

## УНИВЕРСАЛЬНЫЕ АЛГОРИТМИЧЕСКИЕ ПРЕОБРАЗОВАТЕЛИ

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

*Исследованы настраиваемые алгоритмические преобразователи, способы их настройки, оценка логической эффективности, предложен метод построения.*

Унификация и типизация алгоритмических средств имеют перво-степенное значение при разработке автоматизированных систем обработки информации и управления. Под унификацией понимают рациональное сокращение типов изделий или процессов одинакового функционального назначения, а под типизацией – разработку типовых решений, в которых отражаются общие для ряда процессов элементы или характеристики [1, 2]. Унификация и типизация алгоритмов обработки информации позволяет снизить общее их число, что упрощает организацию адаптивного комплекса алгоритмов, а модернизация объектов или расширение их состава не влечет за собой существенных изменений математического обеспечения.

Для разработки унифицированных алгоритмических, программных и аппаратных средств, универсальных в заданном классе, предлагается использовать настраиваемые алгоритмические преобразователи.

Настройка алгоритма – это подстановка в качестве переменных элементов множества  $\{0, 1, x_1, \dots, x_n\}$ . В результате настройки регулярное выражение, описывающее алгоритм, преобразуется в тождественное ему. Полученный в результате преобразований алгоритм назовём частным алгоритмом. Алгоритм, реализующий заданное множество частных алгоритмов, будем называть универсальным в заданном классе алгоритмом.

Если для всех заданных настроек множество номеров настраиваемых переменных фиксировано, т.е. в настройке участвуют одни и те же переменные, то будем говорить, что используются разделимые переменные, в противном случае – неразделимые переменные. Выбор разбиения переменных зависит от требований к настройкам.

При исследовании настраиваемых алгоритмических преобразователей возникают следующие задачи:

- оценить логическую эффективность алгоритма при заданном множестве настроек, т.е. определить множество реализуемых им различных частных алгоритмов;
- построить универсальный алгоритм, реализующий с помощью настроек заданное множество частных алгоритмов.

Для оценки логической эффективности необходимо для каждой настройки определить вид реализуемого частного алгоритма, что удобно производить с помощью алгоритмических позиционных диаграмм (АПД) [3], т.к. каждой настройке соответствует определённая конфигурация на диаграмме. При анализе логической эффективности, в большинстве случаев интересует не только множество реализуемых с помощью настройки частных алгоритмов, а множество реализуемых частных алгоритмов, принадлежащих некоторым классам эквивалентности. Были исследованы неповторные алгоритмические структуры [4] и определены типовые алгоритмы, реализуемые в результате настроек. В табл. 1 приведены полученные результаты.

Таблица 1

Частные алгоритмы, реализуемые при настройках

n	№ п/п	Вид типового полинома	Реализуемые типовые полиномы при настройках (S = 1)
1	1	$2 X$	-
2	1	$4 X^2$	T1.1
	2	$2 X^2 + 1 X$	T1.1
3	1	$8 X^3$	T2.1
	2	$4 X^3 + 2 X^2$	T1.1, T2.1, T2.2
	3	$4 X^2$	T1.1, T2.2
	4	$2 X^3 + 1 X^2 + 1 X$	T1.1, T2.2
	5	$4 X^3 + 1 X$	T2.1, T2.2
4	1	$16 X^4$	T3.1
	2	$8 X^4 + 4 X^3$	T2.1, T3.1, T3.2
	3	$8 X^4 + 2 X^2$	T1.1, T3.1, T3.2, T3.5
	4	$8 X^4 + 1 X$	T3.1, T3.5
	5	$4 X^4 + 2 X^3 + 2 X^2$	T1.1, T2.1, T3.2, T3.4
	6	$8 X^3$	T2.1, T3.2, T3.3
	7	$4 X^4 + 4 X^3 + 1 X^2$	T2.2, T3.2
	8	$4 X^4 + 2 X^3 + 1 X$	T2.2, T3.2, T3.4, T3.5
	9	$4 X^4 + 1 X^2 + 1 X$	T1.1, T3.4, T3.5
	10	$2 X^4 + 1 X^3 + 1 X^2 + 1 X$	T1.1, T2.2, T3.4
	11	$4 X^3 + 1 X$	T2.2, T3.3, T3.4
	12	$4 X^3 + 2 X^2$	T1.1, T2.1, T3.3, T3.5
	13	$2 X^3 + 3 X^2$	T1.1, T2.2, T3.3, T3.4

Приведённые результаты показывают, что структура алгоритма влияет на его логическую эффективность и рассмотренный метод позволяет анализировать логическую эффективность алгоритмов и выбирать наиболее эффективные.

