ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 55 **2019** **Вып. 3**

УДК 621.391.15 : 519.72

© 2019 г. **С.С. Марченков**

ОБ АЛФАВИТНОМ КОДИРОВАНИИ СВЕРХСЛОВ[1]

Рассматривается алфавитное кодирование сверхслов. Устанавливаются критерии однозначности кодирования для случаев конечного и бесконечного кодов. Доказывается, что в случае бесконечного кода проблема распознавания неоднозначности кода является m-полной в классе $\exists^1\forall^0$ аналитической иерархии Клини.

Ключевые слова: алфавитное кодирование, сверхслово, аналитическая иерархия Клини.

DOI: 10.1134/S0555292319030069

В теории кодирования хорошо известен результат Ал.А. Маркова [1–3] – критерий однозначности алфавитного кодирования. Этот результат можно обобщать на сверхслова в двух направлениях: находить критерии однозначности алфавитного кодирования для кодирования сверхслов конечным кодом либо для кодирования сверхслов бесконечным кодом. Эти два направления существенно отличаются по сложности получаемых результатов. Если для конечного кода сложность критерия близка к сложности критерия из [1–3], то для бесконечного кода возникают проблемы, связанные как с заданием кода, так и с описанием однозначных кодов.

В данной статье мы сначала получаем критерий однозначности кода для кодирования сверхслов конечным кодом (теорема 1) и переносим его на случай бесконечных кодов (теорема 2). Затем рассматриваем сложность проблемы распознавания однозначности бесконечного кода. В теореме 3 в терминах аналитической иерархии Клини устанавливается "верхняя оценка" сложности этой проблемы, а в теореме 4 с использованием функциональных уравнений – такая же "нижняя оценка".

Введем необходимые понятия. Пусть A – конечный алфавит. Через A^* обозначим множество всех (конечных) слов в алфавите A, включая пустое слово Λ, а через A^∞ – множество всех сверхслов в алфавите A (упорядоченных счетно-бесконечных последовательностей, составленных из букв алфавита A). Если $\bar{a} \in A^*$, то пусть $|\bar{a}|$ обозначает длину слова $\bar{a}$.

Если $A = \{a_1, \ldots, a_k\}$ и B – алфавит, то *алфавитным кодированием* (из A в B) называется отображение

$$\varphi\colon A \to B^* \setminus \{\Lambda\}.$$

Отображение φ продолжается на множество A^*: если $a_{i_1} \ldots a_{i_\ell} \in A$, то

$$\varphi(a_{i_1} \ldots a_{i_\ell}) = \varphi(a_{i_1}) \ldots \varphi(a_{i_\ell}).$$

Считается также, что $\varphi(\Lambda) = \Lambda$.

[1] Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (номер проекта 19-01-00200).

Обычно в кодировании φ рассматривают только множество

$$C(\varphi) = \{\varphi(a_1), \ldots, \varphi(a_k)\},$$

которое, собственно, и называют *кодом.* В дальнейшем код $C(\varphi)$ будем записывать в виде $\mathcal{C} = \{C_1, \ldots, C_k\}$. Код $\mathcal{C}$ называется *однозначным*, если различным последовательностям $(i_1, \ldots, i_m)$ из $\{1, 2, \ldots, k\}^* \setminus \{\Lambda\}$ соответствуют различные слова $C_{i_1} \ldots C_{i_m}$.

Далее будем рассматривать кодирование и декодирование сверхслов. Это означает, в частности, что последовательности $(i_1, i_2, \ldots)$ берутся из множества $\{1, \ldots, k\}^\infty$, а соответствующие им сверхслова $C_{i_1} C_{i_2} \ldots$ принадлежат множеству B^∞. Понятие однозначности кода естественным образом переносится на сверхслова.

С кодом $\mathcal{C} = \{C_1, \ldots, C_k\}$ свяжем множество $S = \{\beta_0, \beta_1, \ldots, \beta_\ell\}$, состоящее из пустого слова Λ (считаем, что $\beta_0 = \Lambda$) и всех непустых слов $\beta_1, \ldots, \beta_\ell$, которые одновременно являются собственными (отличными от кодовых слов) префиксами и суффиксами подходящих кодовых слов. Определим ориентированный граф $G_\mathcal{C}$ с $\ell + 1$ вершинами, которые помечены словами множества S. Для упрощения изложения вершины графа $G_\mathcal{C}$ будем отождествлять с соответствующими словами из S. Для произвольных двух вершин β_i, β_j (допускается $i = j$) дуга от вершины β_i к вершине β_j существует в графе $G_\mathcal{C}$ только в том случае, когда найдутся такие кодовые слова $C_p, C_{q_1}, \ldots, C_{q_s}$, что

$$C_p = \beta_i C_{q_1} \ldots C_{q_s} \beta_j.$$

При этом если $i \neq 0$ и $j \neq 0$, то возможно $s = 0$; если только один из индексов i, j равен 0, то должно быть $s \geqslant 1$; если же $i = j$, то должно быть $s \geqslant 2$. Считаем, что в случае существования дуги из вершины β_i в вершину β_j на ней выписана соответствующая последовательность кодовых слов $C_{q_1} \ldots C_{q_s}$. Обычным образом вводится понятие ориентированного пути в графе $G_\mathcal{C}$. Длиной пути считаем число входящих в него дуг.

