УДК 618.142.2+044.942

С.Ю. МЕЛЕШЕНКО, А.Р. ЕМАД, А.Н. МАРЧЕНКО

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

МИНИМИЗАЦИЯ КВАЗИРЕГУЛЯРНЫХ СТРУКТУР АЛГОРИТМОВ СЛОЖНЫХ ПРОЕКТОВ

Сформулированы и решены задачи минимизации структур алгоритмов управления сложными проектами. Разработаны методики минимизации структур по основным критериям. Приведены примеры применения разработанных алгоритмов.

минимизация алгоритмов, критерий минимизации, алгоритмические алгебры, автоматизированная разработка алгоритмов

Введение

При конструировании программ сложных проектов задачи решаются, как правило, на уровне схем алгоритмов, поскольку именно для них разработана развитая техника тождественных преобразований [1,2]. Однако правила применения этих тождеств формализованы недостаточно, так как отыскание эффективной стратегии управления в задачах оптимизации алгоритмов является сложным процессом, несмотря на возможность использования моделей на основе искусственного интеллекта. При этом в роли базы данных может выступать совокупность оптимизируемых алгоритмов, в роли систем продукций - системы аксиом, а в роли системы управления - специальный проблемно-ориентированный механизм, обеспечивающий целенаправленный выбор аксиом, при котором перебор вариантов решений сводится к минимуму. Разработанная система оптимизации отражает сущность свойств модифицированной системы алгоритмических алгебр (МСАА) [3]. К возможным критериям оптимизации алгоритмов при соответствующих ограничениях относятся минимальное время реализации и минимальное число:

– логических условий с заданным или произвольным распределением сдвигов;

– логических условий с учетом неиспользуемых наборов значений логических переменных;

– операторов;

– операционных устройств и ячеек памяти, используемых при выполнении программ и др.

Формулирование целей статьи

В общем случае задача оптимизации программ является многокритериальной. Качество решения этой задачи характеризуется числом проводимых при этом промежуточных преобразований.

Исходными формами представления алгоритмов служат дизъюнктивные формы (ДФ) Мили и Мура, для которых имеет место условие $\alpha \in \{0,1\}$, что не снижает практической ценности результатов. Таким образом, необходимо разработать алгоритмы оптимизации структур алгоритмов задач управления сложными проектами по рассмотренным выше критериям [4].

Решение проблемы

Формальная постановка задачи *минимизации алгоритмов по числу операторов и логических условий с произвольным распределением сдвигов* состоит в следующем. Пусть задана ДФ алгоритма $\overline{S_L}$. Требуется построить процедуру минимизации для приведения $\overline{S_L}$ к такому виду $\overline{S_L}^{\min}$ с минимальным числом логических условий x и операторов y, при котором

$$C^{*} = \min C(n_1(x), n_2(y)) \mid \nu(\overline{S_L}^{\min}) = \nu(\overline{S_L}), \quad (1)$$

где $n_1(x)$, $n_2(y)$ - числа x, y; C - функционал сложности алгоритма; $\nu(\overline{S_L})$ - функция алгоритма.

Для формализации процедуры минимизации введем регулярную операцию пересечения уравнений ДФ, которую обозначим через $\underset{r}{\cap}$ в отличие от обычной операции пересечения множеств. Пусть R, Q - отдельные уравнения ДФ алгоритма, причем $S \in R$, $S \in Q$, а P - символ, используемый для обозначения выражения S после применения к уравнениям операции $\underset{r}{\cap}$. Тогда

$$P = R \underset{r}{\cap} Q = Q \underset{r}{\cap} R = S;\ R = R(P);\ Q = Q(P).$$

Определение 1. Равносильные операторы, входящие в различные уравнения ДФ, называются подобными членами.

Определение 2. Минимизация отдельных уравнений ДФ алгоритмов в соответствии с критерием (1) является приведением подобных членов и называется локальной.

Определение 3. Минимизация различных уравнений ДФ алгоритмов с применением операции $\underset{r}{\cap}$ на основании критерия (1) называется совместной.

Сформулируем в виде теорем условия, при которых возможны совместная и локальная минимизации алгоритмов.

Теорема 1. Пусть $P_{ij} = P \underset{ir}{\cap} P_j$ - оператор α-дизъюнкции. Совместная минимизация алгоритмов возможна в том, и только в том случае, если $P_{ij} \neq N$ $(i \neq j; i, j = \overline{1, g})$.

