УДК.681.142.2

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НЕКОТОРЫХ ЛОГИЧЕСКИХ АЛГОРИТМОВ УПРАВЛЕНИЯ

Е. Н. Казаков

(Москва)

Вычислительные среды и системы являются перспективными средствами технической реализации разнообразных алгоритмов. Представляет интерес сравнительное исследование различных практически используемых алгоритмов с целью выяснения характеристик, влияющих на их реализацию. В настоящем докладе рассматривается класс логических алгоритмов управления. Класс логических алгоритмов совпадает (по определению) с классом операторных алгоритмов [I], в котором помимо числовых и операторных переменных используются логические переменные, принимающие в качестве значений слова 0,I, а в качестве списка операций применяется любая функционально полная система функций алгебры логики, например: отрицание ($\neg$), дизъюнкция ($\vee$), конъюнкция ($\wedge$) и сложение по mod 2 ($\oplus$).

Система управления (СУ) реализует некоторое отображение множества значений входных переменных в множество значений выходных переменных. Операторный алгоритм, который реализует то же отображение, что и СУ, назовем алгоритмом управления.

Класс логических алгоритмов управления образуют алгоритмы управления, принадлежащие классу логических алгоритмов. Все алгоритмы из рассматриваемого класса можно представить в виде записи на языке стандартного представления логических элементов управления (язык СП) [2], являющемся расширенным подмножеством языка АЛГОЛ-60. Основные особенности этого языка состоят в следующем:

1. Все переменные делятся на 7 категорий, каждой из которых ставится в соответствие буква. Логические переменные делятся на независимые (X), промежуточные (F), результативные (R) и выходные (B). Различаются также переменные индексные (i), временные (t) и метки (M). Все переменные одной категории нумеруются в пределах алгоритма. Идентификатор переменной образуется из буквы, обозначающей категорию, и номера.

2. Список логических операций состоит из операций $\neg, \vee, \wedge, \oplus$.

3. Арифметические операции применяются только к индексным переменным.

4. Вводятся дополнительные синтаксические конструкции "словарь" и "анализ". В словаре дается содержательное описание всех независимых и выходных переменных алгоритма. В анализе отмечаются значения некоторых параметров (ранг, условный ранг, сложность выражений), описываемых ниже.

5. Полная запись алгоритма управления на языке СП состоит из словаря, анализа и формальной записи (ФЗ), выполняемой на языке АЛГОЛ-60 с учетом пунктов 1 - 3. ФЗ называется w -бесповторной, если не найдется ни одного логического выражения, содержащего операцию и входящего в запись более одного раза. Формальной записи можно взаимно-однозначно поставить в соответствие схему из функциональных элементов [3] в базисе ($\neg$, $\vee$, $\wedge$, $\oplus$, T), где T - элемент задержки. При этом каждому выражению*) соответствует древовидная схема, называемая элементарной. Всем операциям, переменным и выражениям в записи по индуктивному правилу приписывается ранг. Содержательно ранг соответствует максимальной длине цепи из элементов, соединяющей выход элемента, которому приписана данная операция, переменная, выражение, со входом схемы. Для каждого выражения выбирается такой порядок выполнения опе -

---

*) Здесь и далее под выражением понимается логическое выражение, составляющее правую часть оператора присваивания.

раций, который не изменяет значения выражения и при заданных рангах переменных доставляет минимум рангу выражения. Все операторы присваивания упорядочиваются по возрастанию ранга своих выражений.

Стандартной формой логического алгоритма называется упорядоченная по рангам w - бесповторная формальная запись. Исследованию подвергаются только стандартные формы логических алгоритмов. Результатом исследования каждого алгоритма являются значения следующих характеристик, вычисляемых по записи на языке СП.

I. Сложностью алгоритма (Ω) назовем количество операций из базиса ($\neg$, $\vee$, $\wedge$, $\oplus$, T), входящих в запись алгоритма. Очевидно, что эта характеристика отражает затраты на физическую реализацию алгоритма.

