

ОРИГИНАЛЬНАЯ СТАТЬЯ

УДК 519.714.1

doi: 10.26907/2541-7746.2021.3-4.276-290

СИНТЕЗ БИНАРНЫХ ПРОГРАММ С ПРЕОБЛАДАНИЕМ КОМАНД ПЕРЕАДРЕСУЮЩЕГО ТИПА

В.В. Жуков

*Московский государственный университет имени М.В. Ломоносова,
г. Москва, 119991, Россия*

Аннотация

В работе рассмотрена модель бинарных программ, реализующих функции алгебры логики (булевы функции), состоящих из одного или нескольких модулей, которые содержат команды трех типов: вычислительные, переадресующие и команды вызова процедур. В данной модели также допускается рекурсивный вызов процедур, то есть в процессе выполнения бинарной программы процедуры могут непосредственно или через другие процедуры вызывать сами себя. Введено понятие произвольного базиса для команд переадресующего типа в качестве обобщения известных моделей. Представлены методы получения нижних и верхних оценок функции Шеннона для сложности реализации булевых функций в классе бинарных программ. Предложенные методы позволили установить асимптотику функции Шеннона в случае, когда удельный вес переадресующих команд меньше удельного веса команд вычислительного типа.

Ключевые слова: бинарные программы, функция Шеннона, асимптотические оценки

Введение

Задача синтеза, которая впервые была рассмотрена К. Шенноном [1], состоит в поиске наиболее оптимальных методов построения дискретных управляющих систем для произвольной булевой функции или систем таких функций. Для оценки оптимальности метода синтеза вводится функция Шеннона, которая при заданном значении n равна сложности самой сложной функции, зависящей от n переменных. При этом сложностью функции называют наименьшую сложность управляющей системы, реализующей данную функцию. Под сложностью управляющей системы чаще всего понимают количество элементов в ней или их суммарный вес. Точно таким же образом рассматривают методы синтеза, направленные на оптимизацию схем по задержке.

Рассмотрим модель бинарных программ, расширяющую модели, введенные в работах [2, 3]. Подробный обзор существующих результатов в области синтеза в классах программ и некоторых их подклассах можно найти в работе [3].

Возьмем счетное множество булевых переменных (БП) $X = \{x_1, \dots, x_n, \dots\}$, которые будут являться аргументами рассматриваемых далее булевых функций. Каждая булева переменная принимает значения из множества $B = \{0, 1\}$. Булевым кубом размерности n , $n \in \mathbb{N}$, где \mathbb{N} – множество натуральных чисел, называется n -я декартова степень множества B , которая обозначается B^n , а функцией алгебры логики (ФАЛ) или булевой функцией $f(x_1, \dots, x_n)$ – отображение булева

куба B^n на множество B . Отображение $\vec{f}(x_1, \dots, x_n)$ множества B^n на множество B^m для некоторых натуральных n и m будем называть (n, m) -вектор-функцией. Множество всех булевых функций, зависящих от n переменных, обозначим через $P_2(n)$, а множество (n, m) -вектор-функций – через $P_2^m(n)$. Обозначим также множество всех булевых функций как

$$P_2 = \bigcup_{n \in \mathbb{N}} P_2(n),$$

а множество всех булевых (n, m) -вектор-функций – как

$$P_2^m = \bigcup_{n \in \mathbb{N}} P_2^m(n).$$

Введем конечные базисы $\check{B} = \{\check{\varphi}_1, \dots, \check{\varphi}_{\check{b}}\}$ и $\hat{B} = \{\hat{\varphi}_1, \dots, \hat{\varphi}_{\hat{b}}\}$, состоящие из \check{b} булевых функций $\check{\varphi}_1, \dots, \check{\varphi}_{\check{b}}$ и \hat{b} булевых функций $\hat{\varphi}_1, \dots, \hat{\varphi}_{\hat{b}}$ соответственно, в которых каждая из функций $\check{\varphi}_i, i = 1, 2, \dots, \check{b}$, существенно зависит от $\check{k}_i, \check{k}_i \geq 0$, переменных, а каждая из функций $\hat{\varphi}_j, j = 1, 2, \dots, \hat{b}$, – от $\hat{k}_j, \hat{k}_j \geq 0$, булевых переменных. Припишем также каждой функции $\check{\varphi}_i, i = 1, 2, \dots, \check{b}$, и $\hat{\varphi}_j, j = 1, 2, \dots, \hat{b}$, некоторые положительные вещественные веса \check{L}_i и \hat{L}_j соответственно.

Стандартным базисом для переадресующих команд \hat{B}_0 назовем базис, состоящий из единственной тождественной функции $\varphi(x) = x$ и имеющей вес, равный $2\lambda, \lambda \geq 0$. Стандартным базисом для вычислительных команд принято считать базис $\check{B}_0 = \{\&, \vee, \neg\}$, состоящий из конъюнкции, дизъюнкции и отрицания.

Бинарной программой Σ назовем набор подпрограмм $\{\Sigma_1, \dots, \Sigma_s\}$, для каждой из которых задан набор из $\text{in}(i), i = 1, 2, \dots, s$, входных аргументов, $\text{out}(i), i = 1, 2, \dots, s$, выходных аргументов, упорядоченный набор команд $\Gamma_i = \{\Gamma_{i,1}, \dots, \Gamma_{i,c_i}\}$, а также конечный размер области памяти $M_i \geq \text{in}(i) + \text{out}(i)$, используемой данной подпрограммой. Каждая ячейка памяти подпрограмм может содержать значения из множества B . Без ограничения общности считаем, что значения входных аргументов при вызове подпрограммы сохраняются в первые $\text{in}(i)$ ячеек памяти, следующие $\text{out}(i)$ ячеек памяти предназначены для записи выходных значений подпрограммы, а все остальные ячейки могут быть использованы для хранения результатов промежуточных вычислений.

Команды подпрограмм, входящие в наборы $\Gamma_i, i = 1, 2, \dots, s$, могут быть трех типов.

1. Вычислительные команды $\{\check{\varphi}_j; m_1, \dots, m_{\check{k}_j}; m_{\text{out}}\}$ описываются символом булевой функции $\check{\varphi}_j(x_1, \dots, x_{\check{k}_j})$ из заданного базиса \check{B} , номерами входных ячеек памяти $m_1, \dots, m_{\check{k}_j}$ и номером выходной ячейки памяти $m_{\text{out}} > \text{in}(i)$. Функционирует вычислительная команда следующим образом: считываются значения из ячеек памяти с номерами $m_1, \dots, m_{\check{k}_j}$, вычисляется функция $\check{\varphi}_j$ с данными значениями аргументов, записывается результат в ячейку памяти с номером m_{out} . Вес вычислительной команды равен весу \check{L}_j , соответствующему булевой функции $\check{\varphi}_j$.

