УДК 681.325

ЭКВИВАЛЕНТНОСТЬ АЛГОРИТМИЧЕСКИХ СТРУКТУР

д.т.н., проф. И.В. Чумаченко, Д.Н. Бугас

При рассмотрении типовых вариантов алгоритмических структур (АС) выбирается эквивалентная АС только из множества типовых представителей, поэтому задача оптимизации структуры упрощается.

Введение. Сложность современных научно-технических проблем, связанных с интенсивным освоением новых областей применения вычислительной техники, диктует необходимость совершенствования средств их проектирования. Получение обоснованных и устойчивых проектных решений немыслимо без использования методов автоматизации проектирования. При разработке автоматизированных систем обработки информации и управления первостепенное значение имеют вопросы унификации и типизации алгоритмических средств, т.е. создание универсальных в заданном классе средств, реализующих при соответственном преобразовании заданное множество типовых решений. В основе унификации и типизации алгоритмов обработки информации, позволяющих упростить организацию адаптивного комплекса алгоритмов, лежит разбиение множества алгоритмов на классы эквивалентности [1].

Вопрос эквивалентности алгоритмов и разработанных на их основе схем программ особенно важен при разработке алгоритмического и программного обеспечения, поскольку схема программы – это математическая абстракция программы, отражающая ее структурные свойства. Структура программы, в частности информационные и управляющие связи ее команд и других составных частей, при построении схем программ обычно сохраняются и свойства схем распространяются на целый класс программ, которые могут быть получены из схемы при той или иной интерпретации [2].

Д. Лакхэм, Д. Парк, М. Патерсон [3] и А.А. Летичевский [4] доказали неразрешимость проблемы функциональной эквивалентности. Однако, для практики программирования главный интерес представляют не отрицательные результаты о неразрешимости таких проблем, как эквивалентность, тотальность, пустота или свобода, а положительные результаты, связанные с выделением классов схем, в которых такие проблемы разрешимы. Они используются в оптимизации программ, при доказательстве правильности и построении инструментальных систем преобразования программ. Так

Ю.И. Янов [5] формализовал понятие схемы программы, определил отношение эквивалентности схем и исследовал проблему эквивалентности для класса схем, получивших впоследствии название схем Янова. В [6] Н.А. Криницкий исследовал проблему эквивалентности и эквивалентных преобразований стандартных схем, причем для подкласса схем без циклов найден алгоритм распознавания эквивалентности и построена полная система преобразований, позволяющая любую пару эквивалентных схем автоматически преобразовать друг в друга.

В определении понятия «эквивалентность» существуют различные подходы, например «семантический» подход, понятие «эквивалентности» в котором вводится с использованием понятия интерпретации. Суть «синтаксического» подхода к определению отношения эквивалентности состоит в следующем [2]. Понятие интерпретации и интерпретированной схемы отсутствует, вместо него описывается некоторый процесс блуждания по схеме, порождающий множество цепочек схемы. Схемы считаются эквивалентными, если между порожденными множествами цепочек этих схем существует некоторое взаимно однозначное соответствие такое, что сопоставленные друг другу цепочки эквивалентны. Логико-термальную эквивалентность ввел в рассмотрение В.Э. Иткин [7], он же построил первый весьма сложный алгоритм ее распознавания.

При рассмотрении эквивалентности алгоритмов существенную роль играют выбранные преобразования, однако этот вопрос в литературе освещен недостаточно полно.

Целью настоящей работы является рассмотрение видов преобразований и способов классификации алгоритмических структур с коммутативными условиями и аранжируемой структурой.

В результате классификации множество схем алгоритмов распадается на попарно-непересекающиеся классы эквивалентности – множества однотипных схем алгоритмов и соответствующих им регулярных выражений (формул). Каждую схему алгоритма или 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 [1].

Рассмотрим виды эквивалентности алгоритмических структур (АС) с коммутативными условиями и аранжируемой структурой. В соответствии с принятыми в [7] определениями бесскобочная запись алгоритма называется дизъюнктивной нормальной формой алгоритма (ДНФА). ДНФА, который описывается множеством операторов $P = \{P_1, \dots, P_k\}$ и множеством условий $X = \{X_1, \dots, X_n\}$, в общем случае имеет вид

$$A = П_1^{U1} \vee \dots \vee П_i^{Ui} \vee \dots \vee П_h^{Uh},$$

где $П_i^{Ui}$ – произведение соответствующих операторов, реализуемое при выполнении условия $U_i = 1$; U_i – конъюнкция соответствующих условий; h – количество членов в ДНФА.