2. Количество логических переменных (N) в записи определяет нижнюю границу сложности алгоритма.

3. Распределение логических переменных по категориям задается вектором

$$\tilde{n} = (n_X, n_F, n_R, n_B),$$

причем

$$n_P = \frac{N_P}{N},$$

где P любая из букв X , F , R , B , а N_P - количество переменных категории P . Вектор $\tilde{n}$ позволяет судить об относительном объеме внутренней памяти (n_R), объеме рабочей памяти (n_F) и сравнивать сложность реализации самого алгоритма со сложностью реализации его внешних соединений ($n_F + n_R$ и $n_X + n_B$).

4. Распределение операций по типам задается вектором

$$\tilde{\omega} = (\omega_T, \omega_\neg, \omega_\vee, \omega_\wedge, \omega_\oplus),$$

причем

$$\omega_Q = \frac{\Omega_Q}{\Omega},$$

где Q любой из символов T, $\neg$, $\vee$, $\wedge$, $\oplus$, а Ω_Q - количество операций типа Q . Зная вектор $\tilde{\omega}$, можно судить об изменениях сложности алгоритма с изменением базиса.

5. Распределение переменных по степеням кратности задается вектором

$$\tilde{c} = (c_1, c_2, \ldots, c_k, \ldots),$$

причем
$$c_1 = \frac{\Omega - (N - N_X) + N(1)}{N_X + \Omega}; \; c_k = \frac{N(k)}{N_X + \Omega}, \quad k \geq 2,$$
где $N(k)$ – число переменных, имеющих степень кратности k. Степень кратности переменной равна числу разветвлений выхода того элемента, которому она приписана. Вектор $\tilde{c}$ позволяет оценить требуемую нагрузочную способность элементов.

6. В каждой элементарной схеме выделим подсхемы, обладающие свойствами: I) подсхема состоит из элементов одного типа и имеет один выход; 2) все входы и выход подсхемы соединяются с элементами другого типа.

Если выделенная подсхема со свойствами I), 2) содержит l элементов, то будем называть её операционным звеном длины l. Очевидно, что операционное звено длины l можно заменить многовходовым элементом того же типа с числом входов $l + 1$. Распределение операционных звеньев по длине задается вектором
$$\tilde{z} = (z_1, z_2, \ldots, z_l, \ldots),$$
причем
$$z_l = \frac{Z(l)}{\sum\limits_{l=1}^{\Omega} Z(l)},$$
где $Z(l)$ – число операционных звеньев длины l.

Вектор $\tilde{z}$ задает исходную информацию для выбора количества входов элементов, с помощью которых реализуется данный алгоритм.

7. Распределение выходных и результативных переменных по рангам задается вектором
$$\tilde{p} = (p_1, p_2, \ldots, p_r, \ldots),$$
причем
$$p_r = \frac{P(r)}{N_R + N_B},$$
где $P(r)$ – число выходных и результативных переменных ранга r. Ранг переменной задает в некоторых единицах задержку, с которой вычисляется значение переменной, а также максимальное число последовательно соединенных элементов в цепочке, вычисляющей это значение. Поэтому вектор $\tilde{p}$ позволяет оценивать количество вспомогательного оборудования для усиления сигналов и среднюю задержку при выполнении алгоритма.

8. Длиной алгоритма назовем число D , которое равно максимальному рангу. Длина алгоритма отражает максимальное запаздывание при выполнении алгоритма.

9. Распределение операций по рангам задается вектором

$$\tilde{\rho} = (\rho_1, \rho_2, \ldots, \rho_r, \ldots),$$

причем

$$\rho_r = \frac{\Omega(r)}{\Omega},$$

где $\Omega(r)$ - количество операций ранга r . Вектор $\tilde{\rho}$ позволяет оценить достижимую степень параллельности выполнения алгоритма.