Доказательство теоремы 1 проводится в духе доказательства соответствующего результата при кодировании конечных слов [1–3].

Теорема 1. *Алфавитный код $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ является однозначным тогда и только тогда, когда в графе $G_\mathcal{C}$ отсутствуют ориентированные пути длины $\ell + 1$, выходящие из вершины Λ.*

Доказательство. Предположим сначала, что в графе $G_\mathcal{C}$ имеется ориентированный путь Γ длины $\ell+1$, выходящий из вершины Λ. Тогда среди вершин, входящих в путь Γ, по крайне мере одна вершина встречается дважды. Пусть β_i – первая (от вершины Λ) вершина пути Γ, которая входит в путь Γ не один раз. Рассмотрим подпуть Γ' пути Γ, который начинается в вершине Λ и оканчивается вторым вхождением вершины β_i в путь Γ. Образуем из Γ' бесконечный путь Δ, повторяя после вершины β_i бесконечное число раз цикл из вершины β_i в вершину β_i. Пусть сверхслово W получается последовательным выписыванием всех слов, стоящих в вершинах и дугах пути Δ. Утверждается, что сверхслово W допускает по крайней мере два декодирования.

В самом деле, предположим, что указанное сверхслово W имеет вид

$$\beta_0 C_1^{(1)} \ldots C_{s_1}^{(1)} \beta_{j_1} C_1^{(2)} \ldots C_{s_2}^{(2)} \beta_{j_2} \ldots \beta_{j_r} C_1^{(r+1)} \ldots C_{s_{r+1}}^{(r+1)} \beta_{j_{r+1}} \ldots,$$

где в сверхслове W обозначены все вхождения слов β_{j_m} из множества S и кодовых слов $C_q^{(p)}$ (выписанных на дугах пути Δ). Тогда согласно определению графа $G_\mathcal{C}$ одно декодирование сверхслова W имеет вид

$$C_1^{(1)} \ldots C_{s_1}^{(1)} \beta_{j_1},\ C_1^{(2)}, \ldots,\ C_{s_2}^{(2)},\ \beta_{j_2} C_1^{(3)} \ldots C_{s_3}^{(3)} \beta_{j_3},\ \ldots,$$

а другое –

$$C_1^{(1)}, \ldots, C_{s_1}^{(1)},\ \beta_{j_1} C_1^{(2)} \ldots C_{s_2}^{(2)} \beta_{j_2},\ C_1^{(3)}, \ldots, C_{s_3}^{(3)},\ \ldots$$

Обратно, предположим, что существует сверхслово W, которое допускает два различных декодирования:

$$W = C_{i_1} C_{i_2} \ldots C_{i_n} \ldots = C_{j_1} C_{j_2} \ldots C_{j_n} \ldots$$

Если для двух различных последовательностей $i_1, i_2, \ldots, i_n$ и $j_1, j_2, \ldots, j_m$ выполняется равенство

$$C_{i_1} C_{i_2} \ldots C_{i_n} = C_{j_1} C_{j_2} \ldots C_{j_m}, \tag{1}$$

то код $\mathcal{C}$ не является однозначным уже для слов конечной длины. Тогда, как известно [1–3], в графе $G_{\mathcal{C}}$ существует ориентированный цикл, проходящий через вершину Λ. Очевидно, что в этом случае в графе $G_{\mathcal{C}}$ будут существовать и ориентированные пути сколь угодно большой длины, выходящие из вершины Λ.

Далее предполагаем, что равенство (1) не выполняется ни для каких различных последовательностей $i_1, i_2, \ldots, i_n$ и $j_1, j_2, \ldots, j_m$. Это означает, в частности, что при любом n конец слова $C_{i_1} C_{i_2} \ldots C_{i_n}$ в сверхслове W лежит внутри некоторого слова C_{j_m}. То же самое справедливо и для слов $C_{j_1} C_{j_2} \ldots C_{j_m}$.

Рассмотрим слова C_{i_1} и C_{j_1}. При $|C_{i_1}| = |C_{j_1}|$ получаем $i_1 = j_1$, и сверхслово можно "сократить" на слово C_{i_1}. Поэтому при $|C_{i_1}| > |C_{j_1}|$ слово C_{i_1} представимо в виде $C_{i_1} = C_{j_1} \ldots C_{j_m} \beta_1$, где $m \geqslant 1$ и β_1 – собственный префикс слова $C_{j_{m+1}}$, который, очевидно, является также собственным суффиксом слова C_{i_1}. В случае $|C_{i_1}| < |C_{j_1}|$ справедливо симметричное представление $C_{j_1} = C_{i_1} \ldots C_{i_n} \beta_1$, где β_1 – собственный префикс слова $C_{i_{n+1}}$ и собственный суффикс слова C_{j_1}.

