Следствием теоремы 1 и данного метода синтеза является следующая теорема.

Теорема 2. Система функций $\{\vee\}\cup T^{(1)}\cup B^{(1)}$ s-полна в Q.

Синтез s-формул для монотонных функций в базисах, содержащих функции из $\{\vee\}\cup T^{(1)}$

Введем в рассмотрение специальный класс B_M монотонных функций: произвольная функция f из M принадлежит B_M, если и только если найдутся элементы e в E, c в C и d в D_f такие, что $e<c$, $V_f=\{e, c\}$ и для любого элемента d' из D_f верно $f(d')=c \Leftrightarrow d'\geq d$.

Для произвольных e из E, c из C таких, что $e<c$, и для произвольного d из C^m при некотором m обозначим $f_d^{e,c}$ функцию из B_M, определенную на C^m, такую, что для каждого d' из C^m: $f(d')=c$ если $d'\geq d$, и $f(d')=e$ в противном случае.

Лемма 9. Пусть f, f_0 – функции из M и $f\geq f_0$. Тогда найдется число r, функции $s_1,..., s_r$ из S и функции из $f_1,..., f_r$ из B_M такие, что $f=f_0\vee s_1f_1\vee s_2f_2...\vee s_rf_r$.

Доказательство. Случай $f=f_0$ – тривиален. Пусть $f\neq f_0$. Тогда $f>f_0$. Обозначим $D_0=\{d\in D_f | f(d)>f_0(d)\}$. Пусть $D_0=\{d_1,..., d_r\}$. Зафиксируем указанную нумерацию элементов в D_0. Для произвольного i в $\{1,..., r\}$ пусть e_i – произвольный элемент из множества $f_0(d_i)$ и $h_i=f(d_i)$. Для каждого элемента i из $\{1,..., r\}$ возьмем функцию $f_i: D_f\to\{e_i, h_i\}$ в B_M такую, что $f_i=f_{di}^{ei, hi}$. Пусть при любом i из $\{1,..., r\}$ s_i – функция из S такая, что $s_i(e_i)=0$. Для любого $D\subseteq D_f$ построим функцию $f^D: D_f\to V_f$ для произвольного d' в D_f, положив $f^D(d')=f_0(d')+\sum f_i(d)$, где суммирование ведется по всем d из D, содержащимся в элементе d'. Заметим, что $f^D=f$ при $D=D_0$ и $f^{\varnothing}=f_0$. Пусть $i\in\{1,..., r\}$ и $D\subseteq D_0-d_i$. Тогда $f^{D\cup di}=f^D\vee s_if_i$. Покажем это.

Пусть $d\in D_f$ и d не реализуется d_i. Тогда $f_i(d)=e_i$ и, следовательно, $f^D\vee s_if_i=s_i^{-1}(s_if^D(d)\vee s_if_i(d))=s_i^{-1}(s_if^D(d)\vee s_ie_i)=s_i^{-1}(s_if^D(d)\vee 0)=s_i^{-1}(s_if^D(d))=f^D(d)=f^{D\cup di}(d)$. Пусть $d\geq d_i$. Тогда $f_i(d)=h_i>e_i$, $f^D(d)\geq f_0(d)\geq f_0(d_i)=h_i>e_i$ и $f^D(d)\vee s_if_i(d)=s_i^{-1}(s_if^D(d)\vee s_if_i(d))\leq s_i^{-1}(s_if^D(d)+s_if_i(d))=f^D(d)+f_i(d)=f^D(d)+f(d)=f^{D\cup di}(d)$. С другой стороны, так как $f^D\vee s_if_i=s_i^{-1}(s_if^D(d)\vee s_if_i(d))\geq s_i^{-1}\times\times(s_ie_i\vee s_if_i(d))\geq s_i^{-1}(0\vee s_if_i(d))=f_i(d)$ и $f^D\vee s_if_i=s_i^{-1}(s_if^D(d)\vee\vee s_if_i(d))\geq s_i^{-1}(s_if^D(d)\vee s_ie_i)\geq s_i^{-1}(s_if^D(d)\vee 0)=f^D(d)$, то $f^D(d)\vee s_if_i(d)\geq f^D(d)+f_i(d)=f^D(d)+f(d)=f^{D\cup di}(d)$. Следовательно, $f^D(d)\vee\vee s_if_i(d)=f^{D\cup di}(d)$, откуда получаем: $f^{d1}=f_0\vee s_1f_1, f^{d1\cup d2}=f_0\vee\vee s_1f_1\vee s_2f_2,... f=f^{d1\cup d2\cup...\cup dr}=f_0\vee s_1f_1\vee s_2f_2...\vee s_rf_r$. Лемма доказана.