2. Переадресующие команды $\{\hat{\varphi}_j; m_1, \dots, m_{\hat{k}_j}; c_{\text{false}}, c_{\text{true}}\}$ описываются символом булевой функции $\hat{\varphi}_j(x_1, \dots, x_{\hat{k}_j})$ из базиса \hat{B} , номерами ячеек памяти $m_1, \dots, m_{\hat{k}_j}$ и номерами команд текущей подпрограммы c_{false} и c_{true} . Переадресующая команда позволяет совершать условный переход на команды с номерами c_{false} и c_{true} . Если значение функции $\hat{\varphi}_j$ на наборе, записанном в ячейки памяти $m_1, \dots, m_{\hat{k}_j}$, равно 0, то после выполнения переадресующей команды выполнение подпрограммы продолжается с команды с номером c_{false} . Если же значение равно 1, то выполнение продолжается с команды с номером c_{true} . Значения c_{false}

и c_{true} также могут быть на единицу больше, чем максимальный номер команды в текущей подпрограмме. В этом случае выполняется переход в конец подпрограммы, что завершает ее выполнение. Вес переадресующей команды будем считать равным \tilde{L}_j .

3. Команды вызова подпрограмм $\{p; m_1, \dots, m_{\text{in}(p)}; m'_1, \dots, m'_{\text{out}(p)}\}$ описываются номером вызываемой подпрограммы p , номерами ячеек памяти текущей подпрограммы $m_1, \dots, m_{\text{in}(p)}$, значения которых будут использованы как входные аргументы вызываемой подпрограммы, а также номерами ячеек памяти текущей подпрограммы $m'_1, \dots, m'_{\text{out}(p)}$, куда будут записаны результаты выполнения подпрограммы Σ_p . Следуя [2], будем считать, что вес команды вызова подпрограммы равен $\mu \text{in}(p) + \nu$, где μ и ν – некоторые неотрицательные вещественные константы. В настоящей работе будем рассматривать случай, когда $\mu = 0$ и $\nu > 0$, то есть вес команды вызова подпрограммы всегда положительный и не зависит от количества аргументов вызываемой подпрограммы.

Вычисление бинарной программы Σ происходит следующим образом. Выполнение начинается с первой подпрограммы Σ_1 с некоторыми заданными значениями входных аргументов x_1, \dots, x_n , $n \leq \text{in}(1)$. При выполнении команды вызова как первой подпрограммы, так и других подпрограмм для них выделяется область памяти требуемого размера, копируются значения входных аргументов в первые ячейки памяти, все остальные ячейки памяти, кроме выходных, инициализируются нулями. Выполнение подпрограммы начинается с самой первой команды соответствующей подпрограммы, то есть с команды с номером 1, и заканчивается при достижении конца подпрограммы. Результатом работы бинарной программы Σ является набор значений, записанных в $\text{out}(1)$ выходных ячеек памяти самой первой вызванной подпрограммы.

Будем говорить, что бинарная программа Σ вычисляет булеву функцию $f(x_1, \dots, x_n) \in P_2(n)$, если вычисление Σ для всех 2^n наборов входных аргументов x_1, \dots, x_n завершается и результатом является значение функции $f(x_1, \dots, x_n)$. Аналогично можно определить вычислимость вектор-функции $\vec{f}(x_1, \dots, x_n) \in P_2^m(n)$ с помощью заданной бинарной программы Σ , отличие лишь в том, что подпрограмма Σ_1 должна иметь m выходных ячеек памяти, в которые записывается результат вычисления вектор-функции.

Заметим, что подпрограмма Σ_1 при реализации функций или вектор-функций, кроме n входных и 1 или m выходных ячеек, может иметь вспомогательные входные и выходные ячейки памяти, которые полезны при использовании рекурсии, в частности для реализации счетчика глубины рекурсии или выбора режима функционирования программы. При этом, как было сказано выше, все входные ячейки, для которых не заданы входные значения, инициализируются нулями.

В описанной выше модели бинарных программ не задается ограничение на максимально возможную глубину вызовов подпрограмм. Такие программы будем называть бинарными программами с неограниченной глубиной рекурсии. Но в настоящей работе будем рассматривать бинарные программы с ограниченной глубиной рекурсии $r \geq 1$. Такие программы функционируют аналогичным образом, но при превышении глубины рекурсии, то есть при попытке выполнения команды вызова подпрограммы с уровня r вложенности вызовов подпрограмм вызов этой команды не происходит, а выполнение текущей подпрограммы продолжается со следующей команды.

Сложностью $\mathcal{L}(\Sigma)$ бинарной программы Σ называется сумма весов всех команд ее подпрограмм. Обозначим класс бинарных программ в паре базисов \check{B}, \hat{B} с пара-

метрами $\mu = 0, \nu$ и ограниченной глубиной рекурсии r как¹ $\mathcal{U}_{\check{B}, \hat{B}, \nu}^{\Pi(r)}$. Сложностью булевой функции $f(x_1, \dots, x_n) \in P_2(n)$ в классе бинарных программ с ограниченной глубиной рекурсии r назовем минимальную сложность программы, реализующей функцию $f(x_1, \dots, x_n)$:

$$\mathcal{L}^{\Pi(r)}(f) = \min_{\substack{\Sigma \in \mathcal{U}^{\Pi(r)} \\ \Sigma \text{ реал. } f}} \mathcal{L}(\Sigma),$$

а функцией Шеннона для сложности реализации булевых функций в классе бинарных программ – максимальную сложность функции, зависящей от n переменных

$$\mathcal{L}^{\Pi(r)}(n) = \max_{f \in P_2(n)} \mathcal{L}^{\Pi(r)}(f).$$

1. Постановка задачи

Введем следующие величины:

$$\pi_{\check{B}} = \min_{i=1,2,\dots,\check{b}} \frac{\check{L}_i}{\check{k}_i + 1}, \quad \pi_{\hat{B}} = \min_{\substack{i=1,2,\dots,\hat{b}, \\ \hat{k}_i > 0}} \frac{\hat{L}_i}{\hat{k}_i + 1}, \quad \theta_{\hat{B}} = \min_{\substack{i=1,2,\dots,\hat{b}, \\ \hat{k}_i > 0}} \frac{\hat{L}_i}{\hat{k}_i}.$$

В работе [2] было получено следующее утверждение для класса одномодульных бинарных программ, то есть состоящих из одной подпрограммы Σ_1 и не содержащих команд вызова подпрограмм.

Теорема 1. Для функции Шеннона в классе одномодульных бинарных программ $\mathcal{U}_{\check{B}, \hat{B}_0, \nu}^{\Pi(1)}$ справедливо асимптотическое равенство²

$$\mathcal{L}_{\check{B}, \hat{B}_0, \nu}^{\Pi(1)}(n) \sim \min(\pi_{\check{B}}, \lambda) \frac{2^n}{n}.$$

В работе [3] устанавливается следующее асимптотическое равенство для функции Шеннона $\mathcal{L}_{\check{B}, \hat{B}_0, \nu}^{\Pi(r)}(n)$.