Таким образом, в графе $G_{\mathcal{C}}$ вершина Λ будет соединена дугой с вершиной β_1, при этом на дуге выписано слово $C_{j_1} \ldots C_{j_m}$ либо слово $C_{i_1} \ldots C_{j_n}$, составленные из кодовых слов кода $\mathcal{C}$.

Далее продолжаем по индукции. Пусть в сверхслове W уже найдено подслово β_p (оно может совпадать с ранее найденным словом β_q), которое является как собственным префиксом кодового слова из $\mathcal{C}$, так и собственным суффиксом (вообще говоря, другого) кодового слова из $\mathcal{C}$. И пусть слово β_p либо является собственным префиксом слова $C_{i_{n+1}}$ (входящего в начало $C_{i_1} \ldots C_{i_n} C_{i_{n+1}}$ сверхслова W) и собственным суффиксом слова C_{j_m} (входящего в начало $C_{j_1} \ldots C_{j_m}$ сверхслова W), либо наоборот – собственным суффиксом слова C_{i_n} (входящего в начало $C_{i_1} \ldots C_{i_n}$ сверхслова W) и собственным префиксом слова $C_{j_{m+1}}$ (входящего в начало $C_{j_1} \ldots C_{j_m} C_{j_{m+1}}$ сверхслова W). Наконец, предположим, что в графе $G_{\mathcal{C}}$ имеется ориентированный путь, ведущий из вершины Λ в вершину β_p.

Последующий этап доказательства использует только тот факт, что слово C_{j_m} заканчивается внутри слова $C_{i_{n+1}}$, имея с данным словом непустое пересечение β_p (первый случай), или слово C_{i_n} заканчивается внутри слова $C_{j_{m+1}}$, пересекаясь с этим словом по слову β_p (второй случай). Поэтому, как и в базисе индукции, рассматриваем либо расположение сверхслова $C_{j_{m+1}} \ldots$ по отношению к слову $C_{i_{n+1}}$, либо расположение сверхслова $C_{i_{n+1}} \ldots$ по отношению к слову $C_{j_{m+1}}$. Соответствующие рассуждения полностью аналогичны рассуждениям из базиса индукции, и мы их опускаем. ▲

Очевидно, что на основе доказанной теоремы легко определить алгоритм проверки однозначности алфавитного кодирования сверхслов. Отметим еще, что в формулировке теоремы 1 можно с равным успехом говорить об отсутствии в графе $G_{\mathcal{C}}$ ориентированных циклов, "достижимых" из вершины Λ: граф $G_{\mathcal{C}}$ имеет $\ell+1$ вершин, и потому существование в графе $G_{\mathcal{C}}$ ориентированного пути длины $\ell+1$ равносильно существованию в нем достижимого из вершины Λ ориентированного цикла.

Всюду далее мы будем рассматривать алфавитные коды $\mathcal{C} = \{C_1, C_2, \ldots\}$, состоящие из бесконечного числа кодовых слов. Так же, как для конечного случая, определяем множество S и ориентированный граф $G_{\mathcal{C}}$. Однако теперь как множество S, так и граф $G_{\mathcal{C}}$ могут быть бесконечными. Вместе с тем кодировать кодом C мы будем только сверхслова.

Теорема 2 представляет собой несложное обобщение теоремы 1, и потому ее доказательство мы не приводим.

Т е о р е м а 2. *Алфавитное кодирование $\mathcal{C} = \{C_1, C_2, \ldots\}$ является однозначным тогда и только тогда, когда в графе G_C отсутствуют ориентированные пути бесконечной длины, выходящие из вершины Λ.*

В оставшейся части статьи мы хотим оценить сложность распознавания однозначности кодирования. Для этого нам придется некоторым стандартным образом задавать бесконечные множества $\{C_1, C_2, \ldots\}$ кодовых слов. Примем следующее соглашение: множество $\{C_1, C_2, \ldots\}$ задается одноместной функцией f с областью определения B^* (напомним, что кодовые слова $C_1, C_2, \ldots$ рассматриваются в алфавите B) и областью значений $\{0,1\}$. При этом для любого слова $C \in B^*$ равенство $f(C) = 1$ выполняется тогда и только тогда, когда $C \in \mathcal{C}$. Таким образом, f представляет собой характеристическую функцию кода $\mathcal{C}$.

Пусть $\mathbb{N} = \{0, 1, \ldots\}$, и пусть $P_{\mathbb{N}}$ обозначает множество всех (всюду определенных) функций на $\mathbb{N}$. Функции из $P_{\mathbb{N}}$ будем называть функциями счетнозначной логики. Для любого $n \geqslant 1$ и любого множества $Q \subseteq P_{\mathbb{N}}$ обозначаем через $Q^{(n)}$ множество всех n-местных функций из Q.