ДНФА, у которой $R_1 = R_2 = \dots = R_h = n$, называется совершенной дизъюнктивной нормальной формой алгоритма (СДНФА).

Определение 1. Схемы алгоритмов и соответствующие им регулярные выражения называются эквивалентными, если они имеют одинаковые СДНФА.

Определение 2. Две АС называются Р-эквивалентными, если одна переходит в другую при изменении номеров (или обозначения) условных переменных.

На рис. 1 приведены примеры Р-эквивалентных схем алгоритмов.

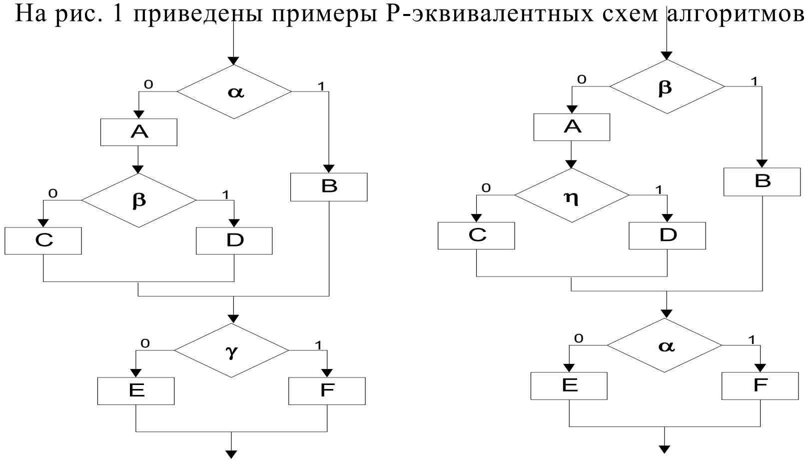

Рис. 1. Р-эквивалентные схемы

Определение 3. Две АС называются N-эквивалентными, если одна переходит в другую при изменении значений некоторых условных переменных на обратные.

На рис. 2 приведены примеры N-эквивалентных АС.

Определение 4. Две АС называются структурно-эквивалентными (S-эквивалентность), если они Р- и (или) N-эквивалентны и отличаются только видом линейных операторов на соответствующих ветвях условных операторов.

Для графического представления структурно-эквивалентных АС предлагается использовать графы специального вида, называемые далее В-графами.

Определение 5. В-графом называется ориентированный мультиграф без петель с множеством вершин $V = \{v_1, \dots, v_n\}$ и множеством ребер $Q = \{q_1, \dots , q_{2(n-1)}\}$, в котором выделены два полюса (начальная вершина v_1 и конечная вершина v_n), полустепень исхода всех вершин, кроме конечной равна двум и для любой вершины v_i ($i = 2, \dots , n - 1$) существует хотя бы одна простая цепь $v_1 \dots v_i \dots v_n$.

На рис. 3 приведена АС и соответствующий ей В-граф.