Лемма 10. Пусть f – произвольная функция из $B_M^{(m)}$. Тогда существуют функции $s_0,..., s_m$ в $B_M^{(1)}$ такие, что $f(x_1,..., x_m)=s_0(s_1x_1\vee...\vee s_mx_m)$.

Доказательство. $f\in B_M$, поэтому существуют элементы e в E, c в C, $d=(d_1,..., d_m)$ в D_f такие, что $e<c$ и f принимает значение c на аргументах, реализуемых d, и значение e на остальных аргументах.

Пусть $h=\{k-1,0\}$ – элемент из C. Положим $s_0=f_h^{e, c}$ и $s_i=f_{di}^{k-1, h}$ для всех $i\in\{1,..., m\}$. Можно показать, что $f(x_1,..., x_m)=s_0(s_1x_1\vee...\vee s_mx_m)$. Лемма доказана.

Леммы 9, 10 позволяют нам сформулировать следующий **метод синтеза s-формул для функций из M в базисе $\{\vee\}\cup T^{(1)}\cup B_M^{(1)}$**.

1. Любым известным способом получим реализацию f_0 функции f.

2. Способом, описанным в доказательстве леммы 9, находим функции $s_1,..., s_r$ в $T^{(1)}$, функции в $f_1,..., f_r$ из B_M и представляем f в форме $f=f_0\vee s_1f_1\vee s_2f_2...\vee s_rf_r$.

3. Способом, описанным в доказательстве леммы 10, для каждой f_i из $f_1,..., f_r$ находим функции $s_{0,i}, ..., s_{m,i}$ в $B^{(1)}$ такие, что $f_i(x_1,..., x_m)=s_{0,i}(s_{1,i}(x_1)\vee...\vee s_{m,i}(x_m))$, и представляем функцию f в форме $f=f_0\vee s_1f_1\vee s_2f_2...\vee s_rf_r=$ $=f_0\vee s_1[s_{0,1}(s_{1,1}(x_1)\vee...\vee s_{m,1}(x_m))]\vee s_2[s_{0,2}(s_{1,2}(x_1)\vee...\vee s_{m,2}(x_m))]$ $...\vee s_r[s_{0,r}(s_{1,r}(x_1)\vee...\vee s_{m,r}(x_m))]$.

Описанный метод синтеза является доказательством следующей теоремы.

Теорема 3. Система функций $\{\vee\}\cup T^{(1)}\cup B_M^{(1)}$ s-полна в M.

ЛИТЕРАТУРА

1. *Агибалов Г.П.* Дискретные автоматы на полурешетках. Томск: Изд-во Том. ун-та, 1993. 227 с.
2. *Агибалов Г.П.* О полных системах операций и синтезе схем для квазимонотонных функций на конечных полурешетках // Новые информационные технологии в исследовании дискретных структур. Екатеринбург: Из-во УрО РАН, 1998. С. 149–152.
3. *Агибалов Г.П.* О полных системах функций на полурешетке подмножеств конечного множества // Всесибирские чтения по математике и механике: Математика. Томск: Изд-во Том. ун-та, 1997. Т. 1. С. 148–149.
4. *Агибалов Г.П.* К синтезу схем, реализующих квазимонотонные функции на полурешетке подмножеств двухэлементного множества. // Там же. С. 147–148.