Обозначим через $U(f)$ отношение “код, определяемый функцией f, не является однозначным”. Чтобы оценить сложность отношения $U(f)$, воспользуемся классом $\exists^1\forall^0$ аналитической иерархии Клини (см., например, [4]). Напомним, что класс $\exists^1\forall^0$ (индекс 1 означает, что квантификация проводится по функциональным переменным, а индекс 0 – по числовым) состоит из всех отношений, которые представимы в виде

$$(\exists^1 g_1)\ldots(\exists^1 g_m)(\forall^0 y_1)\ldots(\forall^0 y_n) R(f_1, \ldots, f_k, g_1, \ldots, g_m, x_1, \ldots, x_\ell, y_1, \ldots, y_n),$$

где $f_1, \ldots, f_k, g_1, \ldots, g_m$ – переменные для функций из $P_{\mathbb{N}}$, $x_1, \ldots, x_\ell, y_1, \ldots, y_n$ – числовые переменные, принимающие значения из $\mathbb{N}$, а R – рекурсивное отношение (от функциональных и числовых переменных).

Т е о р е м а 3. *Отношение $U(f)$ принадлежит классу $\exists^1\forall^0$ аналитической иерархии Клини.*

Д о к а з а т е л ь с т в о. Используем критерий однозначности кода, установленный в теореме 2. Отношение $U(f)$ представим в форме

$$(\exists^1 g)(\forall^0 y) R(f, g, y).$$

Содержательно функция $g(y)$ будет определять бесконечную ветвь в дереве $G_{\mathcal{C}}$, где кодирование $\mathcal{C}$ задается функцией f. Более подробно: по каждому числу $y \geqslant 0$ функция g будет выдавать номер (в некоторой стандартной вычислимой нумерации ν множества всех упорядоченных наборов слов в алфавите B) $(y+1)$-го набора из бесконечной последовательности

$$(\beta_0, C^{(1)}_{i_1}, \ldots, C^{(1)}_{i_p}, \beta_{t_1}),\ (\beta_{t_1}, C^{(2)}_{j_1}, \ldots, C^{(2)}_{j_q}, \beta_{t_2}),\ (\beta_{t_2}, C^{(3)}_{k_1}, \ldots, C^{(3)}_{k_r}, \beta_{t_3}),\ \ldots, \qquad (2)$$

где $\beta_0, \beta_{t_1}, \beta_{t_2}, \dots$ – слова из множества S, построенного для кодирования $\mathcal{C}$, $\beta_0 = \Lambda$, $C_\ell^{(d)}$ – кодовые слова, и каждое из слов

$$\beta_0 C_{i_1}^{(1)} \dots C_{i_p}^{(1)} \beta_{t_1}, \quad \beta_{t_1} C_{j_1}^{(2)} \dots C_{j_q}^{(2)} \beta_{t_2}, \quad \beta_{t_2} C_{k_1}^{(3)} \dots C_{k_r}^{(3)} \beta_{t_3}, \quad \dots \tag{3}$$

также является кодовым словом. В соответствии с данным описанием для каждого y отношение R "находит" набор слов с номером $g(y)$, затем "проверяет", что выбранный набор имеет вид, указанный в последовательности (2). Для этого отношение R "выполняет" следующие действия.

Прежде всего, проверяет, что все слова набора с номером $g(y)$, отличные от первого и последнего слова, являются кодовыми словами. Это действие легко выполнить, обращаясь к функции f (напомним, что f является характеристической функцией рассматриваемого кода). Далее при $y \neq 0$ отношение R находит наборы с номерами $g(y-1)$, $g(y+1)$ и убеждается, что последнее слово набора $\nu(g(y-1))$ совпадает с первым словом набора $\nu(g(y))$, а последнее слово данного набора – с первым словом набора $\nu(g(y+1))$ (если $y = 0$, то первое слово набора $\nu(g(y))$ должно быть пустым, а последнее – совпадать с первым словом набора $\nu(g(1))$). При этом дополнительно проверяется, что в случае совпадения с пустым словом ровно одного из крайних слов набора $\nu(g(y))$ этот набор содержит не менее трех слов, а в случае совпадения с пустым словом обоих крайних слов набора $\nu(g(y))$ – не менее четырех. Затем отношение R "склеивает" все слова набора $\nu(g(y))$ и проверяет, что полученное слово (которое присутствует в последовательности (3)) является кодовым словом (при этом вновь используется функция f). Заметим, что выполняемые действия гарантируют, что все непустые слова последовательности $\beta_{t_1}, \beta_{t_2}, \dots$ являются одновременно собственными префиксами и собственными суффиксами кодовых слов кода $\mathcal{C}$.

Нетрудно теперь понять, что если для данной функции f (принимающей только значения 0 и 1) найдется такая фукция g, что при любом y выполняется определенное выше отношение $R(f, g, y)$, то функция f представляет собой характеристическую функцию кода, не являющегося однозначным. Понятно также, что при несуществовании функции g с указанными свойствами соответствующий код будет однозначным. ▲

Теорему 3 можно рассматривать как "оценку сверху" сложности отношения $U(f)$. Чтобы показать, что отношение $U(f)$ имеет такую же "оценку снизу", нам придется ввести некоторую формализацию процесса "вычисления" произвольного отношения вида

$$(\exists^1 g)(\forall^0 y) R(f, g, y).$$

По ряду соображений технического характера нам будет удобно воспользоваться формализмом функциональных уравнений счетнозначной логики [5, 6].