Теорема 2. В случае $r \geq 2$ и $\pi_{\check{B}} \leq \lambda$ справедливо

$$\mathcal{L}_{\check{B}, \hat{B}_0, \nu}^{\Pi(r)}(n) \sim r\nu^{(r-1)/r} (r-1)^{-(r-1)/r} \sqrt[r]{\pi_{\check{B}}} \frac{2^{n/r}}{\sqrt[r]{n}}.$$

В статье [4] представлен следующий результат, который обобщается и обосновывается в настоящей работе.

Теорема 3. В случае $r \geq 2$ и $2\lambda \leq \pi_{\check{B}}$ справедливо асимптотическое равенство

$$\mathcal{L}_{\check{B}, \hat{B}_0, \nu}^{\Pi(r)}(n) \sim r\nu^{(r-1)/r} (r-1)^{-(r-1)/r} \sqrt[2r]{2\lambda} \frac{2^{n/r}}{\sqrt[r]{n}}.$$

Нашей задачей является получение асимптотики функции Шеннона для сложности реализации булевых функций в классе одномодульных бинарных программ $\mathcal{U}^{\Pi(1)}(n)$ при $\pi_{\hat{B}} < \pi_{\check{B}}$ и в классе многомодульных бинарных программ $\mathcal{U}^{\Pi(r)}(n)$, $r \geq 2$, в случае $\theta_{\hat{B}} < \pi_{\check{B}}$. Основным результатом работы является следующее утверждение.

¹ Нижние индексы \check{B}, \hat{B}, ν будем, как правило, полностью или частично опускать.

² Асимптотическое неравенство $a(n) \lesssim b(n)$, где $n = 1, 2, \dots$, означает, что $a(n) \leq b(n)(1 + \bar{o}(1))$, а асимптотическое равенство $a(n) \sim b(n)$ равносильно неравенствам $a(n) \lesssim b(n)$ и $b(n) \lesssim a(n)$.

Теорема 4. Для функции Шеннона $\mathcal{L}^{\Pi(r)}(n)$ при $r = 1$ и $\pi_{\check{B}} < \pi_{\hat{B}}$ верно асимптотическое равенство

$$\mathcal{L}^{\Pi(1)}(n) \sim \pi_{\check{B}} \frac{2^n}{n},$$

а при $r \geq 2$ и $\theta_{\check{B}} < \pi_{\hat{B}}$ верно

$$\mathcal{L}^{\Pi(r)}(n) \sim r\nu^{(r-1)/r}(r-1)^{-(r-1)/r} \sqrt[r]{\theta_{\check{B}}} \frac{2^{n/r}}{\sqrt[r]{n}}.$$

2. Нижняя оценка функции Шеннона для одномодульных программ

Для получения нижних оценок функции Шеннона мощностным методом в заданном классе схем \mathcal{U} сначала оценивается сверху число $\|\mathcal{U}(\mathcal{L}, n)\|$ попарно неэквивалентных одновыходных схем из данного класса, имеющих n входных переменных и сложность не более \mathcal{L} . Затем к данной верхней оценке применяется стандартное мощностное преобразование [5], результатом которого является искомая нижняя оценка.

Сначала получим оценку числа попарно неэквивалентных одномодульных бинарных программ, имеющих n входных аргументов, единственный выходной аргумент и сложность не более \mathcal{L} . Оценим число вариантов присвоения типов команд и символов булевых функций из базисов \check{B} , \hat{B} . Так как количество функций в базисах \check{B} и \hat{B} равно \check{b} и \hat{b} соответственно, для каждой команды есть $\check{b} + \hat{b}$ вариантов присвоения. Число команд в одномодульной программе равно $O(\mathcal{L})$, поэтому общее число вариантов присвоения типов команд и функций из базисов не превосходит

$$C_1 \mathcal{L}, \quad (1)$$

где C_1 – некоторая константа³.

Обозначим сложность всех вычислительных команд программы как $\check{\mathcal{L}}$, а всех переадресующих команд – $\hat{\mathcal{L}}$. Также обозначим через \hat{N} количество переадресующих команд, а через \hat{K} – общее количество их аргументов, соответствующих ячейкам памяти. Число ячеек памяти M , требующихся для выполнения программы, не превосходит $C_2 \check{\mathcal{L}} + n + 1$, так как команды используют либо ячейки памяти, соответствующие входным и выходным аргументам, либо ячейки памяти, в которые производят запись вычислительные команды.

Оценим сверху количество аргументов вычислительных команд \check{K} , соответствующих входным и выходным ячейкам памяти. Пусть для каждой булевой функции $\check{\varphi}_i \in \check{B}$, $i \in 1, 2, \dots, \check{b}$, суммарная сложность команд, использующих данную функцию, равна $\check{\mathcal{L}}_i$, тогда

$$\check{K} = \check{\mathcal{L}}_1 \frac{(\check{k}_1 + 1)}{\check{L}_1} + \dots + \check{\mathcal{L}}_{\check{b}} \frac{(\check{k}_{\check{b}} + 1)}{\check{L}_{\check{b}}} \leq (\check{\mathcal{L}}_1 + \dots + \check{\mathcal{L}}_{\check{b}}) \cdot \max_{i=1,2,\dots,\check{b}} \frac{\check{k}_i + 1}{\check{L}_i} \leq \frac{\check{\mathcal{L}}}{\pi_{\check{B}}}.$$

Учитывая верхние оценки для M и \check{K} , получаем, что число вариантов выбора номеров ячеек памяти для вычислительных команд не превосходит

$$(C_2 \check{\mathcal{L}} + n + 1)^{\check{\mathcal{L}}/\pi_{\check{B}}}, \quad (2)$$

³ Буквой C с различными индексами будем обозначать константы, зависящие только от базисов \check{B} , \hat{B} .

а для переадресующих команд не превосходит

$$(C_2\check{\mathcal{L}} + n + 1)^{\widehat{K}}. \tag{3}$$

Оценку (2) можно уточнить. Для каждой ячейки памяти, встречающейся в рассматриваемой одномодульной программе, за исключением ячеек, соответствующих входным и выходным аргументам программы, определим первую вычислительную команду, производящую запись в данную ячейку. Обозначим множество этих вычислительных команд как Γ_M , а их суммарную сложность – как $\check{\mathcal{L}}_M$. Поскольку при перенумерации ячеек памяти, не соответствующих входным и выходным аргументам программы, функционирование программы не изменяется, будем считать, что порядок вычислительных команд Γ_M совпадает с порядком номеров соответствующих им ячеек. Перемножая число вариантов выбора вычислительных команд, принадлежащих множеству Γ_M , а также число вариантов выбора номеров выходных ячеек для вычислительных команд, не содержащихся в Γ_M , и входных ячеек памяти для всех вычислительных команд, получим оценку

$$(C_3\check{\mathcal{L}}_M + n + 1)^{\check{\mathcal{L}}/\pi_{\check{B}} - \check{\mathcal{L}}_M/(\pi_{\check{B}}(k_{\check{B}}+1))}, \tag{4}$$

где $k_{\check{B}}$ – максимальное число аргументов у булевых функций из базиса \check{B} .