Статья представлена кафедрой защиты информации и криптографии факультета прикладной математики и кибернетики Томского государственного университета, поступила в научную редакцию 1 марта 2000 г.

УДК 621.391.7

И.В. Пронина, Г.П. Агибалов

НЕКОТОРЫЕ АЛГОРИТМЫ КРИПТАНАЛИЗА ДЛЯ КОДОВЫХ КРИПТОСИСТЕМ

Предлагаются алгоритмы криптанализа для кодовых криптосистем с закрытым ключом с целью нахождения ключа при возможности выбора сообщений и для кодовых криптосистем с открытым ключом с целью нахождения сообщения, если известны открытый ключ и криптограмма.

Введение

Кодовые криптосистемы строятся на основе линейных кодов, исправляющих ошибки. Как и все криптосистемы, они делятся на два класса – симметричные, или с закрытым ключом, и несимметричные, или с открытым ключом. Криптанализу последних посвящены работы [1, 2, 3], где для некоторых кодовых криптоси-

стем с открытым ключом решена задача нахождения закрытого ключа по соответствующему открытому ключу. В данной работе рассматриваются двоичные (на основе двоичных кодов) кодовые криптосистемы. Для таких криптосистем первого класса решается задача криптанализа с целью нахождения закрытого ключа при возможности выбора открытого текста (сообщений). Предлагается вероятностный алгоритм, доставляющий решение задачи с вероятностью, близкой к 1. Приводятся результаты испытаний алгоритма на компьютере. Для кодовых криптосистем второго класса решается задача криптанализа с целью нахождения сообщения, если известны открытый ключ и криптограмма. По существу это есть задача декодирования для произвольного кода, исправляющего ошибки, для которой в настоящее время не известны алгоритмы решения с полиномиальной сложностью. В данной работе предлагается эвристический алгоритм поиска её решения путём перебора возможных кандидатов на решение с использованием некоторых правил сокращения перебора. В отдельных случаях это сокращение может оказываться значительным и приводить к решению задачи за полиномиальное время.

Криптанализ симметричных систем

Шифрование в симметричной двоичной кодовой криптосистеме над полем GF(2) описывается следующим уравнением:

$$y=x\cdot G\oplus e, \qquad (1)$$

где x – булев вектор с k компонентами, являющийся сообщением; y – булев вектор с n компонентами, являющийся криптограммой; G – булева матрица размера $k\times n$, которая является закрытым ключом и представляет собой порождающую матрицу некоторого двоичного линейного (n, k)-кода, исправляющего $t\geq 1$ ошибок; e – случайный булев вектор с n компонентами, который содержит не более t единиц и играет роль вектора ошибок.

Задача криптанализа для нахождения закрытого ключа при возможности выбирать открытый текст заключается в данном случае в том, что требуется так выбрать значения вектора x, чтобы по этим значениям и по соответствующим значениям вектора $y=xG\oplus e$, при случайных значениях вектора e можно было бы найти матрицу G.

Предлагается следующий подход к решению этой задачи. Будем выбирать в качестве значений вектора x единичные векторы (100…0), (010…0), …,(00…01), причем каждый такой вектор может браться некоторое число s раз, $s\geq 3$. Тогда рассматриваемая здесь задача криптанализа сводится к решению k раз следующей частной задачи: пусть булевы векторы $g, y_1, y_2, \ldots, y_s, e_1, e_2, \ldots, e_s$ одной и той же длины n удовлетворяют уравнениям

$$y_i=g\oplus e_i,\ i=1, 2, \ldots, s, \qquad (2)$$

и вес каждого вектора e_i не превосходит заданного числа $t\geq 1$; требуется, зная векторы $y_1, y_2, \ldots, y_s$, найти векторы $g, e_1, e_2, \ldots, e_s$. В самом деле, если в уравнении (1) вектору x придать s раз значение одного и того же единичного вектора с 1 в j-й компоненте, то получим соотношение (2), где g есть j-я строка искомой матрицы G.

Заметим, что есть много других практических ситуаций, в которых возникает эта частная задача. Вот только два примера.