Определим язык $\mathcal{L}_{\mathbb{N}}$ функциональных уравнений. Предполагаем, что каждая функция из $P_{\mathbb{N}}$ имеет индивидуальное обозначение. Для обозначения n-местных функций из $P_{\mathbb{N}}$ используем символы $f_\nu^{(n)}$, которые называем функциональными константами. Наряду с функциональными константами рассматриваем функциональные переменные. Для обозначения n-местных функциональных переменных используем символы $\varphi_i^{(n)}$. Областью значений функциональной переменной $\varphi_i^{(n)}$ служит множество $P_{\mathbb{N}}^{(n)}$. В случае, когда это не приводит к недоразумению, верхние индексы у функциональных констант и функциональных переменных будем опускать. Помимо функциональных переменных используем обычные предметные (числовые) переменные $x_1, x_2, \dots$ с областью значений $\mathbb{N}$.

Язык $\mathcal{L}_{\mathbb{N}}$ функциональных уравнений состоит из предметных переменных x_i ($i = 1, 2, \dots$), функциональных переменных $\varphi_i^{(n)}$ ($i, n = 1, 2, \dots$), функциональных констант $f_\nu^{(n)}$, знака равенства $=$, левой и правой скобок и запятой.

Пусть $Q \subseteq P_{\mathbb{N}}$. Определим понятие терма над Q. Всякая предметная переменная есть терм над Q. Если $t_1, \dots, t_n$ – термы над Q, $f_\nu^{(n)}$ – функциональная константа, служащая обозначением функции из Q, а $\varphi_i^{(n)}$ – функциональная переменная, то выражения

$$f_\nu^{(n)}(t_1, \dots, t_n), \quad \varphi_i^{(n)}(t_1, \dots, t_n)$$

суть термы над Q. Равенством над Q называем любое выражение вида $t_1 = t_2$, где t_1, t_2 – термы над Q. Равенства над Q называем также функциональными уравнениями над Q.

Пусть $t_1 = t_2$ – функциональное уравнение над Q, и пусть $\varphi_{i_1}^{(n_1)}, \dots, \varphi_{i_m}^{(n_m)}$ – все функциональные переменные, входящие в уравнение $t_1 = t_2$. Решением уравнения $t_1 = t_2$ называем систему функций $\{f_{\nu_1}^{(n_1)}, \dots, f_{\nu_m}^{(n_m)}\}$ из $P_{\mathbb{N}}$, которая после замены каждой функциональной переменной $\varphi_{i_s}^{(n_s)}$ соответствующей функциональной константой $f_{\nu_s}^{(n_s)}$ превращает уравнение $t_1 = t_2$ в тождество (относительно всех входящих в уравнение предметных переменных).

Пусть Ξ – конечная система уравнений над Q. Решением системы уравнений Ξ называем систему функций из $P_{\mathbb{N}}$, которая является решением каждого уравнения, входящего в Ξ. Будем говорить, что система уравнений Ξ выполнима, если система Ξ имеет решение. Проблема выполнимости для систем функциональных уравнений состоит в том, чтобы по произвольной системе функциональных уравнений определить, является ли данная система выполнимой.

В [7] установлено, что проблема выполнимости для систем функциональных уравнений, содержащих только функциональные константы $0, x+1$, принадлежит классу $\exists^1\forall^0$ аналитической иерархии Клини. Более того, эта проблема является m-полной в данном классе. Чтобы использовать указанную проблему в теореме 4, нам потребуется определить так называемое дерево решений системы функциональных уравнений над множеством функциональных констант $\{0, x+1\}$. Для произвольного множества функциональных констант это сделано в работе [6]. Мы воспроизведем из нее основные этапы построения дерева решений применительно к множеству констант $\{0, x+1\}$.

Итак, пусть Ξ – система функциональных уравнений, содержащая только функциональные константы $0, x+1$. По системе Ξ будем строить бесконечное дерево Γ (дерево решений системы уравнений Ξ), вершины которого также могут иметь бесконечную степень. В каждой вершине дерева Γ, за исключением корня, будут определяться в конечном числе точек значения всех функций, составляющих предполагаемое решение системы Ξ. В некоторых вершинах дерева Γ процесс дальнейшего построения дерева (из данной вершины) будет обрываться. Основное свойство дерева Γ состоит в том, что существует взаимно-однозначное соответствие между бесконечными ветвями дерева Γ и решениями системы уравнений Ξ.

Пусть $x_1, \dots, x_n$ – все предметные переменные системы Ξ, $\varphi_1, \dots, \varphi_m$ – все ее функциональные переменные, а $\varphi^1, \dots, \varphi^k$ – все вхождения функциональных переменных в систему Ξ (некоторые функциональные переменные могут входить в систему Ξ несколько раз). Зафиксируем эффективные нумерации множеств $\mathbb{N}^n$ и $\mathbb{N}^k$ (можно, например, перечислять элементы множества $\mathbb{N}^n$ группами с одинаковым значением суммы координат в группе и лексикографическим порядком в каждой группе). Наборы с номером i будем обозначать через $\boldsymbol{x}_i^{(n)}$ и $\boldsymbol{x}_i^{(k)}$ соответственно. Будем также предполагать, что $\boldsymbol{x}_0^{(n)}$ – нулевой набор.

