

О КРИПТОАНАЛИТИЧЕСКОЙ ОБРАТИМОСТИ ДИСКРЕТНЫХ ФУНКЦИЙ

И. А. Панкратова, А. Д. Сорокоумова

Томский государственный университет, г. Томск, Россия

E-mail: pank@mail.tsu.ru, a.srkmva@mail.ru

Рассматривается понятие криптоаналитической обратимости функции по переменной, его связь с другими понятиями. Доказаны критерии криптоаналитической обратимости для функций от двух и трёх аргументов. Сформулированы алгоритмы построения функции восстановления и генерации обратимых функций.

Ключевые слова: *обратимость функции по переменной, криптоаналитическая обратимость, критерий обратимости, функция восстановления.*

ON CRYPTANALYTIC INVERTIBILITY OF DISCRETE FUNCTIONS

I. A. Pankratova, A. D. Sorokoumova

Tomsk State University, Tomsk, Russia

The function $g(x_1, \dots, x_n)$ is invertible with respect to the variable x_k of type $K_1 \dots K_n$, where $k \in \{1, \dots, n\}$, $K_i \in \{\exists, \forall\}$ and $K_k = \forall$, if there exists a recovery function f such that the invertibility condition is true:

$$K_1 x_1 \dots K_n x_n (f(g(x_1, \dots, x_n)) = x_k).$$

Criteria for cryptanalytic invertibility of functions $g : D_1 \times D_2 \times D_3 \rightarrow D$ are proven: 1) a function g is invertible with respect to the variable x_1 of the type $\forall\forall\exists$ iff there is a mapping $\varphi : D_1 \times D_2 \rightarrow D_3$ such that the following condition is satisfied:

$$\forall a, c \in D_1 \forall b, d \in D_2 (a \neq c \Rightarrow g(a, b, \varphi(a, b)) \neq g(c, d, \varphi(c, d)));$$

2) a function g is invertible with respect to the variable x_1 of the type $\forall\exists\forall$ iff there is a mapping $\varphi : D_1 \rightarrow D_2$ such that the following condition is satisfied:

$$\forall a, c \in D_1 \forall b, d \in D_3 (a \neq c \Rightarrow g(a, \varphi(a), b) \neq g(c, \varphi(c), d));$$

3) a function g is invertible with respect to the variable x_3 of the type $\forall\exists\forall$ iff there is a mapping $\varphi : D_1 \rightarrow D_2$ such that the following condition is satisfied:

$$\forall a, c \in D_1 \forall b, d \in D_3 (b \neq d \Rightarrow g(a, \varphi(a), b) \neq g(c, \varphi(c), d));$$

4) a function g is invertible with respect to the variable x_2 of the type $\exists\forall\forall$ iff there is $a \in D_1$ such that the following condition is satisfied ($G^{(a,b)} = \{g(a, b, x_3) : x_3 \in D_3\}$):

$$\forall b, d \in D_2 (b \neq d \Rightarrow G^{(a,b)} \cap G^{(a,d)} = \emptyset);$$

5) a function g is invertible with respect to the variable x_2 of the type $\exists\forall\exists$ iff there are $a \in D_1$ and a mapping $\varphi : D_2 \rightarrow D_3$ such that the following condition is satisfied:

$$\forall b, d \in D_2 (b \neq d \Rightarrow g(a, b, \varphi(b)) \neq g(a, d, \varphi(d))).$$

Algorithms for constructing a recovery function and generating invertible functions are formulated too.

Keywords: *cryptanalytic invertibility, invertibility criterion, recovery function.*

Введение

Понятие криптоаналитической обратимости функции введено Г.П. Агибаловым в [1, 2] как обобщение понятия «обычной» обратимости функции.

Определение 1 [2]. Функция $g(x_1, \dots, x_n)$ обратима по переменной x_k типа $K_1 \dots K_n$, где $k \in \{1, \dots, n\}$, $K_i \in \{\exists, \forall\}$ и $K_k = \forall$, если существует функция восстановления f , такая, что верна формула (условие обратимости)

$$K_1 x_1 \dots K_n x_n (f(g(x_1, \dots, x_n)) = x_k).$$

Рассмотрим случай, когда области определения D_i переменных x_i для всех $i = 1, \dots, n$ конечны. Тогда можно дать следующее эквивалентное определение:

Определение 2. Функция $g(x_1, \dots, x_n)$ обратима по переменной x_k типа $K_1 \dots K_n$, $k \in \{1, \dots, n\}$, $K_i \in \{\exists, \forall\}$ и $K_k = \forall$, если выполнима формула

$$O_1 \dots O_n (f(g(x_1, \dots, x_n)) = x_k), \quad (1)$$

где $O_i = \bigvee_{x_i \in D_i}$, если $K_i = \exists$; $O_i = \bigwedge_{x_i \in D_i}$, если $K_i = \forall$, $i = 1, \dots, n$. Любая функция f , обращающая (1) в истину, является функцией восстановления.

Равносильность определений 1 и 2 следует из выражений кванторов общности и существования на конечной области определения $D = \{d_1, \dots, d_r\}$ через логические операции:

$$\begin{aligned} \forall x P(x) &= P(d_1) \wedge P(d_2) \wedge \dots \wedge P(d_r), \\ \exists x P(x) &= P(d_1) \vee P(d_2) \vee \dots \vee P(d_r). \end{aligned}$$

Здесь $P(x)$ — произвольный предикат.

Пример 1. Пусть $n = 2$ и булева функция g в общем виде задана в табл. 1, где $a, b, c, d \in \{0, 1\}$.

Т а б л и ц а 1

x_1	x_2	g
0	0	a
0	1	b
1	0	c
1	1	d

Тогда формула (1) примет следующий вид:

— для обратимости типа $\forall\forall$ по переменной x_1

$$(f(a) = 0 \wedge f(b) = 0) \wedge (f(c) = 1 \wedge f(d) = 1); \quad (2)$$

— для обратимости типа $\forall\exists$ по переменной x_1

$$(f(a) = 0 \vee f(b) = 0) \wedge (f(c) = 1 \vee f(d) = 1); \quad (3)$$

— для обратимости типа $\exists\forall$ по переменной x_2

$$(f(a) = 0 \wedge f(b) = 1) \vee (f(c) = 0 \wedge f(d) = 1). \quad (4)$$

Сравним с условиями критериев обратимости функций от двух переменных [3] (см. далее утверждение 1):

- для обратимости типа $\forall\forall$ по переменной x_1 должно быть $\{a, b\} \cap \{c, d\} = \emptyset$; очевидно, что в этом и только в этом случае выполнено условие функциональности для отображения f , удовлетворяющего (2);
- для обратимости типа $\forall\exists$ по переменной x_1 должно существовать такое отображение $\varphi : \{0, 1\} \rightarrow \{0, 1\}$, что $g(0, \varphi(0)) \neq g(1, \varphi(1))$. Функцию f , удовлетворяющую (3), можно построить, если и только если $\{a, b, c, d\} = \{0, 1\}$, т.е. если $g \neq \text{const}$. Пусть $g(y_1, y_2) \neq g(z_1, z_2)$. Тогда если $y_1 \neq z_1$, то нужное отображение задаётся условиями $\varphi(y_1) = y_2$ и $\varphi(z_1) = z_2$. Если $y_1 = z_1$, то $g(\bar{y}_1, 0) \neq g(y_1, y_2)$ или $g(\bar{y}_1, 0) \neq g(y_1, z_2)$. В первом случае положим $\varphi(y_1) = y_2$, $\varphi(\bar{y}_1) = 0$; во втором — $\varphi(y_1) = z_2$, $\varphi(\bar{y}_1) = 0$;
- для обратимости типа $\exists\forall$ по переменной x_2 необходимо и достаточно, чтобы $|\{a, b\}| = 2$ или $|\{c, d\}| = 2$; в первом случае положим $f(a) = 0$ и $f(b) = 1$, во втором — $f(c) = 0$ и $f(d) = 1$, тем самым условие (4) будет удовлетворено.