Пример 1. Здесь: g – сигнал в форме двоичного вектора, многократно (s раз) передаваемый по симметричному двоичному каналу с шумом; e_i – вектор ошибки, случающейся в нем при i-й передаче; t – максимально возможная кратность ошибки; задача приемника состоит в определении сигнала g путем исключения неизвестных ошибок $e_1, e_2, \ldots, e_s$ из принимаемых векторов $y_1, y_2, \ldots, y_s$.

Пример 2. Здесь: конкатенация $e=e_1e_2\ldots e_s$ – это некоторое сообщение, конкатенация $y=y_1y_2\ldots y_s$ – это криптограмма, полученная из e по ключу g методом Виженера; задача криптоаналитика заключается в определении по y ключа g и сообщения e.

Предлагается следующий алгоритм решения частной задачи. Векторы e_i однозначно определяются по y_i и g, поэтому достаточно найти g. Пусть $y_i=y_{i1}y_{i2}\ldots y_{in}$, $i=1,2,\ldots, s$. Для каждого $j=1,2, \ldots, n$ подсчитаем вес w_j вектора $y_{1j}y_{2j}\ldots y_{sj}$, и полагаем $g'_j=1$, если $w_j>s/2$, и $g'_j=0$, если $w_j<[s/2]$. Полученный вектор g' является результатом работы алгоритма.

Данный алгоритм следует считать вероятностным алгоритмом решения задачи, т.к. полученный вектор g' совпадает с искомым вектором g лишь с некоторой вероятностью P, зависящей от соотношения между параметрами t, n и s.

Теорема. Если $t\leq n/2$, $s=2r+1$, $t\geq 1$ и p – вероятность принятия компонентой вектора e значения 1, то вероятность $P=\Sigma_{i=0}^{r} C_s^i p^i(1-p)^{s-i}$.

Доказательство. Допустим, известны векторы $e_i=e_{i1}e_{i2}\ldots e_{in}$, $i=1, 2, \ldots, s$. Для каждого $j=1, 2, \ldots, n$ подсчитаем вес w_j вектора $e_{1j}e_{2j}\ldots e_{sj}$ и положим $e'_j=1$, если $w_j>s/2$, и $e'_j=0$, если $w_j<[s/2]$. Если $w_j<[s/2]$ для любого $j=1,2, \ldots, n$, то $e'=(00\ldots0)$ и $g'=g$. Следовательно, вероятность получения алгоритмом решения задачи равна вероятности того, что $e'=(00\ldots0)$. Найдем эту вероятность.

Рассмотрим множество Ω всех булевых векторов длины n и веса $w\leq t$. Мощность этого множества равна $C_n^0+C_n^1+\ldots+C_n^t$. Будем проводить опыты, каждый из которых состоит в извлечении из множества Ω некоторого вектора e с возвратом. Пусть A – случайное событие, которое состоит в том, что значение некоторой компоненты извлеченного вектора e равно 1. Вероятность p этого события равна

$$(C_{n-1}^0+C_{n-1}^1+\ldots+C_n^{t-1})/(C_n^0+C_n^1+\ldots+C_n^t).$$

Можно рассматривать p как функцию переменной t при фиксированном значении n. Если t изменяется от 1 до n, то p возрастает и принимает значения $1/(n+1)$, …, 1/2. Известно [4]: вероятность того, что в λ опытах событие A наступит ровно μ раз, будет $P_\lambda(\mu)=C_\lambda^\mu p^\mu(1-p)^{\lambda-\mu}$. Вероятность того, что в $s=2r+1$ опытах событие A наступит меньше либо равно r раз, будет $P=P_s(0)+P_s(1)+\ldots+P_s(r)$. Таким образом, вероятность получения алгоритмом решения задачи равна $P=\Sigma_{i=0}^{r} C_s^i p^i(1-p)^{s-i}$. Теорема доказана.

Приведем значения вероятности P в таблице, где строкам соответствует число проводимых опытов s, столбцам – максимальный вес векторов t. Длина векторов $n=19$.