Дерево Γ определяется по ярусам (вершины, расположенные на одном и том же расстоянии от корня дерева). При этом на ярусе с номером ℓ ($\ell \geqslant 1$) будут рассматриваться набор $\boldsymbol{x}_{\ell-1}^{(n)}$ (для предметных переменных $x_1, \ldots, x_n$) и наборы $\boldsymbol{x}_0^{(k)}, \boldsymbol{x}_1^{(k)}, \ldots$ (для вхождений $\varphi^1, \ldots, \varphi^k$ функциональных переменных). На ℓ-м ярусе выполняется попытка определить в конечном числе точек возможное решение $(g_1, \ldots, g_m)$ системы Ξ, отвечающее значениям $\boldsymbol{x}_{\ell-1}^{(n)}$ предметных переменных системы Ξ и "значениям" $\boldsymbol{x}_0^{(k)}, \boldsymbol{x}_1^{(k)}, \ldots$ функциональных переменных из последовательности $\varphi^1, \ldots, \varphi^k$.

Подробно опишем первый шаг в построении дерева Γ – определение вершин первого яруса. Придадим всем предметным переменным системы Ξ значения из набора $\boldsymbol{x}_0^{(n)}$ (т.е. значения 0). В вершинах $v_0, v_1, \ldots$ первого яруса присвоим вхождениям $\varphi^1, \ldots, \varphi^k$ значения, соответственно, из наборов $\boldsymbol{x}_0^{(k)}, \boldsymbol{x}_1^{(k)}, \ldots$ При этом каждый терм системы Ξ, не начинающийся символом функциональной константы, получит некоторое числовое значение. Если терму t уже присвоено значение a и в систему Ξ входит терм $t+1$, то присвоим этому терму значение $a+1$ (функциональной константе 0, естественно, присваивается значение 0). Таким образом, всем термам системы Ξ будут присвоены значения из $\mathbb{N}$.

Указанное присваивание значений термам системы Ξ может оказаться противоречивым. Так, для одной и той же переменной φ_j за счет использования различных вхождений в последовательность $\varphi^1, \ldots, \varphi^k$ одному и тому же терму вида $\varphi_j(a_1, \ldots, a_p)$ могут быть присвоены различные значения. Кроме того, могут оказаться различными и значения термов, образующих равенство системы Ξ. В этих случаях построение дерева Γ обрываем и соответствующую вершину из дерева Γ удаляем.

Предположим, что описанное выше "означивание" термов системы Ξ в вершине v_i к противоречию не приводит. Тогда, очевидно, в вершине v_i для значений предметных переменных из набора $\boldsymbol{x}_0^{(n)}$ мы имеем корректное определение (в конечном числе точек) функций $g_1, \ldots, g_m$, образующих возможное решение системы уравнений Ξ.

Продолжая далее по индукции, предположим, что уже определены ℓ ярусов дерева Γ. Выберем в ярусе ℓ некоторую вершину v_i. Предполагаем, что в вершине v_i функции $g_1, \ldots, g_m$ (из возможного решения системы уравнений Ξ) получили корректные определения в конечном числе точек, отвечающих наборам $\boldsymbol{x}_0^{(n)}, \ldots, \boldsymbol{x}_{\ell-1}^{(n)}$ для предметных переменных системы Ξ и некоторым наборам $\boldsymbol{x}_{i_1}^{(k)}, \ldots, \boldsymbol{x}_{i_\ell}^{(k)}$ для вхождений функциональных переменных в систему Ξ. Во всех вершинах $v_0, v_1, \ldots$ яруса $(\ell+1)$ дерева Γ для предметных переменных системы Ξ будет рассматриваться набор $\boldsymbol{x}_\ell^{(n)}$ и соответственно наборы $\boldsymbol{x}_0^{(k)}, \boldsymbol{x}_1^{(k)}, \ldots$ для вхождений функциональных переменных.

Определение функций $g_1, \ldots, g_m$ в вершине v_i мало отличается от аналогичного определения на первом шаге. После присваивания значений из набора $\boldsymbol{x}_\ell^{(n)}$ предметным переменным и значений из набора $\boldsymbol{x}_i^{(k)}$ вхождениям $\varphi^1, \ldots, \varphi^k$ функциональных переменных проверяем корректность такого присваивания. Единственное дополнительное требование – новые значения, полученные здесь для функций $g_1, \ldots, g_m$, необходимо проверить на совместимость со значениями этих функций, полученными на пути в вершину v_i.