10. Шириной алгоритма назовем число $S = \max\limits_{r \le D} \Omega(r)$.
S задает объем памяти, необходимой для хранения промежуточных результатов при минимальной задержке в выполнении алгоритма.

11. Условным рангом выражения назовем ранг выражения, вычисленный в предположении, что ранги всех переменных выражения равны нулю. Распределение выражений по условным рангам задается вектором

$$\tilde{u} = (u_1, u_2, \ldots, u_r, \ldots),$$

причем

$$u_r = \frac{U(r)}{N - N_X},$$

где $U(r)$ - количество выражений, имеющих условных ранг r .

12. Сложностью выражения назовем количество вхождений в выражение символов операций. Распределение выражений по сложности задается вектором

$$\tilde{w} = (w_1, w_2, \ldots, w_r, \ldots),$$

причем

$$w_r = \frac{W(r)}{N - N_X},$$

где $W(r)$ - количество выражений со сложностью r .

Разбиение формальной записи на выражения соответствует разбиению схемы на подсхемы такие, что все разветвляющиеся соединения расположены вне данных подсхем. По векторам $\tilde{u}$ и $\tilde{w}$ можно судить о типичной для данного алгоритма сложности таких подсхем, о средней времени задержки, вносимой подсхемой и степени разветвленности схемы алгоритмов.

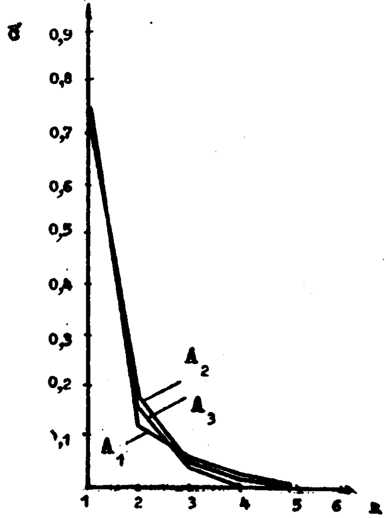

Рис. I

Распределение переменных по степеням кратности

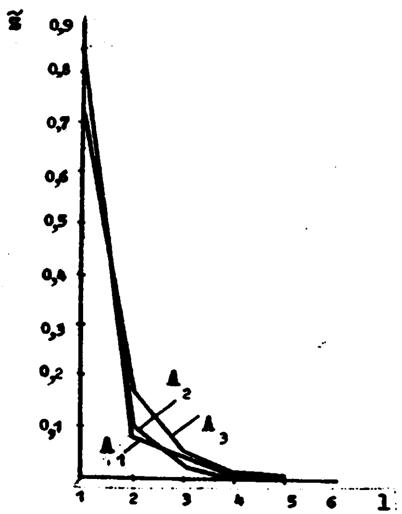

Рис.2.

Распределение операционных звеньев по длине.

Таким образом, в результате исследования каждому алгоритму A сопоставляется совокупность характеристик в виде матрицы

$$M(A)=(\Omega,N,D,S,\tilde{n},\tilde{\omega},\tilde{c},\tilde{z},\tilde{p},\tilde{\rho},\tilde{u},\tilde{w}).$$

По векторам $\tilde{c},\tilde{z},\tilde{p},\tilde{\rho},\tilde{u},\tilde{w}$ можно вычислить средние значения: степени кратности переменной (k_{cp}), длины операционного звена (l_{cp}), длины схемы (d_{cp}), ширины схемы (s_{cp}), длины выражения (u_{cp}), сложности выражения (w_{cp}). Менее громоздко, но более грубо каждый алгоритм A можно охарактеризовать вектором $V(A) = (\Omega, N, D, S, n_X, n_F, n_R, n_B, \omega_T, \omega_{\neg}, \omega_{\vee}, \omega_{\wedge}, \omega_{\oplus}, k_{cp}, l_{cp}, d_{cp}, s_{cp}, u_{cp}, w_{cp})$.