Введённое понятие криптоаналитической обратимости функции связано и с другими известными ранее понятиями, проясним эту связь.

Определение 3 [4, с. 31]. Функция $g(x_1, \dots, x_n) : D_1 \times \dots \times D_n \rightarrow D$ инъективна по переменной x_i , $i \in \{1, \dots, n\}$, если при любых значениях $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n$ переменных $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ отображение $g(a_1, \dots, a_{i-1}, x_i, a_{i+1}, \dots, a_n) : D_i \rightarrow D$ инъективно:

$$\begin{aligned} & \forall a_1 \in D_1 \dots \forall a_{i-1} \in D_{i-1} \forall b, c \in D_i \forall a_{i+1} \in D_{i+1} \dots \forall a_n \in D_n \\ & (b \neq c \Rightarrow g(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n) \neq g(a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_n)). \end{aligned} \quad (5)$$

Условие (5) равносильно тому, что существует функция восстановления f значения переменной x_i , но не только по значению функции g , как в определении 1, а ещё и по значениям остальных аргументов:

$$\exists f \forall x_1 \dots \forall x_n (f(g(x_1, \dots, x_n), x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = x_i).$$

Другими словами, инъективность функции $g(x_1, \dots, x_n)$ по переменной x_i равносильна обратимости типа $\forall\forall \dots \forall$ по той же переменной векторной функции $(g, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$. Если распространить на функции терминологию [5], применённую там к понятию обратимости конечных автоматов, то можно в этом случае говорить об обратимости функции g степени $\forall\forall \dots \forall$ порядка $X \setminus \{x_i\}$, где $X = \{x_1, \dots, x_n\}$ — множество переменных функции g .

Ещё одно понятие — сильно зависимых функций — применимо в случае, если $D_1 = \dots = D_n = D$.

Определение 4 [6]. Функция $g : D^n \rightarrow D$ называется сильно зависимой, если для всех $i = 1, \dots, n$ найдётся фиксация всех переменных, кроме x_i , при которой полученная после фиксации функция становится подстановкой.

Или, что равносильно: функция g сильно зависима, если она обратима по всем переменным x_i , $i = 1, \dots, n$, степени $\underbrace{\exists \dots \exists}_{i-1} \forall \underbrace{\exists \dots \exists}_{n-i}$ порядка $X \setminus \{x_i\}$.

Таким образом, определение 1 в каком-то смысле обобщает понятия инъективной по переменной и сильно зависимой функции, так как допускает большее разнообразие кванторных приставок.

В [3] рассмотрен случай $n = 2$ и для всех типов обратимости решены следующие задачи:

- 1) разработка критерия обратимости;
- 2) разработка алгоритма построения функции восстановления;
- 3) разработка алгоритма генерации обратимых функций.

В [7] решены те же задачи для функций от трёх переменных для тех типов обратимости, которые не сводятся к случаю $n = 2$. Однако не все критерии обратимости конструктивны, некоторые из них требуют существования отображения с определёнными свойствами. Для решения этой проблемы требуется сформулировать алгоритмы построения нужных отображений, а можно пойти другим путём: проверять выполнимость формулы (1), попутно строя функцию восстановления f . В данной работе приведены конструктивные критерии обратимости всех типов для функций от двух и трёх переменных.

1. Критерии обратимости функций от двух переменных

Рассматриваются функции вида $g : D_1 \times D_2 \rightarrow D$. Необходимым условием обратимости функции $g(x_1, x_2)$ по переменной x_k , $k \in \{1, 2\}$, является $|D| \geq |D_k|$; будем всюду считать, что оно выполнено. Для любого $a \in D_1$ обозначим $G^{(a)} = \{g(a, x_2) : x_2 \in D_2\}$.

Утверждение 1 [3]. Имеют место следующие критерии обратимости:

- 1) Функция $g(x_1, x_2)$ обратима типа $\forall\forall$ по переменной x_1 , если и только если

$$\forall a, b \in D_1 (a \neq b \Rightarrow G^{(a)} \cap G^{(b)} = \emptyset). \quad (6)$$

- 2) Функция $g(x_1, x_2)$ обратима типа $\exists\forall$ по переменной x_2 , если и только если существует такое $a \in D_1$, что

$$|G^{(a)}| = |D_2|. \quad (7)$$

- 3) Функция $g(x_1, x_2)$ обратима типа $\forall\exists$ по переменной x_1 , если и только если существует такое отображение $\varphi : D_1 \rightarrow D_2$, что выполнено условие

$$\forall a, b \in D_1 (a \neq b \Rightarrow g(a, \varphi(a)) \neq g(b, \varphi(b))). \quad (8)$$

В утверждении 1 условия (6) и (7) легко проверяются, а вот критерий обратимости типа $\forall\exists$ требует проверки существования отображения φ со свойством (8). Функция восстановления $f : D \rightarrow D_1$ должна удовлетворять условию

$$\forall x_1 \exists x_2 (f(g(x_1, x_2)) = x_1). \quad (9)$$

Если отображение φ удалось найти, то функция f строится так:

- 1) для всех $a \in D_1$ положить $f(g(a, \varphi(a))) = a$;
- 2) для каждого $y \in D$, такого, что значение $f(y)$ не определено на шаге 1, выбрать в качестве $f(y)$ произвольное значение из D_1 .

Функциональность отношения f следует из того, что, в силу условия (8), все значения $g(a, \varphi(a))$ для $a \in D_1$ попарно различны.

Сформулируем алгоритм 1 построения отображения φ . Пусть $D_1 = \{a_1, \dots, a_n\}$, $D_2 = \{b_1, \dots, b_m\}$, множество значений функции g равно

$$\{g(x_1, x_2) : x_1 \in D_1, x_2 \in D_2\} = \{c_1, \dots, c_k\} \subseteq D;$$

заметим, что это множество конечно, даже если D бесконечно: $k \leq nt$.