	1	2	3	4	5	6	7	8	9
3	0,993	0,972	0,94	0,899	0,852	0,799	0,744	0,689	0,637
5	0,999	0,991	0,974	0,945	0,904	0,852	0,793	0,731	0,67
11	0,999	0,999	0,997	0,989	0,97	0,934	0,881	0,813	0,737
15	0,999	0,999	0,999	0,996	0,985	0,96	0,915	0,849	0,769
21	0,999	0,999	0,999	0,999	0,995	0,98	0,947	0,888	0,807
29	0,999	0,999	0,999	0,999	0,998	0,992	0,971	0,923	0,845

Алгоритм и установленные соотношения тривиальным образом переносятся на случай, когда вес каждого вектора e_i не меньше заданного t.

На компьютере проводились испытания с целью выяснить реальные значения s при различных значениях параметров n, t. Генерировалась случайная последовательность булевых векторов e_i размерности n и веса $\le t$ до тех пор, пока из них не получался вектор e'=(00...0), и фиксировалось число s сгенерированных векторов. Для каждой пары n, t эксперимент повторялся 1024 раза. Среди этого множества значений s выбиралось минимальное, максимальное и вычислялось среднее арифметическое значение. В следующей таблице строкам соответствует максимальный вес векторов t, столбцам – длина векторов n и на пересечении указана пара (s максимальное; s среднее). Из таблицы видно, что отношение s/n убывает с ростом n.

	32	64	128	256	512
$t=n/32$	6; 3	8; 3	6; 3	6; 3	6; 3
$t=n/16$	6; 3	8; 3	8; 3	8; 3	10; 3
$t=n/8$	12; 3	12; 3	12; 4	10; 4	12; 4
$t=n/4$	16; 4	18; 5	20; 5	22; 6	24; 8
$t=n/2$	50; 10	50; 12	57; 15	61; 18	70; 22

Криптанализ систем с открытым ключом

Здесь мы рассматриваем несимметричные кодовые криптосистемы, в которых в качестве открытого ключа выступает некоторая порождающая матрица G произвольного двоичного (n,k)-кода, исправляющего ошибки кратности меньшей либо равной t, а криптограмма y и сообщение x являются булевыми векторами длины n и k соответственно и связаны между собой уравнением (1). Примером такой криптосистемы может быть криптосистема Мак-Элиса [5,6], в которой секретным ключом является тройка (G', S, P), открытым ключом – матрица $G=SG'P$, где G' – порождающая матрица (n,k)-кода Гоппы с исправлением t-кратных ошибок, S – невырожденная матрица размера $(k\times k)$, P – перестановочная матрица размера $(n\times n)$.

Исследуем задачу криптанализа, состоящую в нахождении секретного сообщения по открытому ключу и криптограмме, т.е. для заданной матрицы G и заданного вектора y требуется найти вектор x, удовлетворяющий уравнению (1), при некотором (заранее неизвестном) булевом векторе e, $w(e)\le t$. Это есть задача декодирования для кода, порожденного матрицей G, и она является NP-полной. Это значит, что в настоящее время не существует алгоритма полиномиальной сложности, решающего её для произвольной матрицы G. Ниже предлагается эвристический алгоритм, который в некоторых случаях (для некоторых матриц G) решает данную задачу за полиномиальное время.

Искомый вектор x будем называть решением поставленной задачи и трактовать как набор значений булевых переменных $x_1, x_2,..., x_k$, полагая, что i-я компонента в нем равна значению переменной x_i, i=1, 2,..., k. Его префикс любой длины $i\le k$ будем называть частичным решением задачи и трактовать как набор значений первых i из этих переменных. Частичным решением считаем также и пустой вектор λ. Частичное решение $\alpha=a_1a_2...a_i$ для $i\le k$ называется квазиполным, если не превосходит величины t вес вектора $z(\alpha)=y\oplus a_1g_1\oplus ...\oplus a_ig_i$, где g_j для j=1, 2,..., k есть j-я строка матрицы G. В этом случае, очевидно, вектор $\alpha 0^{k-i}$, где 0^m – набор из m нулей, будет (полным) решением задачи, оправдывая название для α.

