УДК 519.682

Е.Е. Малафеев[1], Е.Е. Малафеев[1], А.В. Павлик[2]

[1]*ОАО «АО Научно-исследовательский институт радиотехнических измерений», Харьков*
[2]*Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", Харьков*

СТРУКТУРНО-ФУНКЦИОНАЛЬНЫЙ МЕТОД ПРЕОБРАЗОВАНИЯ АЛГОРИТМИЧЕСКИХ СТРУКТУР

Рассмотрена задача эквивалентного преобразования алгоритмических структур с коммутативными условиями. Предложен структурно-функциональный метод преобразования. Исследованы процедуры выбора структуры алгоритма и формирования логических функций.

Ключевые слова: *логическая функция, алгоритм, структура, коммутативное условие.*

Введение

Постановка проблемы. В связи с развитием информационных технологий возросли требования к алгоритмам обработки информации. Среди задач преобразования алгоритмов не только классические - оптимизация и упрощение структуры, повышение быстродействия, но и специфические, например, запутывание алгоритмов. Основные направления в решении проблемы состоят в поиске эффективных тождественных преобразований. При разработке алгоритмического и программного обеспечения для сложных систем обработки информации и управления необходимо использовать математический аппарат, с помощью которого техника формальных преобразований схем алгоритмов была бы более простой и удобной.

Анализ исследований и публикаций. Возможность формальных преобразований алгоритмов была доказана в процессе развития автоматной теории алгоритмов. В.М. Глушковым [1] был предложен алгебраический подход, основанный на двухосновной алгебре, используемой для представления преобразований, выполняемых алгоритмами. Основными множествами алгебры алгоритмов является множество операторов, действующих в информационной среде, и множество условий, определённых на ней.

Алгоритмы, представленные в виде регулярных программ, удобно анализировать и выполнять над ними оптимизирующие преобразования, пользуясь соотношениями алгебры алгоритмов. Опыт практического применения регулярных схем алгоритмов (РСА) для синтеза алгоритмов и структур специального вида, управляющих вычислительных устройств показал ограниченность класса алгоритмов, которые можно записать в РСА с помощью набора базовых операций.

Применение систем алгоритмических алгебр для реализации структурных программ сопряжено с определенными трудностями в связи с необходимостью априорного представления алгоритмов и программ в структурной форме, обладающей чаще всего некоторой избыточностью [2].

Среди известных алгебр регулярных схем алгоритмов следует отметить расширенную алгебру с коммутативными условиями для регулярных схем алгоритмов [3].

Анализ известных методов преобразования алгоритмических структур показывает, что они имеют высокую трудоемкость и ограниченную область применения.

Целью статьи является разработка нового подхода к преобразованию алгоритмических структур, позволяющего упростить технику формальных преобразований схем алгоритмов.

Изложение основного материала исследования

Для преобразования алгоритмов с коммутативными условиями разработан структурно-функциональный метод, суть которого состоит в следующем.

Для описания алгоритмических структур с коммутативными условиями будем использовать совершенную дизъюнктивную форму нормальную форму алгоритма (СДНФ) [3].

СДНФ алгоритма, который описывается множеством операторов $P = \{P_1, \ldots, P_k\}$ и множеством условий $X = \{X_1,\ldots,X_n\}$, в общем случае имеет вид:

$$A(X) = P_1^{U1} \vee \ldots \vee P_i^{Ui} \vee \ldots \vee P_h^{Uh},$$

где U_i – множество наборов значений условий, при которых выполняется i-й оператор.

Под преобразованием алгоритма будем понимать замену его переменных на множество

$$FN = \{F_1(X), F_2(X), \ldots , F_n(X)\}$$

логических функций.

В результате преобразований исходный алгоритм A(X) преобразуется в тождественный алгоритм A(FN).

Областью определения логической функции $F_i(X)$ будем называть множества наборов логических условий (X), при которых i-ая логическая функция принимает значение "1" (N_i^1) и значение "0" (N_i^0). Мощность этих множеств, соответственно $|N_i^1|$ и $|N_i^0|$. Если $|N_i^1| + |N_i^0| < 2^n$, то логическая функция является недоопределенной и значения ее на наборах, не входящих в $N_i^1 \cup N_i^0$ можно доопределить значениями "0" или "1", в зависимости от требований к тождественному алгоритму. Множество логических функций, определенных на наборах N_i^1 и N_i^0 при $|N_i^1| + |N_i^0| < 2^n$, будем называть N- эквивалентными.

При преобразовании алгоритмов структурно-функциональным методом необходимо определить структуру алгоритма и сформировать соответствующие функции.

Основные этапы метода.