Будем строить отображение φ с помощью построения дерева и обхода его в глубину. Вершинам i -го яруса, $i = 1, \dots, n$, сопоставим значения a_i ; дугам, исходящим из вершин, — значения b_j , $j = 1, \dots, m$. Листья дерева (вершины $(n+1)$ -го яруса) обозначим « $\square$ ». Цель — найти путь $a_1 \xrightarrow{b_{j_1}} a_2 \xrightarrow{b_{j_2}} \dots \rightarrow a_n \xrightarrow{b_{j_n}} \square$, такой, что все значения $g(a_i, b_{j_i})$, $i = 1, \dots, n$, попарно различны; тогда искомое отображение строится по правилу $\varphi(a_i) = b_{j_i}$, $i = 1, \dots, n$. Уникальность значений $g(a_i, b_{j_i})$ будем проверять с помощью характеристического вектора $v \in \{0, 1\}^k$; используем также вспомогательный массив B , полагая $B[i] = j$, если $\varphi(a_i) = b_j$ в текущем варианте строящегося отображения φ .

Алгоритм 1. Построение отображения φ для критерия обратимости типа $\forall\exists$

Вход: функция $g : D_1 \times D_2 \rightarrow D$.

Выход: отображение $\varphi : D_1 \rightarrow D_2$, такое, что выполнено условие (8).

- 1: $v = (v_1 \dots v_k) := 0^k$; $i := 1$.
 - 2: $j := 1$.
 - 3: $B[i] := j$; найти s , такое, что $g(a_i, b_j) = c_s$.
 - 4: **Если** $v_s = 0$, **то**
 $v_s := 1$.
 - 5: **Если** $i = n$, **то**
 перейти к п. 19,
 - 6: **иначе**
 - 7: // спуск по дереву
 - 8: $i := i + 1$; переход к п. 2.
 - 9: // $v_s = 1 \Rightarrow$ надо перестроить отображение
 - 10: **Если** $j < m$, **то**
 - 11: $j := j + 1$; переход к п. 3. // строим дерево в ширину
 - 12: // иначе — возврат по дереву
 - 13: **Если** $i = 1$, **то**
 выход, ответ: «Отображения φ со свойством (8) не существует».
 - 14: **Если** $i > 1$, **то**
 - 15: $i := i - 1$; найти s , такое, что $g(a_i, b_{B[i]}) = c_s$; $v_s := 0$; $j := B[i] + 1$.
 - 16: **Если** $j \leq m$, **то**
 перейти к п. 3,
 - 17: **иначе**
 - 18: перейти к п. 13.
 - 19: Положить $\varphi(a_i) = b_{B[i]}$ для $i = 1, \dots, n$; выход.
-

Пример 2. Пусть $D_1 = D = \{1, 2, 3\}$, $D_2 = \{1, 2\}$, функция g задана табл. 2. Тогда алгоритм 1 построит отображение φ (табл. 3); в свою очередь, с его помощью получим функцию восстановления f (табл. 4). Нетрудно убедиться, что для функции f выполнено условие (9):

$$f(g(1, 1)) = f(1) = 1; \quad f(g(2, 2)) = f(3) = 2; \quad f(g(3, 2)) = f(2) = 3.$$

Т а б л и ц а 2

x_1	x_2	$g(x_1, x_2)$
1	1	1
1	2	1
2	1	2
2	2	3
3	1	1
3	2	2

Т а б л и ц а 3

$a \in D_1$	$\varphi(a)$
1	1
2	2
3	2

Т а б л и ц а 4

$a \in D$	$f(a)$
1	1
2	3
3	2

Рассмотрим теперь второй способ — по сути альтернативный критерий обратимости, состоящий в построении функции восстановления f «напрямую» — без поиска отображения φ . Условие (9) эквивалентно выполнимости формулы

$$\bigwedge_{x_1 \in D_1} \bigvee_{x_2 \in D_2} (f(g(x_1, x_2)) = x_1). \quad (10)$$

Предлагается алгоритм 2 нахождения функции восстановления f , удовлетворяющей (10).

Алгоритм 2. Построение функции восстановления f для обратимости типа $\forall\exists$

Вход: функция $g : D_1 \times D_2 \rightarrow D$, $D_1 = \{a_1, \dots, a_n\}$.

Выход: множество F , задающее (возможно, частично) функцию восстановления, удовлетворяющую (10), или ответ «Функция g не обратима типа $\forall\exists$ по переменной x_1 ».

- 1: Построить корень дерева с меткой $F = \emptyset$, объявить его текущим узлом; $i := 1$.
 - 2: Построить потомков текущего узла (узла с меткой F) на i -м ярусе по правилу:
 - 3: **Для всех** $x \in G^{(a_i)}$:
 - 4: **Если** $x \notin F_1$, **то**
 добавить потомка с меткой $F \cup \{(x, a_i)\}$.
 - 5: **Если** добавлен хотя бы один потомок, **то**
 - 6: объявить первого из них текущим узлом; $i := i + 1$.
 - 7: **Если** $i = n + 1$, **то**
 - 8: // все ярусы пройдены
 выход, ответ — F (метка текущего узла),
 - 9: **иначе**
 - 10: перейти к п. 2;
 - 11: **иначе**
 - 12: $i := i - 1$; объявить текущим предка текущего узла.
 - 13: **Если** у него есть нерассмотренные потомки, **то**
 - 14: объявить очередного из них текущим узлом, $i := i + 1$; перейти к п. 2;
 - 15: **иначе**
 - 16: **Если** $i > 0$, **то**
 перейти к п. 12;
 - 17: **иначе**
 - 18: // вернулись в корень и рассмотрели всех его потомков
 выход, ответ «Функция g не обратима типа $\forall\exists$ по переменной x_1 ».
-

Функция f строится с помощью обхода дерева в глубину. Узлам сопоставляются множества пар $F = \{(x, y)\} \subseteq D \times D_1$, где $f(x) = y$ для искомой функции f . Корень

дерева — узел нулевого яруса с меткой $\emptyset$; остальные ярусы соответствуют элементам $a \in D_1$. Обозначим: F_1 — проекция множества F по первой координате: $F_1 = \{x : \exists y((x, y) \in F)\}$. Цель — найти путь длины $|D_1|$, проходящий через все ярусы, такой, что для метки листа выполнено условие функциональности: $|F_1| = |F|$; или, что то же самое, — первые элементы в парах из F попарно различны. Функция восстановления строится тогда следующим образом:

- 1) полагаем $f(x) = y$ для всех $(x, y) \in F$;
- 2) для $x \notin F_1$ выбираем в качестве $f(x)$ любые значения из D_1 .

Пример 3. Дерево, построенное алгоритмом 2 для функции g из примера 2, представлено на рис. 1. По метке листа на третьем ярусе строится та же функция f (табл. 4).

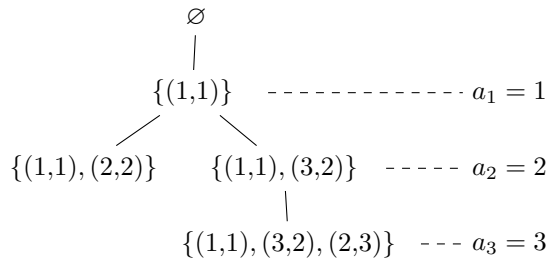


Рис. 1. Иллюстрация алгоритма 2

2. Критерии обратимости функций от трёх переменных

Рассматриваются функции вида

$$g : D_1 \times D_2 \times D_3 \rightarrow D. \quad (11)$$

Все типы обратимости функций от трёх переменных перечислены в табл. 5. Первый, четвёртый и седьмой типы эквивалентны обратимости функций от двух переменных, так как, например, функцию g вида (11) можно рассматривать как функцию $g : D_1 \times (D_2 \times D_3) \rightarrow D$ (для первого и четвёртого типов) или $g : (D_1 \times D_2) \times D_3 \rightarrow D$ (для седьмого типа). Заметим, что типы $\forall\forall\exists$ и $\exists\forall\forall$, несмотря на стоящие рядом кванторы всеобщности, не сводятся к обратимости функций от двух переменных, так как восстанавливается значение только одной переменной.