Перейдем к основному свойству дерева Γ. Предположим, что в дереве Γ имеется бесконечная ветвь Δ. Проходя по всем вершинам ветви Δ, мы получим определение функций $g_1, \ldots, g_m$, которые, как нетрудно видеть, образуют решение системы Ξ. Заметим, что функции $g_1, \ldots, g_m$ на ветви Δ определяются, вообще говоря, не полностью, а лишь на некоторых подмножествах декартовых степеней множества $\mathbb{N}$. Это те минимальные (по включению) подмножества, которые необходимы для выполнения всех уравнений системы Ξ. Вместе с тем значения функций $g_1, \dots, g_m$ вне данных подмножеств никак не сказываются на выполнимости уравнений системы Ξ (эти значения не "имплицируются" системой уравнений Ξ). Поэтому решение $g_1, \dots, g_m$ можно доопределить до всюду определенного решения системы уравнений Ξ. Таким образом, любая бесконечная ветвь дерева Γ определяет решение системы уравнений Ξ, состоящее, вообще говоря, из частичных функций.

Обратно, предположим, что всюду определенные функции $g_1, \dots, g_m$ образуют решение системы уравнений Ξ. Тогда, анализируя процедуру построения дерева Γ, нетрудно убедиться, что в дереве Γ существует бесконечная ветвь. Так, если подставить в систему уравнений Ξ вместо функциональных переменных $\varphi_1, \dots, \varphi_m$ функции $g_1, \dots, g_m$, а вместо всех предметных переменных – значение 0, то в некоторой вершине v_i первого яруса дерева Γ рассматриваемый набор $\boldsymbol{x}_i^{(k)}$ будет являться набором "истинных" значений всех термов системы Ξ, которые начинаются функциональными переменными. Такое же положение возникнет на втором ярусе дерева Γ, когда будут рассматриваться вершина v_j, соединенная ребром с вершиной v_i, и наборы $\boldsymbol{x}_1^{(n)}$ для предметных переменных системы Ξ и $\boldsymbol{x}_j^{(k)}$ для вхождений ее функциональных переменных. В результате в дереве Γ будет выделена бесконечная ветвь, которая соответствует решению $g_1, \dots, g_m$.

Отметим, что дерево Γ определяется по системе уравнений Ξ эффективно. Это значит, что имеется алгоритм (он описан выше), который по координате произвольной вершины v (координата задается системой наборов $\boldsymbol{x}_{\ell-1}^{(n)}, \boldsymbol{x}_{i_1}^{(k)}, \dots, \boldsymbol{x}_{i_\ell}^{(k)}$) определяет, принадлежит ли вершина v дереву Γ, а если принадлежит, то каковы значения функций $g_1, \dots, g_m$ в этой вершине.

Теорема 4. *Отношение $U(f)$ является m-полным в классе $\exists^1\forall^0$ аналитической иерархии Клини.*

Доказательство. В теореме 3 установлена принадлежность отношения $U(f)$ классу $\exists^1\forall^0$. Поэтому для доказательства теоремы достаточно показать, что любое множество из класса $\exists^1\forall^0$ m-сводится к множеству $U(f)$. То, что всякое множество из класса $\exists^1\forall^0$ можно представить в виде множества решений подходящей системы функциональных уравнений над множеством функциональных констант $\{0, x+1\}$, установлено с использованием формализма Эрбрана – Гёделя в работе [7] и с использованием формализма Клини в работе [5]. Следовательно, остается показать, как по произвольной системе функциональных уравнений Ξ над множеством $\{0, x+1\}$ эффективно определить характеристическую функцию f кода $\mathcal{C}$, чтобы система Ξ имела решение тогда и только тогда, когда код $\mathcal{C}$ не является однозначным.

Воспользуемся построенным выше деревом Γ решений системы Ξ. Итак, предполагаем, что каждой вершине v яруса ℓ дерева Γ приписан набор $c(v)$, который состоит из набора $\boldsymbol{x}_{\ell-1}^{(n)}$ значений переменных $x_1, \dots, x_n$, а также всех наборов, задающих части графиков функций $g_1, \dots, g_m$ и определенных на пути от корня дерева Γ к вершине v. Набор из графика функции g_i, отвечающий аргументам $a_1, \dots, a_t$, записываем в виде $(a_1, \dots, a_t, g_i(a_1, \dots, a_t))$. Порядок наборов в $c(v)$ следующий: сначала набор $\boldsymbol{x}_{\ell-1}^{(n)}$, затем в порядке возрастания индекса i наборы из графика функции $g_i(y_1, \dots, y_t)$, при этом для фиксированного i наборы из графика функции g_i упорядочены в соответствии с нумерацией множества $\mathbb{N}^{(t)}$.

В дальнейшем набор из $c(v)$ будем записывать в виде слова в алфавите $\{0, 1, 2, 3, 4\}$, а именно: каждое число из $\mathbb{N}$ представляется в стандартной двоичной записи; двоичные записи чисел из $\boldsymbol{x}_{\ell-1}^{(n)}$ отделяются символом 2, то же самое для двоичных записей аргументов и значений функций в графиках функций g_i; символ 3 отделяет "блок" $\boldsymbol{x}_{\ell-1}^{(n)}$ от графика функции g_1, а также от графиков соседних функций g_i; символ 4 стоит после графика функции g_m.