Так как максимум по $\check{\mathcal{L}}_M$, $0 < \check{\mathcal{L}}_M \leq \check{\mathcal{L}}$, функции, задаваемой формулой (4), достигается при $\check{\mathcal{L}}_M = C_4\check{\mathcal{L}}/\log \check{\mathcal{L}}$, в результате получаем следующую верхнюю оценку числа вариантов выбора номеров ячеек памяти вычислительных команд:

$$\left(C_5 \left(\frac{\check{\mathcal{L}}}{\log \check{\mathcal{L}}} + n + 1 \right) \right)^{\check{\mathcal{L}}/\pi_{\check{B}}}. \tag{5}$$

Осталось оценить число вариантов выбора аргументов переадресующих команд, соответствующих номерам команд, на которые осуществляется условный переход. Назовем блоком максимальную последовательность команд, заканчивающуюся переадресующей командой, все остальные команды которой являются вычислительными. Блок может в том числе состоять только из одной переадресующей команды. Вся последовательность команд бинарной программы разбивается переадресующими командами не более чем на \widehat{N} блоков. Последние команды программы, вообще говоря, могут не принадлежать ни одному из блоков, если за ними не следует переадресующих команд.

Без ограничения общности будем считать, что все команды программы являются достижимыми, то есть для каждой команды существует такой набор входных аргументов, при котором она будет выполнена. Построим дерево достижимости блоков, корнем которого является вершина, соответствующая первому блоку программы, а ребра соответствуют переходам между блоками. Число таких корневых деревьев не превосходит

$$C_6^{\widehat{N}-1}. \tag{6}$$

Заметим, что блоки программы, за исключением первого, можно переставлять местами, изменяя соответствующим образом номера команд в переадресующих командах. Поэтому можно считать, что блоки в программе расположены в порядке, соответствующем некоторому обходу дерева достижимости (например, в глубину или ширину).

Ребра дерева достижимости определяют $\widehat{N} - 1$ из $2\widehat{N}$ аргументов переадресующих команд с точностью до блоков. Для них требуется доопределить, на какую

именно команду блока осуществляется переход, а также какому из двух аргументов переадресующей команды соответствует данное ребро. Число вариантов этого доопределения не превосходит

$$(C_7(1 + \check{\mathcal{L}}/\hat{\mathcal{L}}))^{\hat{N}-1}. \quad (7)$$

У каждого из оставшихся $\hat{N} + 1$ аргументов переадресующих команд число вариантов определения номера команды для перехода равно $O(\mathcal{L})$, то есть общее число вариантов не превосходит

$$(C_8\mathcal{L})^{\hat{N}+1}. \quad (8)$$

Перемножая оценки (1), (3), (5), (6), (7), (8) и оценивая их произведение сверху, получим

$$C_9\mathcal{L} \left(C_5 \left(\frac{\check{\mathcal{L}}}{\log \check{\mathcal{L}}} + n + 1 \right) \right)^{\check{\mathcal{L}}/\pi_{\check{B}}} (C_2\check{\mathcal{L}} + n + 1)^{\hat{K}} (C_{10}\mathcal{L}(1 + \check{\mathcal{L}}/\hat{\mathcal{L}}))^{\hat{N}+1}. \quad (9)$$

Будем считать, что $n = o(\mathcal{L})$. Максимум формулы (9) в случае $\pi_{\check{B}} < \pi_{\hat{B}}$ достигается при $\check{\mathcal{L}} = C_{11}\mathcal{L}/\log \mathcal{L}$. В результате получаем, что число попарно неэквивалентных одномодульных бинарных программ сложности не более \mathcal{L} с n входными и одним выходным аргументом не превосходит

$$(C_{12}\mathcal{L})^{\mathcal{L}/\pi_{\hat{B}}+1}. \quad (10)$$

Применяя стандартное мощностное преобразование к (10), получим следующее утверждение.

Лемма 1. Для функции Шеннона $\mathcal{L}^{\Pi(1)}(n)$ при $\pi_{\check{B}} < \pi_{\hat{B}}$ верно асимптотическое неравенство

$$\mathcal{L}^{\Pi(1)}(n) \gtrsim \pi_{\hat{B}} \frac{2^n}{n}.$$

3. Нижняя оценка функции Шеннона для многомодульных программ

Перейдем к получению нижних оценок функции Шеннона для класса многомодульных бинарных программ с ограниченной глубиной рекурсии r . Оценим число попарно неэквивалентных бинарных программ Σ из данного класса, сложность которых не превышает \mathcal{L} . По сравнению с одномодульными программами, в программе Σ для каждой команды количество вариантов присвоения ей типа и функционального символа увеличилось на единицу за счет наличия команд вызова подпрограмм и стало равно $\check{b} + \hat{b} + 1$. Поэтому верхняя оценка общего числа вариантов присвоения типов команд и функций из базисов \check{B} , \hat{B} командам программы Σ имеет вид

$$C_{13}\mathcal{L}. \quad (11)$$

Далее оценим сверху число вариантов присвоений адресных аргументов переадресующим командам Σ . Поскольку каждая из \hat{N} переадресующих команд имеет два адресных аргумента, принимающих $O(\mathcal{L})$ различных значений, количество таких присвоений можно оценить сверху как

$$(C_{14}\mathcal{L})^{2\hat{N}}. \quad (12)$$

Осталось оценить число вариантов разбиения команд на подпрограммы, а также выбора количества входных и выходных ячеек памяти данных подпрограмм; способов определения номеров вызываемых подпрограмм для команд вызова подпрограмм; вариантов присвоения значений, которые связаны с номерами ячеек памяти, для всех типов команд.

Вместо непосредственной оценки указанных величин воспользуемся приемом раскрытия рекурсии, идея которого заключается в переходе от многомодульной бинарной программы к эквивалентной одномодульной программе, не содержащей команд вызова подпрограмм.

Опишем применение одного шага раскрытия рекурсии к бинарной программе Σ . Данной процедуре подвергается только подпрограмма Σ_1 бинарной программы Σ и при этом происходит следующее.

1. Заменяются все команды вызова подпрограмм на реализации соответствующих им вызываемых подпрограмм. Для корректности подстановки для подпрограммы Σ_1 заводятся новые ячейки памяти, которые ранее не использовались в программе, и выполняется взаимно однозначная замена всех вхождений номеров ячеек памяти, используемых добавленными командами, за исключением входных и выходных ячеек памяти вызываемых подпрограмм.