Т а б л и ц а 5

№ п/п	Тип обратимости	По переменным	Экв. тип для $n = 2$
1	$\forall\forall\forall$	x_1	$\forall\forall$
2	$\forall\forall\exists$	x_1	—
3	$\forall\exists\forall$	x_1, x_3	—
4	$\forall\exists\exists$	x_1	$\forall\exists$
5	$\exists\forall\forall$	x_2	—
6	$\exists\forall\exists$	x_2	—
7	$\exists\exists\forall$	x_3	$\exists\forall$

Кроме того, ввиду коммутативности одноимённых кванторов нет смысла отдельно рассматривать обратимость по разным переменным для второго и пятого типов. Таким образом, для функций от трёх переменных интерес представляют типы обратимости $\forall\forall\exists$ (по переменной x_1); $\forall\exists\forall$ (по переменным x_1 и x_3); $\exists\forall\forall$ и $\exists\forall\exists$ (по переменной x_2).

В работе [7] сформулированы без доказательств критерии обратимости и способы построения функций восстановления для всех пяти случаев, а также алгоритм генерации функции, обратимой типа $\forall\forall\exists$, также без доказательства его полноты и корректности. В данной работе приведены все утверждения и алгоритмы с полными доказательствами и примерами.

2.1. Обратимость типа $\forall\forall\exists$ по переменной x_1

Функция $g(x_1, x_2, x_3)$ является обратимой типа $\forall\forall\exists$ по переменной x_1 , если существует функция $f : D \rightarrow D_1$, такая, что выполняется условие

$$\forall x_1 \forall x_2 \exists x_3 (f(g(x_1, x_2, x_3)) = x_1). \quad (12)$$

Условие (12) эквивалентно выполнимости формулы

$$\bigwedge_{x_1 \in D_1} \bigwedge_{x_2 \in D_2} \bigvee_{x_3 \in D_3} (f(g(x_1, x_2, x_3)) = x_1). \quad (13)$$

Утверждение 2 (критерий обратимости типа $\forall\forall\exists$ [7]). Функция $g : D_1 \times D_2 \times D_3 \rightarrow D$ обратима типа $\forall\forall\exists$ по переменной x_1 , если и только если существует такое отображение $\varphi : D_1 \times D_2 \rightarrow D_3$, что выполнено следующее условие:

$$\forall a, c \in D_1 \forall b, d \in D_2 (a \neq c \Rightarrow g(a, b, \varphi(a, b)) \neq g(c, d, \varphi(c, d))). \quad (14)$$

Доказательство.

Д о с т а т о ч н о с т ь. Функция восстановления $f : D \rightarrow D_1$, удовлетворяющая (13), строится следующим образом:

- 1) $f(g(a, b, \varphi(a, b))) = a$ для всех $a \in D_1, b \in D_2$;
- 2) $f(x)$ равна любому значению из D_1 для тех $x \in D$, значения на которых не определены на шаге 1.

Функциональность отношения f следует из условия (14).

Н е о б х о д и м о с т ь. Пусть для функций g и f выполнено условие (12). Для всех $x_1 \in D_1, x_2 \in D_2$ положим $\varphi(x_1, x_2) = x_3$, такому, что $f(g(x_1, x_2, x_3)) = x_1$. Тогда

$$\forall a, c \in D_1 \forall b, d \in D_2 (a \neq c \Rightarrow f(g(a, b, \varphi(a, b))) \neq f(g(c, d, \varphi(c, d)))),$$

откуда ввиду функциональности f получаем (14).

Утверждение 2 доказано. ■

Как видно, критерий неконструктивен. Сформулируем алгоритм 3, который находит функцию восстановления f , удовлетворяющую (13), без построения отображения φ .

Функция f строится с помощью обхода дерева в глубину. Вершинам (узлам) сопоставляются множества пар $F = \{(x, y)\} \subseteq D \times D_1$, где $f(x) = y$ для искомой функции f ; ярусы дерева соответствуют парам $(a, b) \in D_1 \times D_2$. Обозначим: $G^{(a,b)} = \{g(a, b, x_3) : x_3 \in D_3\}$; F_1 — проекция множества F по первой координате: $F_1 = \{x : \exists y (x, y) \in F\}$. Цель — найти путь длины $|D_1| \cdot |D_2|$, проходящий через все ярусы, такой, что для метки листа выполнено условие функциональности: $|F_1| = |F|$; или, что то же самое, — первые элементы в парах из F попарно различны. Функция восстановления строится тогда следующим образом:

- 1) полагаем $f(x) = y$ для всех $(x, y) \in F$;
- 2) для $x \notin F_1$ выбираем в качестве $f(x)$ любые значения из D_1 .

Алгоритм 3. Построение функции восстановления f для обратимости типа $\forall\forall\exists$ **Вход:** функция $g : D_1 \times D_2 \times D_3 \rightarrow D$.**Выход:** множество F , задающее (возможно, частично) функцию восстановления, удовлетворяющую (13), или ответ «Функция g не обратима типа $\forall\forall\exists$ по переменной x_1 ».

- 1: Стартуем с корня дерева, его метка — $F = \emptyset$.
- 2: Строим потомков текущего узла на следующем ярусе (a, b) по правилу:
- 3: **Для всех** $x \in G^{(a,b)}$:
- 4: **Если** $x \notin F_1$, **то**
 добавляем потомка с меткой $F \cup \{(x, a)\}$.
- 5: **Если** $x \in F_1$, **то**
- 6: **если** $(x, a) \in F$, **то** удаляем всех потомков текущего узла, добавляем потомка с меткой F , объявляем его текущим узлом, переходим к шагу 2;
 иначе не добавляем потомка.
- 7: **Если** добавлен хотя бы один потомок, **то**
- 8: объявляем первого из них текущим узлом.
- 9: **Если** все ярусы пройдены, **то**
- 10: **выход**, ответ — F (метка текущего узла),
- 11: **иначе** переходим к шагу 2;
- 12: **иначе** возвращаемся по пути к корню до ближайшей точки ветвления (узла, имеющего нерассмотренных потомков), объявляем очередного потомка текущим узлом, переходим к шагу 2.
- 13: **Если** вернулись в корень и рассмотрели всех его потомков, **то**
- 14: **выход**, ответ «Функция g не обратима типа $\forall\forall\exists$ по переменной x_1 ».

Комментарий к шагу 6 алгоритма 3: если узел имеет потомков с метками F' и F , такими, что $F \subset F'$, то первый потомок «избыточен»: его можно удалить из дерева без потери решения, так как F' задаёт более жёсткие требования к функции f , чем F , и эти множества требований связаны операцией дизъюнкции. Это соответствует применению закона поглощения $AB \vee A = A$ в формуле (13).