1. Для заданного количества операторов формируется множество вариантов структур алгоритма. Под структурой алгоритма будем понимать вид алгоритма с указанным расположением логических функций и операторов.

2. В зависимости от цели преобразования (оптимизация, запутывание и др.) выбирается соответствующий вариант структуры.

3. Последовательно (от начала алгоритма до соответствующих операторов) формируется для каждой логической функции область определения.

4. Если $|N_i^1| + |N_i^0| < 2^n$, то формируется множество N-эквивалентных логических функций, из которых выбирается, в зависимости от требований к тождественному алгоритму, соответствующая функция.

Рассмотрим ***пример преобразования алгоритма*** структурно-функциональным методом.

Пусть задана СДНФ алгоритма:

$$A = a^{0000} \vee a^{0001} \vee b^{0010} \vee b^{0011} \vee c^{0100} \vee c^{0101} \vee c^{0110} \vee c^{0111} \vee d^{1000} \vee e^{1001} \vee d^{1010} \vee e^{1011} \vee d^{1100} \vee e^{1101} \vee e^{1110} \vee e^{1111}.$$

Для пяти условных переменных существуют варианты структур алгоритма, приведенные на рис. 1.

Для запутывающих преобразований алгоритма, очевидно, более подходит первая структура, которая, в отличие от других, не выделяет уже при первой проверке оператор, что усложняет вид функций.

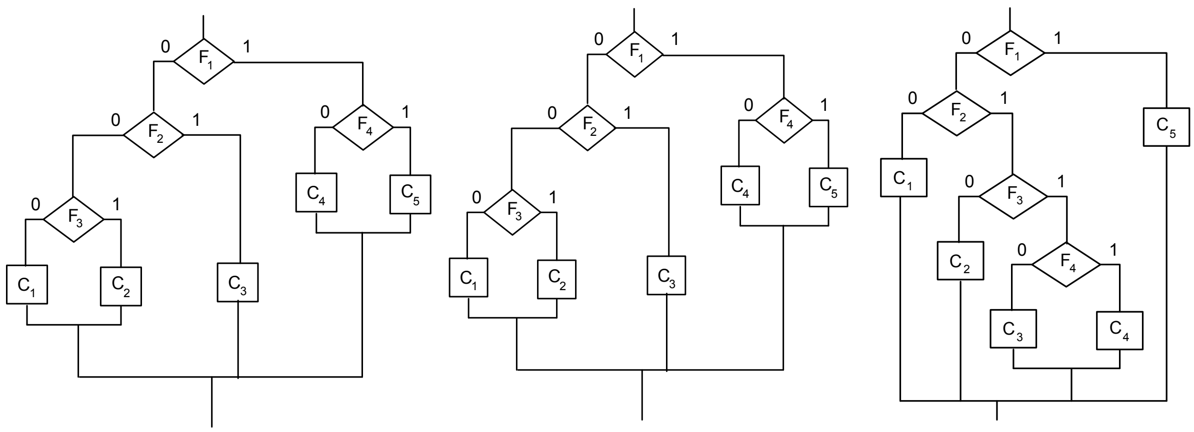

Рис. 1. Варианты алгоритмических структур

Расставляем операторы на выходах условных операторов (рис. 2).

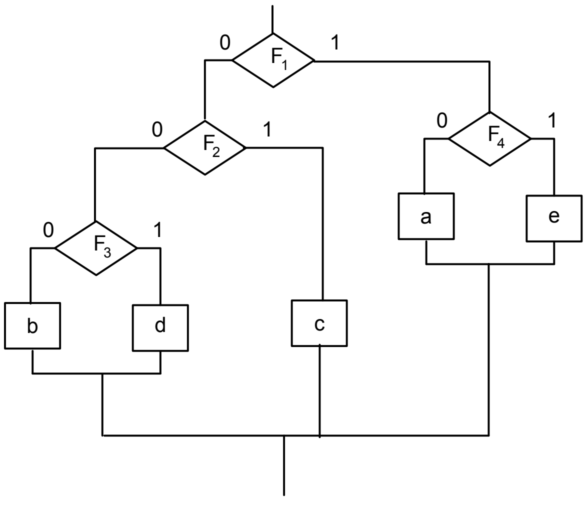

Рис. 2. Выбранная структура преобразованного алгоритма

Преобразуем совершенную дизъюнктивную форму нормальную форму алгоритма с учетом структуры.

$$A = (a \vee e)^{0000 \vee 0001 \vee 1001 \vee 1011 \vee 1101 \vee 1110 \vee 1111} \vee$$
$$\vee (b \vee c \vee d)^{0010 \vee 0011 \vee 0100 \vee 0101 \vee 0110 \vee 0111 \vee 1000 \vee 1010 \vee 1100}.$$