Для автоматизации процесса анализа эффективности алгоритмов было разработано программное обеспечение, описанное в [5].

Класс алгоритмов (регулярных выражений), реализуемых алгоритмом путём настройки, назовём функциональным базисом алгоритма.

Программную или аппаратную реализацию алгоритма, реализующего заданное множество частных алгоритмов, будем называть универсальным в заданном классе алгоритмов алгоритмическим преобразователем (УАП).

Задача построения УАП может быть сформулирована следующим образом.

Для заданного функционального базиса УАП

$$\Psi = \{A_1(Z_1), \dots, A_k(Z_k)\},$$

где  $Z_1, \dots, Z_k$  – множества переменных частных алгоритмов, требуется найти универсальный алгоритм  $A(x_1, \dots, x_n)$  такой, что для любого частного алгоритма  $A_i \in \Psi, i = 1, \dots, k$  существует настройка  $H_j$ , такая, что  $A(H_j) = A_i(Z_i)$ .

Для построения УАП разработан метод, состоящий из следующих этапов:

1. Оценка верхней границы количества переменных УАП.
2. Построение множества настроек УАП.
3. Формирование начальной АПД УАП.
4. Выделение частных алгоритмов, реализуемых УАП при настройках, анализ возможности их доопределения для реализации алгоритмов из заданного базиса.
5. Формирование обобщённой АПД УАП.
6. Анализ возможных вариантов доопределения АПД УАП и выбор оптимального.
7. Построение программной или аппаратной реализации УАП.

В основе оценки верхней границы количества переменных УАП лежит следующая теорема.

*Теорема.* Максимальное количество переменных универсального алгоритма  $A(x_1, \dots, x_n)$ , реализующего путем настройки  $k$  алгоритмов  $A_1, \dots, A_k$ , от  $m$  определяется выражением  $n = m + \lceil \log 2k \rceil$ .

*Доказательство.* Наибольшее количество переменных имеет УАП с разделимыми настроечными переменными, т.е. номера переменных, выделенных для настройки, фиксированы. Количество настроечных переменных в этом случае равно  $n - m$ . Количество настроек равно  $2^{n - m}$ .

Поскольку каждому частному алгоритму должна соответствовать своя настройка, то  $2^n - m \geq k$  или  $n = m + \lceil \log 2k \rceil$ , что и требовалось доказать.

Предложенный подход к разработке унифицированных алгоритмических, программных и аппаратных средств, универсальных в заданном классе, имеет высокую эффективность и позволяет создавать патентоспособные технические решения [6].

## 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. Диагностирование и прогнозирование технического состояния авиационного оборудования / В.Г. Воробьев, В.В. Глухов, Ю.В. Козлов и др. Под ред. И.М. Синдеева. – М.: Транспорт, 1984. – 191 с.
3. Чумаченко И.В. Алгоритмические позиционные диаграммы // *Авіаційно-космічна техніка і технологія*. – Х.: Нац. аерокосміч. ун-т «ХАІ». – 2001. – Вип. 22. – С. 163 - 167.
4. Чумаченко И.В., Косенко В.В., Доценко Н.В. Бесповторные алгоритмические структуры // *Системи обробки інформації*. – Х.: НАНУ, ПАНМ, ХВУ. – 2002. – Вип. 3(19). – С. 220 - 223.
5. Чумаченко І.В., Косенко В.В. Комп'ютерна програма "Програма аналізу алгоритмів": Свід. Держ. реєстр. прав автора на твір № 5966. – За-реєстр. в Держ. департ. інтелектуальної власності Мін. освіти і науки України 23 липня 2002 р.
6. Патент України № 38733 А, G06F17/11. Пристрій обробки інформації / Чумаченко І.В. – № 2000095247; Заявлено 12.09.2000; Опубл. 15.05.2001, Бюл. № 4.– 5 с.

Поступила 17.08.2002

**ЧУМАЧЕНКО Игорь Владимирович**, доктор техн. наук, зав. кафедрой Национального аэрокосмического университета «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 Национального аэрокосмического университета «ХАИ», который окончила в 2001 году. Область научных интересов – автоматизированные системы обработки информации и управления.