Пример 4. Пусть $D_1 = \{1, 2, 3\}$, $D_2 = \{4, 5\}$, $D_3 = \{0, 1\}$, $D = \{0, 1, 4, 5\}$ и функция g задана в табл. 6. На рис. 2 представлено дерево, построенное алгоритмом 3; в табл. 7 — получившаяся функция восстановления.

Т а б л и ц а 6

x_1	x_2	x_3	g
1	4	0	0
1	4	1	1
1	5	0	4
1	5	1	0
2	4	0	1
2	4	1	4
2	5	0	0
2	5	1	4
3	4	0	1
3	4	1	0
3	5	0	5
3	5	1	0

Т а б л и ц а 7

x	$f(x)$
0	1
1	3
4	2
5	3

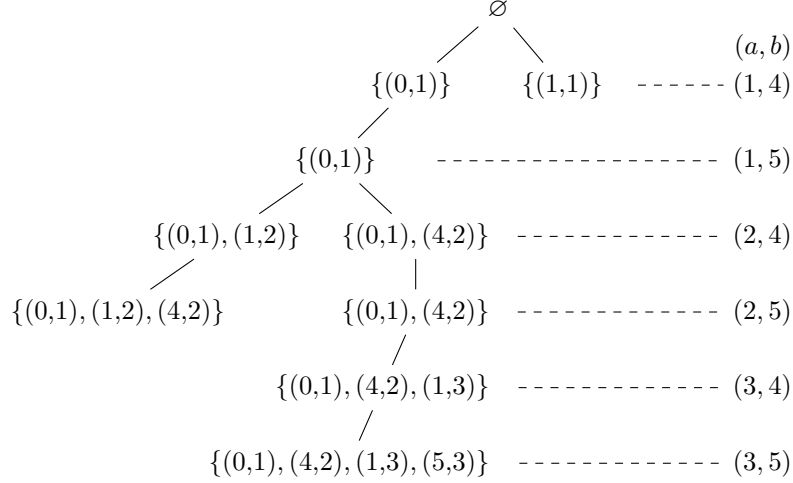


Рис. 2. Иллюстрация алгоритма 3

Алгоритм 4 задаёт способ генерации функции $g : D_1 \times D_2 \times D_3 \rightarrow D$, обратной типа $\forall\forall\exists$ по переменной x_1 .

Алгоритм 4. Генерация функции, обратной типа $\forall\forall\exists$ по переменной x_1 [7]

- 1: Построить произвольное разбиение множества D на классы H_a , $a \in D_1$.
 - 2: **Для всех** $a \in D_1$:
 - 3: **Для всех** $b \in D_2$:
 - 4: выбрать $y \in D_3$, $z \in H_a$;
 - 5: положить $g(a, b, y) := z$.
 - 6: **Для всех** $x_3 \in D_3 \setminus \{y\}$:
 - 7: выбрать в качестве $g(a, b, x_3)$ произвольное значение $s \in D$.
-

Корректность алгоритма 4: построим отображение φ , где $\varphi(a, b)$ равно значению z , выбранному для этих a, b на шаге 4. Тогда для этого φ и построенной функции g выполнено условие (14), поскольку для разных $a, c \in D_1$ блоки разбиения H_a и H_c не пересекаются, а значит, $g(a, b, \varphi(a, b)) \neq g(c, d, \varphi(c, d))$ для любых $b, d \in D_2$.

Полнота алгоритма 4: пусть для некоторой функции g и отображения φ выполнено условие (14). Обозначим $G_\varphi^{(a)} = \{g(a, b, \varphi(a, b)) : b \in D_2\}$; из (14) следует, что множества $G_\varphi^{(a)}$ для $a \in D_1$ попарно не пересекаются. Поэтому в шаге 1 алгоритма 4 разбиение множества D может быть выбрано так, что $G_\varphi^{(a)} \subseteq H_a$ для всех $a \in D_1$. Тогда именно функция g будет построена алгоритмом 4 при выборе значений $y = \varphi(a, b)$ и $z = g(a, b, y) \in G_\varphi^{(a)}$ в шаге 4 и значений $g(a, b, x_3)$ в качестве соответствующих «произвольных» в шаге 7.

Полнота и корректность алгоритмов генерации обратимых функций для остальных типов обратимости (алгоритмы 6, 8–10) доказываются аналогично.

2.2. Обратимость типа $\forall\exists\forall$ по переменной x_1

Функция $g(x_1, x_2, x_3)$ обратима типа $\forall\exists\forall$ по переменной x_1 , если существует функция $f : D \rightarrow D_1$, такая, что

$$\forall x_1 \exists x_2 \forall x_3 (f(g(x_1, x_2, x_3)) = x_1). \quad (15)$$

Утверждение 3 (критерий обратимости типа $\forall\exists\forall$ по x_1 [7]). Функция $g : D_1 \times D_2 \times D_3 \rightarrow D$ обратима типа $\forall\exists\forall$ по переменной x_1 , если и только если существует такое отображение $\varphi : D_1 \rightarrow D_2$, что выполнено условие

$$\forall a, c \in D_1 \forall b, d \in D_3 (a \neq c \Rightarrow g(a, \varphi(a), b) \neq g(c, \varphi(c), d)). \quad (16)$$

Доказательство.

Достаточность. Функция восстановления $f : D \rightarrow D_1$, удовлетворяющая (15), строится следующим образом:

- 1) $f(g(a, \varphi(a), b)) = a$ для всех $a \in D_1, b \in D_3$;
- 2) $f(x)$ равна любому значению из D_1 для тех $x \in D$, значения на которых не определены на шаге 1.

Необходимость. Пусть для функций g и f выполнено условие (15). Для всех $x_1 \in D_1, x_2 \in D_2$ положим $\varphi(x_1) = x_2$, такому, что

$$\forall x_3 (f(g(x_1, x_2, x_3)) = x_1).$$

Тогда

$$\forall a, c \in D_1 \forall b, d \in D_3 (a \neq c \Rightarrow f(g(a, \varphi(a), b)) \neq f(g(c, \varphi(c), d))),$$

откуда ввиду функциональности f получаем (16).

Утверждение 3 доказано. ■

В алгоритме 5 описан способ построения функции восстановления f , удовлетворяющей (15). Функция f строится с помощью обхода дерева в глубину. Вершинам сопоставляются подмножества $E \subseteq D$ тех значений x , для которых $f(x)$ к данному моменту определена; ярусы дерева соответствуют элементам из $D_1 = \{a_1, \dots, a_n\}$, дуги — элементам из $D_2 = \{b_1, \dots, b_m\}$; в массиве B будем запоминать номера j меток b_j дуг текущего пути.

По построению (шаги 15, 16) имеем $f(g(a_i, b_j, x_3)) = a_i$ для всех $i = 1, \dots, n$, некоторого $j \in \{1, \dots, m\}$ и всех $x_3 \in D_3$, т. е. выполняется (15). Функциональность отношения f обеспечивается проверкой в шаге 3, в результате которой множества $G^{(a_i, b_j)}$, соответствующие дугам любого пути от корня к листу, попарно не пересекаются.