По дереву Γ (с пометками в вершинах) определим код $C(\Gamma)$. Прежде всего, для любой вершины v первого яруса внесем в код $C(\Gamma)$ слово $c(v)$. Далее, для любой (конечной или бесконечной) ветви $v_1, v_2, \ldots$ дерева Γ внесем в код $C(\Gamma)$ слова

$$c(v_1)c(v_2),\ c(v_3)c(v_4),\ \ldots,\ c(v_{2j-1})c(v_{2j}),\ \ldots,$$

которые будем называть нечетными парами (если ветвь содержит нечетное число вершин, то заканчиваем последней нечетной парой $c(v_{2j-1})c(v_{2j})$). Аналогично, вносим в код $C(\Gamma)$ слова

$$c(v_2)c(v_3),\ c(v_4)c(v_5),\ \ldots,\ c(v_{2j})c(v_{2j+1}),\ \ldots,$$

которые называем четными парами (если ветвь содержит четное число вершин, то, как и выше, заканчиваем последней четной парой $c(v_{2j})c(v_{2j+1})$).

Очевидно, что для любой бесконечной ветви $v_1, v_2, \ldots$ сверхслово

$$c(v_1)c(v_2)c(v_3)c(v_4)\ldots c(v_{2j-1})c(v_{2j})\ldots$$

допускает два декодирования: одно – последовательными нечетными парами, а другое – словом $c(v_1)$ и последующими четными парами.

Докажем теперь, что если имеется сверхслово W, которое допускает (в коде $C(\Gamma)$) два различных декодирования, то в дереве Γ существует бесконечная ветвь.

Вообще говоря, указанные декодирования могут давать один и тот же результат на некотором непустом начале сверхслова W. Мы хотим исключить этот вариант из рассмотрения (на дальнейшем построении бесконечной ветви в дереве Γ данное преобразование не скажется). Поэтому будем предполагать, что два упомянутых декодирования различаются уже по первым кодовым словам.

Заметим теперь, что сверхслово W не может начинаться словом вида $c(v)c(v)$, где v – вершина первого яруса дерева Γ. В самом деле, поскольку нечетные и четные пары из $C(\Gamma)$ имеют вид $c(v_t)c(v_{t+1})$, где v_t, v_{t+1} – различные вершины дерева Γ, два различных декодирования сверхслова W должны начинаться одним и тем же кодовым словом $c(v)$. Однако эту возможность мы исключили.

Далее замечаем, что началом сверхслова не может являться слово $c(v_t)$, где $t \geqslant 2$. Действительно, если $t \geqslant 2$, то в случае нечетного t любая из расшифровок должна начинаться с одной и той же нечетной пары $c(v_t)c(v_{t+1})$, что мы исключили. Аналогичные рассуждения применимы в случае четного t.

Итак, сверхслово W начинается словом $c(v_1)$, за которым следует слово $c(v_2)$, причем v_1, v_2 – вершины первого и второго ярусов дерева Γ. Следовательно, в одном из декодирований первым кодовым словом должна быть нечетная пара $c(v_1)c(v_2)$. В силу сделанного выше предположения относительно сверхслова W второе декодирование не может начинаться с этой нечетной пары. Значит, остается одна возможность: второе декодирование начинается с кодового слова $c(v_1)$. Тогда вторым кодовым словом в этом декодировании должна быть, конечно, четная пара вида $c(v_2)c(v_3)$. Понятно, что вторым кодовым словом в первом декодировании будет нечетная пара вида $c(v_3)c(v_4)$, третьим кодовым словом во втором декодировании – слово вида $c(v_4)c(v_5)$ и т.д. ▲

Автор выражает признательность рецензенту, указавшему на пробел в доказательстве.

СПИСОК ЛИТЕРАТУРЫ

1. *Марков Ал.А.* Об алфавитном кодировании // ДАН СССР. 1960. Т. 132. № 3. С. 521–523.
2. *Марков Ал.А.* Нерекуррентное кодирование // Проблемы кибернетики. Вып. 8. М.: Физматгиз, 1962. С. 169–186.

3. *Марков А.А.* Введение в теорию кодирования. М.: Наука, 1982.
4. *Роджерс Х.* Теория рекурсивных функций и эффективная вычислимость. М.: Мир, 1972.
5. *Марченков С.С.* Определимость в языке функциональных уравнений счетнозначной логики // Дискрет. матем. 2013. Т. 25. № 4. С. 13–23.
6. *Марченков С.С.* О сложности решения систем функциональных уравнений счетнозначной логики // Дискретный анализ и исслед. опер. 2015. Т. 22. № 2. С. 49–62.
7. *Марченков С.С., Калинина И.С.* Оператор FE-замыкания в счетнозначной логике // Вестн. Моск. ун-та. Сер. 15. Вычисл. матем. киберн. 2013. № 3. С. 42–47.

Марченков Сергей Серафимович
Московский государственный университет им. М.В.Ломоносова,
факультет вычислительной математики и кибернетики
ssmarchen@yandex.ru

Поступила в редакцию
09.01.2019
После доработки
29.04.2019
Принята к публикации
14.05.2019