УДК 681.325

БЕСПОВТОРНЫЕ АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ

д.т.н. И.В. Чумаченко, В.В. Косенко, Н.В. Доценко

Исследованы бесповторные алгоритмические структуры, предложена их классификация и методы перечисления.

Для унификации программно-аппаратных средств необходимо определить вид алгоритмов, наиболее широко применяемых на практике, исследовать их свойства. Анализ алгоритмов обработки информации и управления, проведенный в работах [1, 2] показал, что формулы, описывающие системы управления газотурбинными установками компрессорных станций газопроводов, маслонасосов и кранов, устройства управления различным технологическим оборудованием, устройств автоматической телефонии, нерегулярные логические схемы систем управления судовыми техническими средствами имеют много общего. Выяснилось, что каждая из этих формул принадлежит хотя бы одному из четырех классов упорядоченного типа: бесповторным формулам; функциям, обладающим групповой инвариантностью; функциям, допускающим разделительную декомпозицию; однородным функциям.

Таким образом, при проектировании систем управления и дискретных устройств, в большинстве случаев, используется довольно узкий класс формул – и для этих устройств бесповторность формул является характерным свойством, тем более что и однородные, и разделимые функции обычно имеют малую повторность переменных в соответствующих формулах.

Свойство бесповторности используется при анализе различных алгоритмов, схем, функций, однако общего подхода к исследованию этого класса алгоритмических структур в настоящее время нет. Это, безусловно, затрудняет анализ и разработку унифицированных алгоритмических структур. В связи с этим предлагается различать различные виды бесповторности в зависимости от формы представления алгоритмов.

Пусть алгоритмическая структура (АС) содержит условные операторы, заданные множеством **У** и линейные, заданные множеством **Р**. Для описания алгоритмов используется расширенная алгебра с коммутативными условиями [3].

Определение 1. АС называется условно бесповторной, если в регулярное выражение условные переменные входят только один раз.

Например, АС вида

$$\mathbf{A_1 = P_1(\,(P_3(P_2 \vee P_4)^{\alpha}(P_5 \vee P_6)^{\mu} \vee P_2)^{\beta} \vee P_3(P_6 \vee P_8)^{\varphi})^{\gamma}}$$

является условно бесповторной АС.

Определение 2. АС называется операторно бесповторной, если в регулярное выражение линейные операторы входят только один раз.

Например, АС вида

$$\mathbf{A_2 = (P_3(P_2 \vee P_4)^{\alpha} \vee P_1)^{\beta} \vee (P_5 \vee P_6)^{\varphi})^{\gamma}}$$

является операторно бесповторной АС.

Определение 3. АС называется операторно-условно бесповторной, если АС является условно и операторно бесповторной.

Повторные АС могут быть преобразованы в бесповторные путем введения дополнительных переменных.

Частным случаем бесповторных АС является подмножество бесповторных АС, у которых все операторы являются тождественными. Это относится в первую очередь к управляющим алгоритмам. Описание таких АС представляет собой бесповторную логическую (булеву) функцию, т.е. функцию, в формульной записи которой число букв равно числу переменных.

Примеры бесповторных логических функций:

$$\mathbf{F_1(x_1,\ldots,x_4) = x_2 \vee \bar{x}_1(x_3 \vee \bar{x}_4)}; \quad \mathbf{F_2(x_1,\ldots,x_6) = x_1 \vee x_2x_3 \vee x_4 \vee x_5 \vee x_6}.$$

Следующим этапом исследования бесповторных АС является их классификация, которая включает в себя:

– анализ видов эквивалентности для рассматриваемого класса алгоритмов и выбор наиболее рационального с точки зрения их реализации;

– табулирование, т.е. подсчет числа классов эквивалентности и построение множества типовых представителей (каталога типовых схем).

Для решения задачи классификации бесповторных АС необходимо определить разбиение алгоритмов на классы, при котором: с одной стороны, эквивалентные преобразования были бы легко осуществимы; с другой – количество вариантов было минимальным.

В результате классификации множество бесповторных АС распадается на попарно-непересекающиеся классы – множества однотипных схем алгоритмов и соответствующих им регулярных выражений (формул). Каждую формулу данного класса можно выбрать в качестве представителя этого класса. Схемы алгоритмов, принадлежащие одному классу, реализуются одинаковыми программными или аппаратными средствами. Поэтому для каждого класса достаточно реализовать лишь одну схему алгоритма, структура которой описывается формулой представителя класса. Получение любой схемы алгоритма, принадлежащей классу, при этом осуществляется путем заданных преобразований.

Для классификации бесповторных АС разработан метод, основанный на разбиении схем алгоритмов на классы эквивалентности, с учетом их структуры («скелета алгоритма»). Выбор такого метода классификации обусловлен тем, что каждому условному оператору соответствует переход, т.е. изменение последовательности выполнения команд в соответствии с алгоритмом программного обеспечения. Согласно статистике переходы встречаются в среднем через каждые шесть команд. Существуют безусловные переходы (типа GOTO) – когда управление передается по новому указанному адресу и условные (типа IF) – когда изменяется ход выполнения программы в зависимости от результатов сравнения. Условные переходы снижают общую производительность центрального процессора, т.к. в ожидании этого перехода конвейер работает вхолостую. Оптимизируя структуру алгоритма, можно повысить производительность центрального процессора при решении заданного класса задач.