Заметим, что с помощью алгоритма 5 можно также найти отображение φ , удовлетворяющее (16): для этого на шаге 16 нужно положить $\varphi(a_i) = b_{B[i]}$.

Функция g из примера 4 не является обратимой типа $\forall\exists\forall$ по переменной x_1 , что видно из рис. 3: дерево удаётся достроить только до первого яруса.

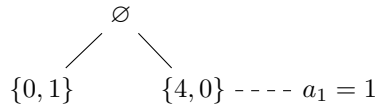


Рис. 3. Иллюстрация алгоритма 5 для функции g из примера 4

Пример 5. Пусть функция $g : \{1, 2, 3\} \times \{4, 5\} \times \{0, 1\} \rightarrow \{0, 1, 4, 5\}$ задана в табл. 8. На рис. 4 представлено дерево, построенное алгоритмом 5; в табл. 9 и 10 — функция f и отображение φ , удовлетворяющие условиям (15) и (16) соответственно.

Алгоритм 5. Построение функции восстановления для обратимости типа $\forall\exists\forall$ по переменной x_1

Вход: функция $g : D_1 \times D_2 \times D_3 \rightarrow D$.

Выход: функция f , удовлетворяющая (15), или ответ «Функция g не обратима типа $\forall\exists\forall$ по переменной x_1 ».

- 1: Построить корень дерева с меткой $E = \emptyset$, объявить его текущим узлом; $i := 1$.
 - 2: $j := 1$.
 - 3: **Если** $E \cap G^{(a_i, b_j)} = \emptyset$, **то**
 добавить потомка с меткой $E \cup G^{(a_i, b_j)}$, объявить его текущим узлом; $B[i] := j$,
 $i := i + 1$.
 - 4: **Если** $i = n + 1$, **то**
 - 5: перейти к п. 15, // все ярусы пройдены
 - 6: **иначе**
 - 7: перейти к п. 2; // обход в глубину
 - 8: **иначе**
 - 9: **Если** $j < m$, **то**
 $j := j + 1$, перейти к п. 3; // построение дерева в ширину
 - 10: **иначе**
 - 11: объявить текущим предка текущего узла; $j := B[i]$; // возврат по дереву
 - 12: **Если** $i > 0$, **то**
 $i := i - 1$, перейти к п. 9;
 - 13: **иначе**
 - 14: // вернулись в корень и рассмотрели все возможные дуги
 выход, ответ «Функция g не обратима типа $\forall\exists\forall$ по переменной x_1 ».
 - 15: **Для** $i = 1, \dots, n$:
 - 16: положить $f(x) = a_i$ для всех $x \in G^{(a_i, b_{B[i]})}$.
-

Т а б л и ц а 8

x_1	x_2	x_3	g
1	4	0	1
1	4	1	1
1	5	0	5
1	5	1	1
2	4	0	4
2	4	1	0
2	5	0	4
2	5	1	5
3	4	0	0
3	4	1	0
3	5	0	1
3	5	1	0

Т а б л и ц а 9

x	$f(x)$
0	3
1	1
4	2
5	2

Т а б л и ц а 10

$a \in D_1$	$\varphi(a)$
1	4
2	5
3	4

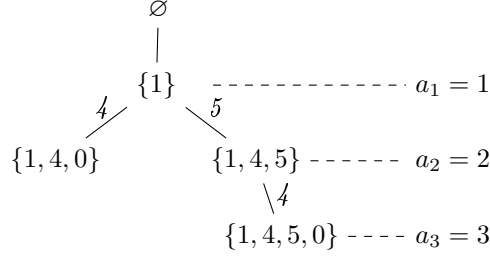


Рис. 4. Иллюстрация алгоритма 5

В алгоритме 6 описан способ генерации функции $g : D_1 \times D_2 \times D_3 \rightarrow D$, обратной типа $\forall\exists\forall$ по переменной x_1 .

Алгоритм 6. Генерация функции, обратной типа $\forall\exists\forall$ по переменной x_1

- 1: Построить произвольное разбиение множества D на классы H_a , $a \in D_1$.
 - 2: **Для всех** $a \in D_1$:
 - 3: выбрать $b \in D_2$.
 - 4: **Для всех** $c \in D_3$:
 - 5: выбрать $z \in H_a$, положить $g(a, b, c) := z$.
 - 6: **Для всех** $x_2 \in D_2 \setminus \{b\}$:
 - 7: **Для всех** $x_3 \in D_3$:
 - 8: выбрать в качестве $g(a, x_2, x_3)$ произвольное значение $y \in D$.
-

2.3. Обратимость типа $\forall\exists\forall$ по переменной x_3

Функция $g(x_1, x_2, x_3)$ обратима типа $\forall\exists\forall$ по переменной x_3 , если существует функция $f : D \rightarrow D_3$, такая, что

$$\forall x_1 \exists x_2 \forall x_3 (f(g(x_1, x_2, x_3)) = x_3). \quad (17)$$

Утверждение 4 (критерий обратимости типа $\forall\exists\forall$ по x_3 [7]). Функция $g : D_1 \times D_2 \times D_3 \rightarrow D$ обратима типа $\forall\exists\forall$ по переменной x_3 , если и только если существует такое отображение $\varphi : D_1 \rightarrow D_2$, что выполнено условие

$$\forall a, c \in D_1 \forall b, d \in D_3 (b \neq d \Rightarrow g(a, \varphi(a), b) \neq g(c, \varphi(c), d)). \quad (18)$$

Доказательство полностью аналогично доказательству утверждения 3, за одним исключением: в шаге 1 построения функции восстановления f полагаем $f(g(a, \varphi(a), b)) = b$. ■

Алгоритм 7 строит функцию восстановления f , удовлетворяющую (17). Пусть $D_1 = \{a_1, \dots, a_n\}$, $D_2 = \{b_1, \dots, b_m\}$, $D_3 = \{c_1, \dots, c_k\}$. Функция f строится с помощью обхода дерева в глубину. Как и в алгоритме 5, ярусы дерева соответствуют элементам из D_1 , дуги — элементам из D_2 . Для $a \in D_1$, $b \in D_2$ через $v(a, b)$ обозначим вектор значений подфункции $g(a, b, \cdot)$: $v(a, b) = (g(a, b, c_1), \dots, g(a, b, c_k)) \in D^k$.

Вершине i -го яруса сопоставим множество векторов $V = \{v(a_r, b_{B[r]}) : r = 1, \dots, i\}$, т.е. векторов $v(a, b)$ «вдоль» пути от корня до данной вершины. Для обеспечения (17) потребуем выполнения следующих условий:

- 1) в каждом векторе $v(a, b) \in V$ все координаты различны (что эквивалентно равенству $|G^{(a,b)}| = k$);
- 2) для любых $v = (v_1, \dots, v_k), w = (w_1, \dots, w_k) \in V$ и для всех $i, j \in \{1, \dots, k\}$ если $v_i = w_j$, то $i = j$.

Алгоритм 7. Построение функции восстановления для обратимости типа $\forall\exists\forall$ по переменной x_3

Вход: функция $g : D_1 \times D_2 \times D_3 \rightarrow D$.