Формируем область определения логической функции $F_1(X)$:

$$N_1^1 =$$
$$= \{0000, 0001, 1001, 1011, 1101, 1110, 1111\};$$

$$N_i^0 =$$
$$= \{0010, 0011, 0100, 0101, 0110, 0111, 1000, 1010, 1100\}.$$

Поскольку

$$|N_i^1| + |N_i^0| = 2^n,$$

то логическая функция $F_1(X)$ является полностью определенной, и после упрощения имеет вид:

$$F_1(X) = \bar{X}_1X_2 \text{ v } X_1X_4 \text{ v } X_1X_2X_3.$$

Результаты вычислений для логических функций $F_2(X)$, $F_3(X)$ и $F_4(X)$ приведены в табл. 1.

Таблица 1

Характеристики логических функций

Вид функции	N_1^1	N_1^0	N –эквивалентные логические функции
$F_4(X)$	1001, 1011, 1101, 1110, 1111	0000, 0001	X1X4 v X1X2X3, X1, X1 v X2, X1 v X3, X1 v X3X4, X1 v X2 vX3
$F_2(X)$	0100, 0101, 0110, 0111	0010, 0011, 1000, 1010, 1100	$\bar{X}1X2$, $\bar{X}1$ $\bar{X}3$ v $\bar{X}1X2$, $\bar{X}1X2$ v X2X4, $\bar{X}1$ $\bar{X}3$ v X2X3
$F_3(X)$	0010, 0011	1000, 1100	X1, $\bar{X}3$, X1 v X2, X1 v $\bar{X}3$, X1 v $\bar{X}3X4$, X1 $\bar{X}4$

Приведенный пример структурно-функционального преобразования алгоритмов показал, что рассматриваемая задача является многовариантной.

При выборе варианта построения эквивалентных алгоритмов необходимо учитывать характеристики разработанного алгоритма и программы и стоимость работы, требуемой для решения вычислительной проблемы.

При программной реализации булевых функций, как правило, минимизируются показатели сложности программ: время их работы и память.

Оптимальная (по различным показателям качества и по трудоемкости) программная реализация систем булевых функций в разных базисах представляет собой актуальную проблему и в настоящее время отсутствуют эффективные методы ее оценки.

Выводы

Рассмотренный метод структурно-функционального преобразования алгоритмических структур с коммутативными условиями позволяет решить различные задачи преобразования алгоритмов.

Для реализации метода и оценки его эффективности для различного количества условных переменных и операторов необходимо разработать программное обеспечение, позволяющее автоматизировать процесс преобразований; разработать математические модели для компьютерного представления булевых функций и на их основе различных преобразований, включая декомпозицию в разных базисах.

Список литературы

1. Глушков В.М. Алгебра. Языки. Программирование. II изд. / В.М.пГлушков, Г.Е.тЦейтлин Е.Л., Ющенко. – К.: Наук. думка, 1978. – 319 с.

2. Жихарев В.Я. Математические основы проектирования рекурсивных автоматов с программируемой логикой / В.Я. Жихарев, В.М. Илюшко, И.В.Чумаченко. – Х.: Факт, 1999. – 144 с.

3. Чумаченко И.В. Расширенная алгебра регулярных схем алгоритмов с коммутативными условиями / И.В.Чумаченко // Авіаційно-космічна техніка і технологія: Зб. наук. праць. – Х.: Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2000. – Вип. 20. – С. 154-158.

Поступила в редколлегию 3.06.2010

Рецензент: д-р техн. наук, проф. В.М. Илюшко, Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", Харьков.

СТРУКТУРНО-ФУНКЦІОНАЛЬНИЙ МЕТОД ПЕРЕТВОРЕННЯ АЛГОРИТМІЧНИХ СТРУКТУР

Є.Є. Малафєєв, Є.Є. Малафєєв, Г.В. Павлік

Розглянуто завдання еквівалентного перетворення алгоритмічних структур з комутативними умовами. Запропонований структурно-функціональний метод перетворення. Досліджені процедури вибору структури алгоритму і формування логічних функцій.

Ключові слова: *логічна функція, алгоритм, структура, комутативна умова.*

STRUCTURALLY-FUNCTIONAL METHOD OF TRANSFORMATION OF ALGORITHMIC STRUCTURES

Ye. Ye. Malafeev, Ye. Ye. Malafeev, A.B. Pavlik

The task of equivalent transformation of algorithmic structures is considered with commutative terms. The structurally-functional method of transformation is offered. Procedures of choice of structure of algorithm and forming of boolean functions are explored.

Keywords: *boolean function, algorithm, structure, commutative condition.*