В таблице и на рис.I–4 приводятся некоторые характеристики, полученные при исследовании реальных, разработанных в ЦНИИКА, алгоритмов управления технологическими процессами. A_1 – алгоритм управления механическими фильтрами цеха химводоподготовки ТЭЦ, A_2 –алгоритм управления химическими фильтрами того же цеха, A_3 –алгоритм пуска-останова котлоагрегата ТЭЦ.

Значения величин Ω и N свидетельствуют о том, что рассматриваемые алгоритмы являются достаточно сложными. Наиболее интересным результатом исследования является факт сходства по некоторым характеристикам столь различных по назначению алгоритмов. Все алгоритмы почти не содержат операций $\oplus$ и содержат одинаковую, причем незначительную до-

Таблица

	$V(A_1)$	$V(A_2)$	$V(A_3)$
Ω	1148	2094	902
N	476	929	6[illegible]6
D	25	26	30
S	154	205	256
n_X	0,32	0,38	0,42
n_F	0,42	0,39	0,14
n_R	0,11	0,08	0,07
n_B	0,15	0,15	0,37
ω_T	0,01	0,00	0,08
$\omega_\neg$	0,12	0,11	0,11
$\omega_\vee$	0,42	0,40	0,20
$\omega_\wedge$	0,45	0,49	0,59
$\omega_\oplus$	0	0	0,02
k_{cp}	1,53	1,52	1,51
l_{cp}	1,34	1,34	1,54
d_{cp}	11,4	12,0	5,8
s_{cp}	46,0	80,6	30,0
u_{cp}	2,4	2,2	2,1
w_{cp}	3,6	3,6	2,7

лю операции , что свидетельствует о нецелесообразности расширения базиса. Интересно также, что вектора $\tilde{c}$ и $\tilde{z}$ (рис.1.2) для всех алгоритмов почти совпадают, причем наибольшими по значению являются компоненты с малыми номерами, что показывает целесообразность применения элементов с небольшим числом входов и небольшой нагрузочной способностью. Вектора $\tilde{u}$ и $\tilde{w}$ (рис.3,4) показывают, что схемы всех алгоритмов разбиваются на большое количество относительно несложных подсхем, причем с ростом сложности доля подсхем уменьшается.

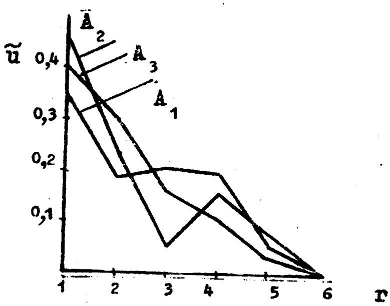

Рис. 3
Распределение выражений по условным рангам.

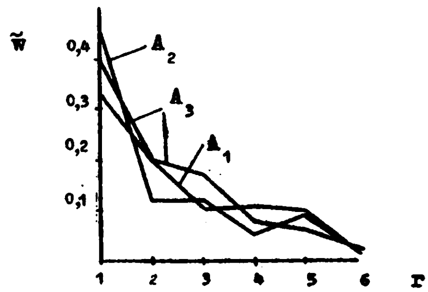

Рис. 4
Распределение выражений по сложности.

Выводы

Запись алгоритма на алгоритмическом языке является математическим эквивалентом любой реализации, поэтому рассмотренные характеристики отражают собственные свойства алгоритма и могут быть использованы при определении требований к элементам и устройству, реализующему алгоритм.

Если за меру близости алгоритмов принять расстояние между векторами характеристик, то рассмотренные характеристики можно использовать для сравнения и классификации реальных алгоритмов.

Литература

1. Ершов А.П. Операторные алгорифмы I. Проблемы кибернетики, вып. 3, М., Физматгиз, 1960.
2. Казаков Е.Н. О стандартном представлении логических алгоритмов управления. Труды ЦНИИКА, вып.20 (в печати).
3. Лупанов О.Б. О синтезе некоторых классов управляющих систем Проблемы кибернетики, вып.10. М.,Наука, 1963 .