Для описания структуры алгоритма введем в рассмотрение понятие полиномиальной формы алгоритма.

Определение 4. Полиномиальной формой алгоритма (ПФА) называется представление структуры алгоритма в виде полинома

$$\mathbf{M = a_1 x^{r1} + a_2 x^{r2} + \ldots + a_i x^{ri} + \ldots + a_n x^{rn}},$$

где $\mathbf{a_i}$ – количество различных маршрутов от начальной вершины алгоритма до конечной, проходящих через **i** (**i = 1, …, n**) условных вершин; **x** – условная переменная; $\mathbf{r_i}$ – ранг конфигурации маршрута алгоритма.

Метод построения полиномиальной формы алгоритма состоит в последовательном рассмотрении маршрутов из начальной вершины в конечную, при этом в оценке длины маршрута участвуют только условные операторы.

Определение 5. Две бесповторных АС называются **M**-эквивалентными, если они имеют одинаковую полиномиальную форму алгоритма.

Рассмотрим свойства эквивалентных бесповторных АС. Пусть регулярное выражение **A** является произведением частных алгоритмов **A1(X1)**, **A2(X2)**,…, **Ak (Xk)** и множества переменных **X1**, **X2**,…, **Xk** не пересекаются. В этом случае все алгоритмы, полученные в результате перестановки в регулярном выражении указанных частных алгоритмов, относятся к одному классу **M**-эквивалентности.

Введенные многочлены можно рассматривать как частный случай алгебраических многочленов от одной переменной

$$\mathbf{M = a_1 x^{r1} + a_2 x^{r2} + \ldots + a_i x^{ri} + \ldots + a_n x^{rn}},$$

обладающих специфическими свойствами. Для однозначности, будем называть эти многочлены **D**-многочленами. Рассмотрим основные определения, свойства **D**-многочленов и операции над ними.

Порядок членов **D**-многочлена, а также порядок множителей в каждом члене **D**-многочлена можно менять произвольно, точно так же можно вводить или опускать члены **D**-многочлена с нулевыми коэффициентами. Коэффициенты принадлежат полю неотрицательных целых чисел.

Определение 6. Два **D**-многочлена называются равными, если после приведения подобных все члены **D**-многочлена с отличными от нуля коэффициентами оказываются попарно одинаковыми.

Многочлен, полученный в результате действия над **D**-многочленами сложения и умножения на основании переместительного, сочетательного и распределительного закона является **D**-многочленом.

Совокупность всех **D**-многочленов с коэффициентами из заданного поля образуют кольцо – кольцо **D**-многочленов над данным полем. Это кольцо не имеет делителей нуля, т.е. произведение **D**-многочленов, не равных 0, не может дать 0.

Определение 7. Если для двух **D**-многочленов **P(X)** и **Q(X)** можно найти такой **D**-многочлен **R(X)**, что **P(X) = Q(X)*R(X)**, то **P(X)** делится на **Q(X)**. При этом **Q(X)** называется делителем, а **R(X)** – частным.

Определение 8. **D**-многочлен, который можно представить в виде произведения **D**-многочленов низших степеней с коэффициентами из данного поля, называется приводимым в данном поле, в противном случае – неприводимым. В табл. 1 приведены примеры **D**-многочленов.

Таблица 1

Приводимость D-многочленов

N	№ п/п	Вид полинома	Разложение
1	1	2 X	M(1,1)
2	1	4 X^2	M(1,1) * M(1,1)
	2	2 X^2 + 1 X	X (M(1,1) + 1)
3	1	8 X^3	M(1,1) * M(1,1) * M(1,1)
	2	4 X^3 +2 X^2	M(1,1) * M(2,2)
	3	4 X^2	X (M(1,1) + M(1,1))
	4	2 X^3 + 1 X^2 + 1 X	X (M(2,2) + 1)
	5	4 X^3 + 1 X	X (M(1,1) * M(1,1) +1)

Предложенный подход позволяет создавать каталоги типовых бесповторных АС, необходимых для унификации программно-аппаратных средств.

ЛИТЕРАТУРА

1. *Жихарев В.Я., Илюшко В.М., Чумаченко И.В. Математические основы проектирования рекурсивных автоматов с программируемой логикой: Монография. – Х.: Факт, 1999. – 144 с.*
2. *Жихарев В.Я., Илюшко В.М., Чумаченко И.В. Проектирование электронных компиляторов: Монография. – Х.: Факт, 1999. – 88 с.*
3. *Чумаченко И.В. Расширенная алгебра регулярных схем алгоритмов с коммутативными условиями // Авіаційно-космічна техніка і технологія: – Х.: Нац. аерокосміч. ун-т «Харк. авіац. ін-т».- 2000. – Вип. 20. – С. 154 - 158.*

Поступила 16.05.2002

***ЧУМАЧЕНКО Игорь Владимирович**, доктор техн. наук, зав. кафедрой НАКУ «ХАИ». В 1977 году окончил ХАИ. Область научных интересов – автоматизированные системы обработки информации и управления.*

***КОСЕНКО Виктор Васильевич,** нач. лаборатории ХВУ. В 1982 году окончил ХВВКИУ. Область научных интересов – автоматизированные системы обработки информации и управления.*

***ДОЦЕНКО Наталья Владимировна**, аспирантка НАКУ «ХАИ», который окончила в 2001 году. Область научных интересов – автоматизированные системы обработки информации и управления.*