Будем искать решение задачи переборным методом, использующим дерево поиска. Вершины в последнем суть вектор λ и всевозможные булевы векторы длин 1, 2,..., k, распределенные по $(k+1)$ ярусам так, что корнем дерева (и вершиной 0-го яруса) служит вектор λ, а вершинами i-го яруса для i=1,..., k являются все векторы длины i. Ребра в дереве расположены между соседними ярусами, причем для каждого i=0, 1,..., k-1 вершина α i-го яруса соединена ребром с вершиной β $(i+1)$-го яруса, если и только если $\beta=\alpha\sigma$ для $\sigma\in\{0,1\}$. По определению, каждая вершина i-го яруса смежна двум вершинам $(i+1)$-го яруса и каждая вершина $(i+1)$-го яруса смежна одной вершине i-го яруса. Полным решением задачи является некоторый лист (вершина k-го яруса) дерева поиска. Все частичные решения лежат (в качестве вершин) на пути из корня в этот лист.

Понятие частичного решения не является конструктивным, и нам неизвестен тест, позволяющий для любого булева вектора α длины $i\le k$ или, что одно и то же, для любой вершины α i-го яруса эффективно проверять, является ли α частичным решением задачи. Вместе с тем можно указать (это будет сделано ниже) некоторые легко проверяемые условия, при которых α заведомо не будет частич-

ным решением. Располагая такими условиями, можно для нахождения решения задачи предложить следующий алгоритм поиска с возвращением (по дереву).

Алгоритм моделирует процесс перехода из корня дерева в искомый лист путем выполнения серии подъемов и спусков, осуществляемых при определенных условиях. Подъем – это перемещение из достигнутой вершины i-го яруса в смежную вершину $(i+1)$-го яруса для $i<k$, а спуск – возвращение из текущей вершины i-го яруса в смежную вершину $(i-1)$-го яруса для $i>0$. Подъем может выполняться только в вершину, в которую до этого еще не поднимались. В отсутствие таковой или, если достигнутая вершина не может быть частичным решением задачи, осуществляется спуск. По достижению вершины дерева, являющейся квазиполным решением задачи, алгоритм останавливается. Ниже дается строгое описание данного алгоритма. В нем через $\alpha[i]$ и $\alpha]_i$ обозначены соответственно i-я компонента и префикс длины i вектора α.

1. (Начальная установка) $\alpha:=\lambda$, $i:=0$, $z(\alpha)=y$.

2. $i=i+1$ и

3. (Подъем).

3.1. $\alpha:=\alpha 0$, и если α не может быть частичным решением, то п. 3.2, иначе – п. 2.

3.2. $\alpha:=\alpha 1$, $z(\alpha):=z(\alpha)\oplus g_i$, и если α не может быть частичным решением, то п. 4, в противном случае, если α – квазиполное, то п. 5, иначе п. 2.

4. (Спуск) $z(\alpha):=z(\alpha)\oplus g_i$, $i:=i-1$, $\alpha:=\alpha]_i$, и если $\alpha[i]=1$, то п. 4, иначе $\alpha:=\alpha]_{i-1}$ и п. 3.2.

5. Вектор $\alpha 0^{k-i}$ есть решение.

Сформулируем условия, при которых булев вектор $\alpha=a_1a_2\ldots a_i$ для $i<k$ не может быть частичным решением задачи. Вычислим вектор $z(\alpha)=z_1z_2\ldots z_n$. Пусть матрица H получена из G вычеркиванием первых i строк и t_0 обозначает количество таких r в $\{1, 2,\ldots, n\}$, что $z_r=1$ и r-й столбец матрицы H состоит из одних нулей (является нулевым). Разобьем множество всех ненулевых столбцов матрицы H на классы одинаковых столбцов. Пусть число классов равно m, и s-й из них, $s=1, 2,\ldots, m$, состоит из столбцов с номерами $r_1,\ldots,r_p$. Обозначим t_{s0} и t_{s1} количество нулей и единиц соответственно среди компонент вектора $z(\alpha)$, имеющих номера $r_1,\ldots, r_p$. Пусть, наконец, t_s есть наименьшее из t_{s0} и t_{s1}, а t' равно сумме всех t_s, $s=1,\ldots, m$.