Выход: функция f , удовлетворяющая (17), или ответ «Функция g не обратима типа $\forall\exists\forall$ по переменной x_3 ».

- 1: Построить корень дерева с меткой $V = \emptyset$, объявить его текущим узлом; $i := 1$.
 - 2: $j := 1$.
 - 3: **Если** $|G^{(a_i, b_j)}| < k$, **то**
 перейти к п. 11.
 - 4: Положить $w := (w_1, \dots, w_k) = (g(a_i, b_j, c_1), \dots, g(a_i, b_j, c_k))$.
 - 5: **Если** $\forall s, t \in \{1, \dots, k\} \forall v = (v_1, \dots, v_k) \in V (w_s = v_t \Rightarrow s = t)$, **то**
 добавить потомка с меткой $V \cup \{w\}$, объявить его текущим узлом; $B[i] := j$,
 $i := i + 1$.
 - 6: **Если** $i = n + 1$, **то**
 - 7: перейти к п. 17, // все ярусы пройдены
 - 8: **иначе**
 - 9: перейти к п. 2; // обход в глубину
 - 10: **иначе**
 - 11: **Если** $j < m$, **то**
 $j := j + 1$, перейти к п. 3; // построение дерева в ширину
 - 12: **иначе**
 - 13: объявить текущим предка текущего узла; $j := B[i]$. // возврат по дереву
 - 14: **Если** $i > 0$, **то**
 $i := i - 1$, перейти к п. 11;
 - 15: **иначе**
 - 16: **выход**, ответ «Функция g не обратима типа $\forall\exists\forall$ по переменной x_3 ».
 - 17: **Для всех** $v \in V$:
 - 18: положить $f(v_i) = c_i$.
-

Для построения отображения φ , удовлетворяющего (18), на шаге 18 алгоритма 7 нужно положить $\varphi(a_i) = b_{B[i]}$, $i = 1, \dots, n$.

Нетрудно убедиться, что функции g из примеров 4 и 5 не являются обратимыми типа $\forall\exists\forall$ по переменной x_3 .

Пример 6. Пусть функция $g : \{1, 2, 3\} \times \{4, 5\} \times \{0, 1\} \rightarrow \{0, 1, 4, 5\}$ задана в табл. 11. На рис. 5 представлено дерево, построенное алгоритмом 6; в табл. 12 и 13 — функция f и отображение φ , удовлетворяющие условиям (17) и (18) соответственно.

Т а б л и ц а 11

x_1	x_2	x_3	g
1	4	0	1
1	4	1	5
1	5	0	5
1	5	1	5
2	4	0	1
2	4	1	4
2	5	0	1
2	5	1	0
3	4	0	4
3	4	1	5
3	5	0	0
3	5	1	1

Т а б л и ц а 12

x	$f(x)$
0	1
1	0
4	0
5	1

Т а б л и ц а 13

$a \in D_1$	$\varphi(a)$
1	4
2	5
3	4

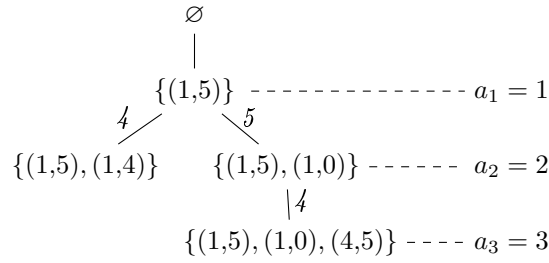


Рис. 5. Иллюстрация алгоритма 6

Алгоритм 8 строит функцию $g : D_1 \times D_2 \times D_3 \rightarrow D$, обратимую типа $\forall\exists\forall$ по переменной x_3 .

Алгоритм 8. Генерация функции, обратимой типа $\forall\exists\forall$ по переменной x_3

- 1: Построить произвольное разбиение множества D на классы H_c , $c \in D_3$.
 - 2: **Для всех** $a \in D_1$:
 - 3: выбрать $b \in D_2$.
 - 4: **Для всех** $c \in D_3$:
 - 5: выбрать $z \in H_c$, положить $g(a, b, c) := z$.
 - 6: **Для всех** $x_2 \in D_2 \setminus \{b\}$:
 - 7: **Для всех** $x_3 \in D_3$:
 - 8: выбрать в качестве $g(a, x_2, x_3)$ произвольное значение $y \in D$.
-

2.4. Обратимость типа $\exists\forall\forall$ по переменной x_2

Функция $g(x_1, x_2, x_3)$ обратима типа $\exists\forall\forall$ по переменной x_2 , если

$$\exists f \exists x_1 \forall x_2 \forall x_3 (f(g(x_1, x_2, x_3)) = x_2),$$

или, что то же самое, если

$$\exists x_1 \exists f \forall x_2 \forall x_3 (f(g(x_1, x_2, x_3)) = x_2),$$

что равносильно обратимости типа $\forall\forall$ по переменной x_2 подфункции $g(a, x_2, x_3)$ для некоторого $a \in D_1$. Тогда из п. 1 утверждения 1 следует

Утверждение 5 (критерий обратимости типа $\exists\forall\forall$). Функция $g : D_1 \times D_2 \times D_3 \rightarrow D$ обратима типа $\exists\forall\forall$ по переменной x_2 , если и только если существует такое значение $a \in D_1$, для которого выполнено условие

$$\forall b, d \in D_2 (b \neq d \Rightarrow G^{(a,b)} \cap G^{(a,d)} = \emptyset). \quad (19)$$

Функция восстановления $f : D \rightarrow D_2$ строится следующим образом:

- 1) $f(g(a, b, c)) = b$ для $a \in D_1$, удовлетворяющего условию (19), и всех $b \in D_2, c \in D_3$;
- 2) $f(x)$ равна любому значению из D_2 для тех $x \in D$, значения на которых не определены на шаге 1.

Нетрудно видеть, что функция, заданная табл. 6, ни для одного $a \in \{1, 2, 3\}$ условию (19) не удовлетворяет:

$$\begin{aligned} G^{(1,4)} &= \{0, 1\}, & G^{(1,5)} &= \{4, 0\}; \\ G^{(2,4)} &= \{1, 4\}, & G^{(2,5)} &= \{0, 4\}; \\ G^{(3,4)} &= \{1, 0\}, & G^{(3,5)} &= \{5, 0\}; \end{aligned}$$

аналогично — для функции из табл. 8. Для функции g из табл. 11 и $a = 3$ условие выполнено: $G^{(3,4)} = \{4, 5\}$ и $G^{(3,5)} = \{0, 1\}$. Следовательно, она обратима типа $\exists\forall\forall$ по переменной x_2 ; функция восстановления f представлена в табл. 14.

Т а б л и ц а 14

x	$f(x)$
0	5
1	5
4	4
5	4

Способ генерации функции $g : D_1 \times D_2 \times D_3 \rightarrow D$, обратимой типа $\exists\forall\forall$ по переменной x_2 , описан в алгоритме 9.