2. Номера входных и выходных ячеек памяти в командах вызываемых подпрограмм заменяются в соответствии с аргументами команд вызова подпрограмм.

3. Происходит коррекция номеров команд в аргументах переадресующих команд с целью сохранения эквивалентности.

Для перехода от многомодульной бинарной программы Σ с ограниченной глубиной рекурсии r , $r \geq 2$, к одномодульной программе Σ^* последовательно выполним $r - 1$ шагов раскрытия рекурсии, удалим подпрограммы $\Sigma_2, \dots, \Sigma_s$, а также все команды вызова подпрограмм из подпрограммы Σ_1 с коррекцией аргументов переадресующих команд. Легко видеть, что полученная бинарная программа Σ^* эквивалентна исходной программе Σ .

Оценим сложность программы Σ^* , получившейся после применения операции раскрытия рекурсии. Обозначим через \mathcal{L}' суммарную сложность вычислительных и переадресующих команд исходной бинарной программы Σ , а через \mathcal{L}'' – суммарную сложность всех команд вызова подпрограмм. Рассмотрим сначала один шаг раскрытия рекурсии. Максимальная сложность команд вызова подпрограмм в Σ_1 может достигать \mathcal{L}'' , если команды данного типа содержатся только в подпрограмме Σ_1 . Так как вес каждой команды вызова подпрограммы равен ν , в результате одного шага раскрытия рекурсии происходит замена до \mathcal{L}''/ν команд на реализации соответствующих им подпрограмм, каждая из которых содержит вычислительные и переадресующие команды с суммарным весом до \mathcal{L}' и команды вызова подпрограмм с суммарным весом до \mathcal{L}'' . Сложность программы после данной операции можно оценить сверху величиной

$$\mathcal{L}' + \frac{\mathcal{L}''}{\nu} (\mathcal{L}' + \mathcal{L}''),$$

при этом максимальный суммарный вес вычислительных и переадресующих команд получившейся программы равен $\mathcal{L}' + (\mathcal{L}''/\nu)\mathcal{L}'$, а команд вызова подпрограмм – $(\mathcal{L}''/\nu)\mathcal{L}''$. Проводя аналогичные рассуждения, получаем, что после $r - 1$ шагов применения операции раскрытия рекурсии к программе Σ с последующим удалением всех команд вызова подпрограмм справедлива следующая оценка:

$$\mathcal{L}(\Sigma^*) \leq \mathcal{L}' + \frac{\mathcal{L}''}{\nu} \mathcal{L}' + \dots + \left(\frac{\mathcal{L}''}{\nu}\right)^{r-1} \mathcal{L}' = \left(\frac{\mathcal{L}''}{\nu}\right)^{r-1} \mathcal{L}' + O(\mathcal{L}^{r-1}).$$

При условии $\mathcal{L}' + \mathcal{L}'' \leq \mathcal{L}$ слагаемое $(\mathcal{L}''/\nu)^{r-1} \mathcal{L}'$ достигает своего максимума при

$$\mathcal{L}' = \frac{\mathcal{L}}{r} \quad \text{и} \quad \mathcal{L}'' = (r-1) \frac{\mathcal{L}}{r},$$

откуда получаем следующее неравенство:

$$\mathcal{L}(\Sigma^*) \leq \frac{(r-1)^{r-1}}{r^r \nu^{r-1}} \mathcal{L}(\Sigma)^r + O(\mathcal{L}(\Sigma)^{r-1}) \leq \frac{(r-1)^{r-1}}{r^r \nu^{r-1}} (\mathcal{L}(\Sigma) + C_{15})^r. \quad (13)$$

Вернемся к оценке числа попарно неэквивалентных бинарных программ. С учетом того что попарно неэквивалентным многомодульным программам соответствуют попарно неэквивалентные одномодульные программы, получаемые в результате раскрытия рекурсии, достаточно оценить количество вариантов определения номеров ячеек памяти для аргументов вычислительных и переадресующих команд программы Σ^* .

Обозначим суммарную сложность всех вычислительных команд одномодульной программы Σ^* как $\check{\mathcal{L}}^*$, а всех переадресующих команд как $\hat{\mathcal{L}}^*$. Общее число ячеек памяти M^* , используемых в программе Σ^* , не превосходит $C_{16} \check{\mathcal{L}}^* + n + 1$, поэтому число вариантов выбора ячеек памяти для аргументов вычислительных и переадресующих команд не превосходит

$$(C_{16} \check{\mathcal{L}}^* + n + 1)^{\check{\mathcal{L}}^*/\pi_{\check{B}} + \hat{\mathcal{L}}^*/\theta_{\check{B}}}. \quad (14)$$

Будем считать, что $n = o(\mathcal{L}(\Sigma^*))$. Максимальное значение выражения (14) в случае $\theta_{\check{B}} < \pi_{\check{B}}$ достигается при $\check{\mathcal{L}}^* = C_{17} \mathcal{L}(\Sigma^*) / \ln \mathcal{L}(\Sigma^*)$ и оценивается сверху величиной

$$\left(C_{18} \frac{\mathcal{L}(\Sigma^*)}{\ln \mathcal{L}(\Sigma^*)} \right)^{\mathcal{L}(\Sigma^*)/\theta_{\check{B}} - C_{19} \mathcal{L}(\Sigma^*)/\ln \mathcal{L}(\Sigma^*)}. \quad (15)$$

Перемножая (11), (12), (15) и оценивая полученное выражение сверху, получим следующую верхнюю оценку числа попарно неэквивалентных многомодульных бинарных программ, реализующих булевы функции от n переменных со сложностью не более чем \mathcal{L} :

$$\|\mathcal{U}^{\Pi(r)}(\mathcal{L}, n)\| \leq (C_{20} \mathcal{L})^{C_{21} \mathcal{L}} (C_{22} \mathcal{L}(\Sigma^*))^{\mathcal{L}(\Sigma^*)/\theta_{\check{B}}}. \quad (16)$$

При $r \geq 2$ сложность равна $\mathcal{L} = o(\mathcal{L}(\Sigma^*))$, и при применении к правой части (16) стандартного мощностного преобразования или решения неравенства $\|\mathcal{U}^{\Pi(r)}(\mathcal{L}, n)\| \geq 2^{2^n}$ множитель $(C_{20} \mathcal{L})^{C_{21} \mathcal{L}}$ не влияет на нижнюю асимптотическую оценку функции Шеннона. Учитывая дополнительно (13), получаем следующее утверждение.