Теорема. Вектор α не может быть частичным решением задачи, если $t_0+t'>t$.

Утверждение справедливо, так как в противном случае вектор e в уравнении (1) должен иметь вес больше t.

ЛИТЕРАТУРА

1. *Сидельников В.М.* Открытое шифрование на основе двоичных кодов Рида-Маллера // Дискретная математика. 1994. Вып. 2.
2. *Сидельников В.М., Шестаков С.О.* О системе шифрования, построенной на основе обобщенных кодов Рида-Соломона // Дискретная математика. 1992. Вып. 3.
3. *Брикелл Э.Ф., Одлижко Э.М.* Криптоанализ: Обзор новейших результатов // ТИИЭР. 1998. № 5.
4. *Радюк Л.Е., Терпугов А.Ф.* Теория вероятностей и случайных процессов. Томск: Изд-во Том. ун-та, 1988.
5. *Аснис И.Л., Федоренко С.В., Шабунов К.Б.* Краткий обзор криптосистем с открытым ключом // Защита информации. 1994. № 2.
6. *Агибалов Г.П.* Адекватные модели полурешеток, функций и автоматов на полурешетках // Вестник ТГУ. 2000. № 271.

Статья представлена кафедрой защиты информации и криптографии факультета прикладной математики и кибернетики Томского государственного университета, поступила в научную редакцию 1 марта 2000 г.

УДК 519.7

Г.П. Агибалов

АДЕКВАТНЫЕ МОДЕЛИ ПОЛУРЕШЕТОК, ФУНКЦИЙ И АВТОМАТОВ НА ПОЛУРЕШЕТКАХ

Работа выполнена при финансовой поддержке РФФИ, грант № 98-01-00288

Определяются понятия адекватной модели и ее точности для конечной верхней полурешетки, для функции и автомата на таких полурешетках и для полурешеточно упорядоченной алгебры. В этих определениях элементами адекватной модели полурешетки служат все наибольшие элементы смежных классов некоторой конгруэнции, которая, в свою очередь, характеризует точность модели. Изучаются свойства и даются методы построения адекватных моделей для полурешеток, функций и автоматов на них.

Введение

Важнейшими характеристиками всякой математической модели являются ее адекватность и степень точности. В случае дискретных моделей первая понимается как безошибочность в том смысле, что результат адекватного моделирования всегда содержит в себе истинное значение моделируемой величины, а вторая – как степень неопределенности этого результата [1]. Для управляющих систем на конечных верхних полурешетках удается формализовать эти понятия, а утверждения о них сделать доказательными [2]. Это достигается определением адекватной модели на полурешетках, составленных из наибольших элементов смежных классов полурешеток заданной системы по некоторым конгруэнциям, которыми и характеризуется степень точности модели.

В данной работе понятия адекватной модели и ее точности определяются для полурешеток, функций и автоматов на полурешетках и для полурешеточно упорядоченных алгебр. Указан метод построения адекватных моделей для полурешеток подмножеств и интервалов с точностью до эквивалентности на множестве минимальных элементов полурешетки. Показано, что свойства монотонности и квазимонотонности функции на полурешетках передаются любой ее адекватной модели, а свойство аддитивности – лишь модели, точность которой сохраняется данной функцией. Установлено, что адекватная модель суперпозиции монотонных функций реализует суперпозицию их адекватных моделей и совпадает с ней в случае сохранения функциями точностей их моделей. В терминах стабильных троек конгруэнций охарактеризованы всевозможные адекватные модели произвольного конечного автомата на полурешетках. Показано, что свойства аддитивности, монотонности и квазимонотонности автомата на полурешетках сохраняются в его адекватных моделях.