одной динамической системе / А. В. Власова; Саратов. гос. ун-т. Саратов, 2007. 17 с. Деп. в ВИНТИ 17.12.07, № 1181–В2007.