Лемма 2. Для функции Шеннона $\mathcal{L}^{\Pi(r)}(n)$ при $r \geq 2$ и $\theta_{\check{B}} < \pi_{\check{B}}$ верно асимптотическое неравенство

$$\mathcal{L}^{\Pi(r)}(n) \gtrsim r \nu^{(r-1)/r} (r-1)^{-(r-1)/r} \sqrt[r]{\theta_{\check{B}}} \frac{2^{n/r}}{\sqrt[r]{n}}.$$

4. Верхняя оценка функции Шеннона

Перед получением верхних оценок функции Шеннона опишем построение вспомогательных программ. Предположим, что нам требуется реализовать при помощи бинарной программы Σ множество из не менее чем k однотипных последовательностей команд S , где каждая последовательность S имеет сложность L_S ,

не содержит команд вызова подпрограмм, имеет свой набор входных и выходных ячеек памяти, а также некоторое множество ячеек для промежуточных вычислений. Кроме того, будем считать, что переадресующие команды последовательности S могут осуществлять переход только на команды в пределах данной последовательности или в ее конец. При $r = 1$ одномодульная бинарная программа Σ содержит непосредственно k последовательностей команд S и имеет сложность, равную kL_S . При $r \geq 2$ положим $Q = \sqrt[r]{\nu^{r-1}(r-1)^{-(r-1)}kL_S}$ и построим бинарную программу Σ с ограниченной глубиной рекурсии r , состоящую из единственной подпрограммы Σ_1 , которая содержит $[Q/L_S]$ последовательностей команд S , $[(r-1)Q/\nu]$ команд вызова подпрограммы Σ_1 и $O(r)$ дополнительных команд. Программа функционирует следующим образом.

1. Если выполнение подпрограммы Σ_1 происходит не на последнем уровне рекурсии, то выполняются $[(r-1)Q/\nu]$ команд вызова подпрограммы Σ_1 . При этом на каждом уровне рекурсии у данной подпрограммы существуют входные и выходные ячейки памяти для всех последовательностей команд S , реализуемых в данном поддереве вычислений, которые распределяются таким образом, чтобы на последнем уровне рекурсии были вычислены последовательности S для всех требуемых входов и выходов.

2. На последнем уровне рекурсии r происходит вычисление $[Q/L_S]$ последовательностей команд S с заданными входами и выходами.

3. Определение уровня рекурсии требует реализации счетчика, а также компаратора размерности r , то есть программы, производящей поразрядное сравнение двух r -разрядных чисел.

4. Легко видеть, что описанная программа реализует не менее $[(r-1)Q/\nu]^{r-1} \times [Q/L_S] \geq k$ последовательностей команд S со сложностью, не превосходящей $rQ + O(L_S)$.

В итоге получаем, что сложность бинарной программы Σ с глубиной рекурсии r , $r \geq 2$, реализующая множество из не менее чем k последовательностей команд S сложности L_S , удовлетворяет неравенству

$$\mathcal{L}(\Sigma) \leq r \sqrt[r]{\nu^{r-1}(r-1)^{-(r-1)}kL_S} + O(L_S). \quad (17)$$

Пусть $\psi(y_1, \dots, y_p)$ – булева функция, существенно зависящая от всех своих переменных. Следуя [5], будем говорить, что множество ФАЛ $G \subseteq P_2(m)$ является ψ -универсальным множеством (ψ -УМ) порядка m , если для любой функции $g \in P_2(m)$ существуют функции $g_1, \dots, g_p \in G$ такие, что $g = \psi(g_1, \dots, g_p)$.

Построить ψ -универсальное множество G порядка m можно следующим образом:

- 1) выбрать параметры s_1, \dots, s_p , такие что $s_1 + \dots + s_p = 2^m$;
- 2) разбить булев куб B^m на p попарно непересекающихся компонент

$$B^m = \tau_1 \cup \dots \cup \tau_p, \quad \tau_i \cap \tau_j = \emptyset \quad \text{при } i \neq j, \quad |\tau_i| = s_i;$$

3) с учетом существенной зависимости ФАЛ $\psi(y_1, \dots, y_p)$ от всех своих переменных выбрать для каждого i , $i = 1, 2, \dots, p$, константы $\alpha_{1,i}, \dots, \alpha_{p,i} \in B$, такие что $\psi(\alpha_{1,i}, \dots, \alpha_{i-1,i}, y_i, \alpha_{i+1,i}, \dots, \alpha_{p,i}) = y_i \oplus \alpha_{i,i}$;

4) определить G как объединение множеств $G^{(1)} \cup \dots \cup G^{(p)}$, где $G^{(i)}$ – множество из всевозможных 2^{s_i} функций, равных $\alpha_{i,j}$ на всех компонентах $\tau_j \neq \tau_i$, то есть

$$G^{(i)} = \{f(x_1, \dots, x_m) : f(\beta_1, \dots, \beta_m) = \alpha_{i,j} \quad \forall \beta \in \tau_j, \quad \forall j, j \neq i\}.$$

Построенное ψ -универсальное множество порядка m состоит из $2^{s_1} + \dots + 2^{s_p}$ функций. Докажем следующую лемму о его реализации в классе бинарных программ.

Лемма 3. Пусть s_1, \dots, s_p – положительные четные числа, такие что $s_1 + \dots + s_p \geq 2^m$. Тогда существует ψ -УМ G порядка m , такое что

- 1) $|G| \leq 2^{s_1} + \dots + 2^{s_p}$;
- 2) сложность реализации системы функций из G в классе $\mathcal{U}^{\Pi(r)}$ не превосходит

$$O\left(\sqrt[r]{|G|} + p \cdot 2^{(s/2+m)/r}\right), \quad \text{где } s = \max\{s_1, \dots, s_p\}.$$

Доказательство. Построим ψ -универсальное множество функций G мощности $|G| \leq 2^{s_1} + \dots + 2^{s_p}$ описанным выше способом, разбивая булев куб B^m на «полосы» τ_1, \dots, τ_p , состоящие не более чем из s_1, \dots, s_p двоичных наборов соответственно, где под «полосой» булева куба подразумевается множество наборов, лексикографически следующих друг за другом.

Возьмем функцию $g \in G^{(i)}$ и рассмотрим ее разложение Шеннона по последней переменной x_m :

$$g(x_1, \dots, x_m) = \overline{x_m}g_0(x_1, \dots, x_{m-1}) \vee x_mg_1(x_1, \dots, x_{m-1}). \quad (18)$$

Так как мощность i -й «полосы» – четное число, то функции g_0 и g_1 принадлежат одному и тому же множеству $G^{(i)}$ всевозможных функций $g^i(x_1, \dots, x_{m-1})$ от $m-1$ переменных, равных $\alpha_{i,j}$ на «полосах» τ_j при $j \neq i$. Реализуем множество функций $G^{(i)}$ в классе $\mathcal{U}^{\Pi(r)}$ следующим образом.