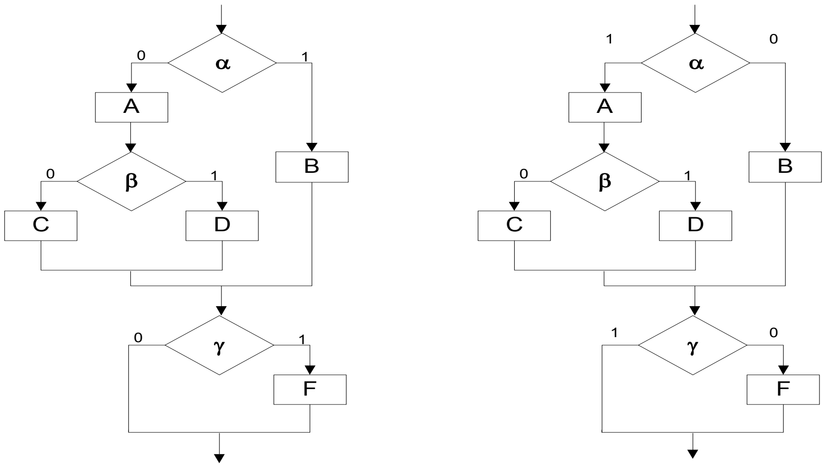

Рис. 2. N-эквивалентные схемы алгоритмов

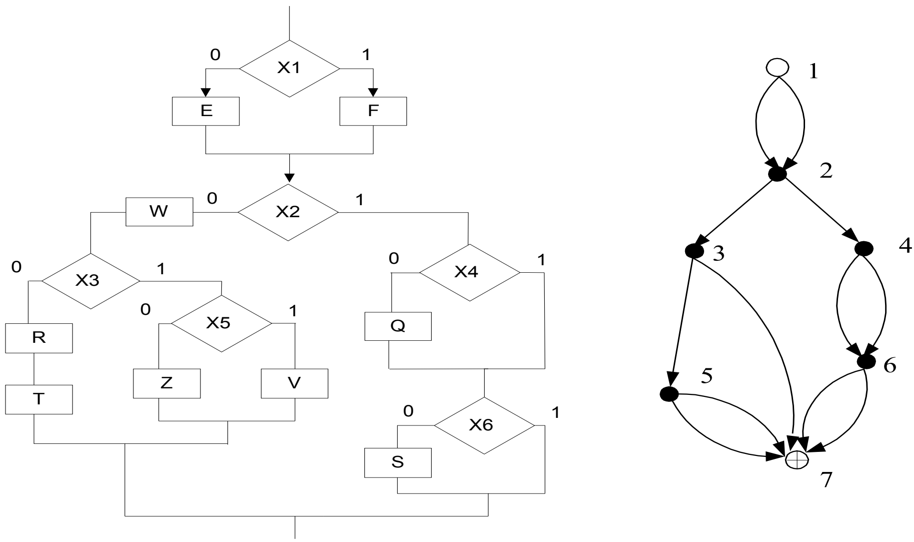

Рис. 3. Схема алгоритма и его графовая модель

Интерес к структурно-эквивалентным АС обусловлен тем, что каждому условному оператору соответствует переход, т.е. изменение последовательности выполнения команд в соответствии с алгоритмом программного обеспечения. Согласно статистике [8] 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.

Выводы. Условные переходы снижают общую производительность центрального процессора, так как в ожидании этого перехода конвейер работает вхолостую. Оптимизируя структуру алгоритма, можно повысить производительность центрального процессора при решении заданного класса задач. При рассмотрении типовых вариантов АС задача оптимизации структуры упрощается, так как необходимо выбрать эквивалентную АС не из всего множества АС, а только из множества типовых представителей, количество которых значительно меньше. Для этого необходимо определить их количество и построить каталоги типовых вариантов.

ЛИТЕРАТУРА

1. *Кучмиев В.Г., Чумаченко И.В. Алгоритмические, инструментальные и программные средства автоматизированных систем обработки информации и управления: Монография. – Х.: ХАИ, 2003. – 280 с.*
2. *Котов В.Е. Введению в теорию схем программ. – Новосибирск, 1978.– 184 с.*
3. *Luckham D.C. Park D.M., Patterson M.S. On Formalizad Computer Programs // Computer and System Sci. – 1970. – V. 4, № 3.– P. 34 – 43.*
4. *Летичевский А.А. Функциональная эквивалентность дискретных преобразователей // Кибернетика. – 1969. – № 2. – С. 5 – 15.*
5. *Янов Ю.И. О локальных преобразованиях схем алгоритмов // Проблемы кибернетики: Сб. статей. – М.: Наука, 1968. – Вып. 20. – С. 201 – 216.*
6. *Криницкий Н.А. Равносильные преобразования алгоритмов и программирование. – М.: Сов. радио, 1970. – 304 с.*
7. *Чумаченко И.В. Расширенная алгебра регулярных схем алгоритмов с коммутативными условиями // Авіаційно-космічна техніка і технологія. – Х.: Нац. аерокосміч. ун-т «ХАІ». – 2000. – Вип. 20. – С. 154 – 158.*
8. *Аппаратные средства PC / К. Айден, О. Колесниченко и др. – С.-Пб.: BHV-Санкт-Петербург, 1998. – 608 с.*

Поступила 17.08.2004

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.*

БУГАС Дмитрий Николаевич*, аспирант Национального аэрокосмического университета "ХАИ", который окончил в 2001 году. Область научных интересов – автоматизированные системы обработки информации и управления.*