Алгоритм 9. Генерация функции, обратимой типа $\exists\forall\forall$ по переменной x_2

- 1: Выбрать случайное $a \in D_1$.
 - 2: Построить произвольное разбиение множества D на классы $H_b, b \in D_2$.
 - 3: **Для всех** $b \in D_2$:
 - 4: **Для всех** $c \in D_3$:
 - 5: выбрать $z \in H_b$, положить $g(a, b, c) := z$.
 - 6: **Для всех** $x_1 \in D_1 \setminus \{a\}$:
 - 7: **Для всех** $x_2 \in D_2$:
 - 8: **Для всех** $x_3 \in D_3$:
 - 9: выбрать в качестве $g(x_1, x_2, x_3)$ произвольное значение $y \in D$.
-

2.5. Обратимость типа $\exists\forall\exists$ по переменной x_2

Функция $g(x_1, x_2, x_3)$ обратима типа $\exists\forall\exists$ по переменной x_2 , если

$$\exists f \exists x_1 \forall x_2 \exists x_3 (f(g(x_1, x_2, x_3)) = x_2).$$

Аналогично п. 2.4, это равносильно обратимости типа $\forall\exists$ подфункции $g(a, x_2, x_3)$ для некоторого $a \in D_1$. Тогда из п. 3 утверждения 1 следует

Утверждение 6 (критерий обратимости типа $\exists\forall\exists$ [7]). Функция $g : D_1 \times D_2 \times D_3 \rightarrow D$ обратима типа $\exists\forall\exists$ по переменной x_2 , если и только если существуют такие значение $a \in D_1$ и отображение $\varphi : D_2 \rightarrow D_3$, для которых выполнено условие

$$\forall b, d \in D_2 (b \neq d \Rightarrow g(a, b, \varphi(b)) \neq g(a, d, \varphi(d))). \quad (20)$$

Функция восстановления $f : D \rightarrow D_2$ строится следующим образом:

- 1) $f(g(a, b, \varphi(b))) = b$ для $a \in D_1$, удовлетворяющего условию (20), и всех $b \in D_2$;
- 2) $f(x)$ равна любому значению из D_2 для тех $x \in D$, значения на которых не определены на шаге 1.

Проверить обратимость функции g можно, применив алгоритм 1 (построения отображения φ) или алгоритм 2 (построения функции восстановления) поочерёдно для всех подфункций $g(a, x_2, x_3)$, $a \in D_1$.

Ввиду общезначимости формулы $(\forall x P(x) \Rightarrow \exists x P(x))$, где $P(x)$ — произвольный предикат с непустой областью определения, получаем: из обратимости функции $g(x_1, x_2, x_3)$ типа $\exists\forall\forall$ по переменной x_2 следует её обратимость типа $\exists\forall\exists$ по той же переменной. Обратное в общем случае неверно; так, например, функция из табл. 8, которая не является обратимой типа $\exists\forall\forall$, обратима типа $\exists\forall\exists$. Отображение φ , удовлетворяющее условию (20) для $a = 1$, и соответствующая функция восстановления f представлены в табл. 15 и 16; символы «*» в табл. 16 означают, что $f(0)$ и $f(4)$ могут принимать любые значения из $\{4, 5\}$.

Т а б л и ц а 15

$a \in D_2$	$\varphi(a)$
4	0
5	0

Т а б л и ц а 16

x	$f(x)$
0	*
1	4
4	*
5	5

Способ генерации функции $g : D_1 \times D_2 \times D_3 \rightarrow D$, обратимой типа $\exists\forall\exists$ по переменной x_2 , описан в алгоритме 10.

Алгоритм 10. Генерация функции, обратимой типа $\exists\forall\exists$ по переменной x_2

- 1: Выбрать случайное $a \in D_1$.
 - 2: $Z := D$.
 - 3: **Для всех** $b \in D_2$:
 - 4: выбрать $c \in D_3$, $z \in Z$;
 - 5: положить $g(a, b, c) := z$;
 - 6: $Z := Z \setminus \{z\}$.
 - 7: **Для всех** $x_3 \in D_3 \setminus \{c\}$:
 - 8: выбрать в качестве $g(a, b, x_3)$ произвольное значение $y \in D$.
 - 9: **Для всех** $x_1 \in D_1 \setminus \{a\}$:
 - 10: **Для всех** $x_2 \in D_2$:
 - 11: **Для всех** $x_3 \in D_3$:
 - 12: выбрать в качестве $g(x_1, x_2, x_3)$ произвольное значение $y \in D$.
-

ЛИТЕРАТУРА

1. Agibalov G. P. Cryptanalytical finite automaton invertibility with finite delay // Прикладная дискретная математика. 2019. № 46. С. 27–37.

2. *Agibalov G. P.* Problems in theory of cryptanalytical invertibility of finite automata // Прикладная дискретная математика. 2020. № 50. С. 62–71.
3. *Бердникова Н. Ю., Панкратова И. А.* Кriptoаналитическая обратимость функций двух аргументов // Прикладная дискретная математика. Приложение. 2021. № 14. С. 67–71.
4. *Фомичев В. М.* Методы дискретной математики в криптологии. М.: Диалог-МИФИ, 2010. 424 с.
5. *Агибалов Г. П.* О криптоаналитической обратимости с конечной задержкой конечных автоматов // Прикладная дискретная математика. Приложение. 2019. № 12. С. 84–86.
6. *Черемушкин А. В.* Обобщённые тождества медиальности и парамедиальности для сильно зависимых операций // Прикладная дискретная математика. 2024. № 65. С. 21–40.
7. *Панкратова И. А., Сорокоумова А. Д.* Кriptoаналитическая обратимость функций трёх аргументов // Прикладная дискретная математика. Приложение. 2024. № 17. С. 44–48.

REFERENCES

1. *Agibalov G. P.* Cryptanalytical finite automaton invertibility with finite delay. Prikladnaya Diskretnaya Matematika, 2019, no. 46, pp. 27–37.
2. *Agibalov G. P.* Problems in theory of cryptanalytical invertibility of finite automata. Prikladnaya Diskretnaya Matematika, 2020, no. 50, pp. 62–71.
3. *Berdnikova N. Yu. and Pankratova I. A.* Kriptoanaliticheskaya obratimost' funktsiy dvukh argumentov [Cryptanalytic invertibility of two-argument functions]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2021, no. 14, pp. 67–71. (in Russian)
4. *Fomichev V. M.* Metody diskretnoy matematiki v kriptologii [Methods of Discrete Mathematics in Cryptology]. Moscow, Dialog-MIFI Publ., 2010. 424 p. (in Russian)
5. *Agibalov G. P.* O kriptoanaliticheskoy obratimosti s konechnoy zaderzhkoy konechnykh avtomatov [Cryptanalytic invertibility with finite delay of finite automata]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2019, no. 12, pp. 84–86.
6. *Cheremushkin A. V.* Obobshchennye tozhdestva medial'nosti i paramedial'nosti dlya sil'no zavisimykh operatsiy [Medial and paramedial general identities for strong dependance operations]. Prikladnaya Diskretnaya Matematika, 2024, no. 65, pp. 21–40. (in Russian)
7. *Pankratova I. A. and Sorokoumova A. D.* Kriptoanaliticheskaya obratimost' funktsiy trekh argumentov [Cryptanalytic invertibility of three-argument functions]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2024, no. 17, pp. 44–48. (in Russian)