1. Построим дешифратор Q_{m-1} порядка $m-1$ как множество из 2^{m-1} последовательностей команд, реализующих всевозможные конъюнкции вида $x_1^{\sigma_1}, \dots, x_{m-1}^{\sigma_{m-1}}$. Сложность каждой такой последовательности равна $O(m)$.

2. Для каждой из $2^{s_i/2}$ функций множества $G^{(i)}$ согласно ее совершенной ДНФ построим дизъюнкцию не более 2^{m-1} входов, подключенных к выходам дешифратора Q_{m-1} .

3. Используя равенство (17) для сложности реализации множеств однотипных последовательностей команд, получаем, что сложность реализации множества $G^{(i)}$ в классе программ не превышает $O(2^{(s_i/2+m)/r})$.

Суммируя сложности реализации множеств $G^{(i)}$ и применяя равенство (17) для множества из $|G|$ мультиплексорных функций μ_1 , участвующих в разложении (18), получаем, что сложность реализации ψ -УМ G в классе $\mathcal{U}^{\Pi(r)}$ не превышает

$$O\left(\sqrt[r]{|G|} + p \cdot 2^{(s/2+m)/r}\right), \quad \text{где } s = \max\{s_1, \dots, s_p\}.$$

□

Вернемся к решению задачи синтеза бинарных программ и получению верхних оценок соответствующих функций Шеннона. Воспользуемся асимптотически наилучшим методом синтеза О.Б. Лупанова [6] в изложении [5] и запишем для произвольной функции $f(x_1, \dots, x_n) \in P_2(n)$ разложение Шеннона по последним $n-q$ переменным

$$f(x_1, \dots, x_q, x_{q+1}, \dots, x_n) = \bigvee_{\sigma=(\sigma_{q+1}, \dots, \sigma_n) \in B^{n-q}} x_{q+1}^{\sigma_{q+1}} \cdots x_n^{\sigma_n} f_\sigma(x_1, \dots, x_q), \quad (19)$$

где $f_\sigma(x_1, \dots, x_q) = f(x_1, \dots, x_q, \sigma_{q+1}, \dots, \sigma_n)$.

Остаточные функции $f_\sigma(x_1, \dots, x_q)$ представим в виде суперпозиции $\psi(g_{\sigma,1}, \dots, g_{\sigma,p})$, где функции $g_{\sigma,i}$, $i = 1, 2, \dots, p$, принадлежат ψ -универсальному множеству порядка q для некоторой функции $\psi(y_1, \dots, y_p)$, вид которой будет определен ниже. Построим часть бинарной программы, соответствующую вычислению всех остаточных функций, как множество из 2^{n-q} последовательностей команд, реализующих функции ψ . Для каждого набора значений переменных

x_{q+1}, \dots, x_n необходимо вычислить только одну такую функцию, поэтому в начало каждой последовательности добавим переадресующую команду, пропускающую вычисление соответствующей функции, если конъюнкция $x_{q+1}^{\sigma_{q+1}} \cdots x_n^{\sigma_n}$ равна нулю. Значения всех 2^{n-q} требуемых конъюнкций можно предварительно вычислить и записать в некоторые ячейки памяти со сложностью не более $O(\sqrt[n]{n2^{n-q}} + n)$.

Рассмотрим два случая, отличающиеся выбором функции ψ и, соответственно, их реализациями при помощи последовательности вычислительных и переадресующих команд. В первом случае для многомодульных бинарных программ при $\theta_{\hat{B}} < \pi_{\hat{B}}$ определим функцию ψ' , равную конъюнкции t' , $t' \geq 1$, функций $\varphi' \in \hat{B}$ с минимальным удельным весом $\theta_{\hat{B}}$ зависящих от r' булевых переменных

$$\psi' = \bigwedge_{i=1}^{t'} \varphi'(y_{1+(i-1)r'}, \dots, y_{r'+(i-1)r'}).$$

Данная функция ψ' существенно зависит от $t'r'$ переменных и реализуется при помощи t' переадресующих команд с функцией φ' . В случае равенства нулю некоторой функции φ' переадресующие команды выполняют переход на последовательность команд, записывающих значение 0 в качестве результата вычисления функции $f(x_1, \dots, x_n)$. Если же значение функции φ' равно единице, то выполняется переход либо на следующую из t' переадресующих команд, либо на последовательность команд, записывающую константу 1 в итоговую ячейку памяти, если текущая переадресующая команда является последней в реализации ψ' . Сложность дополнительной части программы, необходимой для записи констант 0 и 1 в финальную ячейку памяти, равна $O(1)$.

Во втором случае рассмотрим одномодульные бинарные программы, для которых $\pi_{\hat{B}} < \pi_{\check{B}}$. Выберем функцию ψ'' как суперпозицию t'' , $t'' \geq 1$, мультиплексорных функций $\mu_1(\varphi''(y_1, \dots, y_{r''}), z_0, z_1)$ по одной из информационных переменных, где φ'' соответствует переадресующей команде с минимальным удельным весом $\pi_{\hat{B}}$ и зависит от r'' булевых переменных. Полученная функция ψ'' существенно зависит от $t''(r'' + 1) + 1$ переменных и может быть реализована при помощи последовательности, состоящей из t'' переадресующих команд с функцией φ'' . Адресные аргументы данных команд равны либо номеру следующей команды этой последовательности, если по соответствующей информационной переменной была выполнена суперпозиция, либо адресу последовательности команд, производящей запись значения некоторой функции ψ'' -универсального множества G'' в выходную ячейку памяти программы, в которой располагается итоговое значение функции $f(x_1, \dots, x_n)$. Дополнительная часть программы, выполняющая запись результата вычисления синтезируемой функции f , является общей для всех остаточных функций разложения (19), ее сложность равна $O(|G''|)$.

Учитывая все описанные выше конструкции, получаем, что реализация произвольной функции $f(x_1, \dots, x_n)$ в классе бинарных программ $\mathcal{U}^{\Pi(r)}$ включает в себя следующие части.

1. Вычисление всех функций ψ -универсального множества G порядка q со сложностью $O(\sqrt[r]{|G|} + p \cdot 2^{(s/2+q)/r})$ по лемме 3. При этом $|G|$ оценивается сверху как $p \cdot 2^s$.

2. Множество из 2^{n-q} однолинейных последовательностей команд, реализующих одну из функций ψ' , ψ'' . Для оценки сложности данного множества используется неравенство (17).

3. Реализация дешифратора порядка $n - q$, необходимого для осуществления выбора одной из 2^{n-q} функций ψ со сложностью $O(\sqrt[n]{n2^{n-q}} + n)$.

4. Дополнительные последовательности команд, выполняющие запись итогового результата вычисления функции $f(x_1, \dots, x_n)$, сложность которых не превышает $O(\sqrt[r]{|G|})$.

Замечание. Для объединения нескольких частей в одну бинарную программу можно использовать дополнительные входные ячейки памяти, в которых в двоичном представлении кодируется режим ее выполнения. В начале работы программы, являющейся объединением нескольких, осуществляется определение номера режима работы программы и выполняется переход на начало соответствующей последовательности команд. В случае объединения некоторого числа T бинарных программ, суммарная сложность дополнительных команд, использующихся для проверки режима функционирования программы, равна $O(T)$.

Выбирая значения параметров $q = \lceil 3 \log n \rceil$, $s = \lceil n - 5 \log n \rceil$ и оценивая сложность построенной программы, получим следующую лемму о верхних оценках функции Шеннона для сложности реализации булевых функций в классе бинарных программ.

Лемма 4. Для функции Шеннона $\mathcal{L}^{\Pi(r)}(n)$ при $r = 1$ и $\pi_{\mathbb{B}} < \pi_{\mathbb{B}}$ верно асимптотическое неравенство

$$\mathcal{L}^{\Pi(1)}(n) \lesssim \pi_{\mathbb{B}} \frac{2^n}{n},$$

а при $r \geq 2$ и $\theta_{\mathbb{B}} < \pi_{\mathbb{B}}$ верно

$$\mathcal{L}^{\Pi(r)}(n) \lesssim r \nu^{(r-1)/r} (r-1)^{-(r-1)/r} \sqrt[r]{\theta_{\mathbb{B}}} \frac{2^{n/r}}{\sqrt[r]{n}}.$$

Из данной леммы и лемм 1, 2 о нижних оценках функции Шеннона вытекает утверждение теоремы 4.

Благодарности. Работа выполнена при финансовой поддержке Минобрнауки РФ в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2019-1621.

Литература

1. Шеннон К. Работы по теории информации и кибернетике. – М.: Изд-во иностр. лит., 1963. – 830 с.
2. Грибок С.В. О реализации функций алгебры логики в некоторых классах программ: Дис. ... канд. физ.-мат. наук. – М., 2003. – 90 с.
3. Жуков В.В. Методы синтеза бинарных программ, допускающих рекурсивный вызов процедур // Вестн. Моск. ун-та. Сер. 15. Вычисл. матем. и кибернетики. – 2021. – № 3. – С. 3–12.
4. Жуков В.В. Синтез некоторых типов бинарных программ, допускающих рекурсивный вызов процедур ограниченной глубины // Проблемы теоретической кибернетики: Материалы XIX междунар. конф. – Казань: Казан. фед. ун-т, 2021. – С. 53–55.
5. Ложкин С.А. Лекции по основам кибернетики. – М.: Изд. отдел фак. ВМиК МГУ, 2004. – 253 с.
6. Лупанов О.Б. Асимптотические оценки сложности управляющих систем. – М.: Изд-во МГУ, 1984. – 137 с.

Поступила в редакцию
23.08.2021

Жуков Владимир Владимирович, аспирант факультета вычислительной математики и кибернетики

Московский государственный университет имени М.В. Ломоносова
ул. Ленинские Горы, д. 1, г. Москва, 119991, Россия
E-mail: zhvv117@gmail.com

ISSN 2541-7746 (Print)

ISSN 2500-2198 (Online)

UCHENYE ZAPISKI KAZANSKOGO UNIVERSITETA.
SERIYA FIZIKO-MATEMATICHESKIE NAUKI
(Proceedings of Kazan University. Physics and Mathematics Series)

2021, vol. 163, no. 3–4, pp. 276–290

ORIGINAL ARTICLE

doi: 10.26907/2541-7746.2021.3-4.276-290

Synthesis of Binary Programs with Predominance of Branching Commands

V. V. Zhukov

Moscow State University, Moscow, 119991 Russia

E-mail: zhvv117@gmail.com

Received August 23, 2021

Abstract

In this article, a model of binary programs that implement the logic algebra functions (Boolean functions) is considered. These programs consist of one or several modules, which include the following three types of commands: computational, branching, and procedure call commands. The model under study also allows recursive procedure calls. It means that the procedures can call themselves, either directly or through other procedures, during the program execution. The concept of an arbitrary basis for branching commands is introduced as a generalization of the known models. The methods for calculating the lower and upper bounds of the Shannon function for the complexity of Boolean functions implementation in the class of binary programs are presented. The complexity of a binary program is understood as the integral weight of all commands of its subprograms. The methods allow establishing the Shannon function asymptotic bounds in the case when the specific weight of branching commands is less than the specific weight of computational commands. The obtained results contribute substantially to further development of the theory of synthesis and complexity of discrete control systems.

Keywords: binary programs, Shannon function, asymptotic bounds

Acknowledgments. This study was supported by the Ministry of Education and Science of the Russian Federation as part of the program of the Moscow Center for Fundamental and Applied Mathematics under agreement no. 075-15-2019-1621.

References

1. Shannon C. *Raboty po teorii informatsii i kibernetike* [Papers in Information Theory and Cybernetics]. Moscow, Izd. Inostr. Lit., 1963. 830 p. (In Russian)
2. Gribok S.V. On the implementation of logic algebra functions in some classes of programs. *Cand. Phys.-Math. Sci. Diss.* Moscow, 2003. 90 p. (In Russian)

3. Zhukov V.V. Ways of synthesizing binary programs admitting recursive call of procedures. *Moscow Univ. Comput. Math. Cybern.*, 2021, vol. 45, no. 3, pp. 87–95. doi: 10.3103/S0278641921030079.
4. Zhukov V.V. Synthesis of some types of binary programs admitting recursive call of procedures with limited depth. *Problemy teoreticheskoi kibernetiki: Materialy XIX mezhdunar. konf.* [Problems of Theoretical Cybernetics: Proc. XIX Int. Conf.]. Kazan, Kazan. Fed. Univ., 2021, pp. 53–55. (In Russian)
5. Lozhkin S.A. *Lektsii po osnovam kibernetiki* [Lectures on the Principles of Cybernetics]. Moscow, Izd. Itd. Fak. VMiK MGU, 2004. 235 p. (In Russian)
6. Lupanov O.B. *Asimptoticheskie otsenki slozhnosti upravlyayushchikh sistem* [Asymptotic Complexity Bounds of Control Systems]. Moscow, Izd. MGU, 1984. 137 p. (In Russian)

⟨ **Для цитирования:** Жуков В.В. Синтез бинарных программ с преобладанием команд переадресующего типа // Учен. зап. Казан. ун-та. Сер. Физ.-матем. науки. – 2021. – Т. 163, кн. 3–4. – С. 276–290. – doi: 10.26907/2541-7746.2021.3-4.276-290. ⟩

⟨ **For citation:** Zhukov V.V. Synthesis of binary programs with predominance of branching commands. *Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki*, 2021, vol. 163, no. 3–4, pp. 276–290. doi: 10.26907/2541-7746.2021.3-4.276-290. (In Russian) ⟩