Теорема 2. Если в отдельном уравнении ДФ имеются подобные члены, то возможна локальная минимизация алгоритмов.

Теорема 3. Пусть $P_{ij} = P \underset{ir}{\cap} P_j = y_g P_g$. Совместная минимизация алгоритмов по числу операторов возможна только в том случае, если $P_{ij} \neq N$ $(i \neq j; i, j = \overline{1, g})$.

Доказательство теорем не очевидно.

Кроме приведенных выше соотношений на первом этапе минимизации следует использовать тождества [5]

$$\underset{\alpha}{(}P \vee \underset{\alpha}{(}R \vee Q)) = \underset{\alpha}{(}P \vee Q); \quad (2)$$

$$\underset{\alpha\alpha}{((}P \vee R) \vee Q) = \underset{\alpha}{(}P \vee Q), \quad (3)$$

с учетом которых алгоритм минимизации ДФ формулируется следующим образом.

Методика решения задачи минимизации структур алгоритмов по числу операторов и логических условий с произвольным распределением сдвигов реализуется в следующем виде.

Алгоритм 1. Минимизация алгоритмов в соответствии с критерием (1).

Вход. ДФ алгоритма.

Выход. Минимальная ДФ алгоритма.

Метод. Локальная и совместная минимизации ДФ.

Шаг 1. Если в уравнениях ДФ имеются операторы вида (2), (3), осуществить локальную минимизацию алгоритма [1].

Шаг 2. Если в различных уравнениях ДФ содержатся подобные члены, на основании операции $\underset{r}{\cap}$ произвести совместную минимизацию ДФ, начиная с операторов α-дизъюнкции, имеющих минимальную дизъюнктивную глубину.

Шаг 3. Если в одних и тех же уравнениях ДФ содержатся подобные члены, осуществить локальную минимизацию алгоритмов на основе тождеств

$$\underset{\alpha}{(}R \vee Q) = \underset{\alpha}{(}Q \vee R);$$

$$\underset{\alpha\beta}{((}R \vee Q) \vee \underset{\beta}{(}P \vee S)) = \underset{\beta\alpha}{((}R \vee P) \vee \underset{\alpha}{(}Q \vee S));$$

$$\underset{\alpha}{(}P\vee\underset{\beta}{(}Q\vee R)=\underset{\alpha\vee\beta}{(}\underset{\alpha}{(}P\vee Q)\vee R);$$

$$\underset{\alpha}{(}\underset{\beta}{(}P\vee Q)\vee R)=\underset{\alpha\beta}{(}P\vee\underset{\alpha}{(}Q\vee R)).$$

Шаг 4. Если в различных уравнениях ДФ имеются подобные члены вида y_qP_q, на основании операции $\underset{r}{\cap}$ произвести совместную минимизацию алгоритмов по числу операторов.

Шаг 5. Конец.

<u>Пример 1.</u> Найти минимальную ДФ алгоритма исходя из ДФ Мили. Запишем:

$$P_1=\underset{x_1}{(}y_2P_2\vee\underset{x_6}{(}\underset{x_3}{(}y_2P_2\vee y_4P_5)\vee\underset{x_4}{(}y_4P_5\vee$$
$$\vee\underset{x_1}{(}y_{10}P_2\vee\underset{x_7}{(}y_7P_1\vee y_6P_1)))));$$

$$P_2=\underset{x_3}{(}\underset{x_2}{(}\underset{x_7}{(}y_5P_4\vee y_{10}P_2)\vee y_3P_3)\vee\underset{x_2}{(}y_5P_4\vee$$
$$\vee\underset{x_4}{(}y_4P_5\vee\underset{x_5}{(}y_9P_6\vee\underset{x_1}{(}y_{10}P_2\vee\underset{x_7}{(}y_7P_1\vee y_6P_1))))));$$

$$P_3=\underset{x_6}{(}\underset{x_7}{(}y_7P_1\vee y_6P_1)\vee y_{10}P_2);\tag{4}$$

$$P_4=\underset{x_8}{(}\underset{x_1}{(}y_{10}P_2\vee\underset{x_7}{(}y_7P_1\vee y_6P_1))\vee\underset{x_4}{(}y_4P_5\vee$$
$$\vee\underset{x_5}{(}y_9P_6\vee\underset{x_1}{(}y_{10}P_2\vee\underset{x_7}{(}y_7P_1\vee y_6P_1)))));$$

$$P_5=\underset{x_1}{(}y_{10}P_2\vee\underset{x_7}{(}y_7P_1\vee y_6P_1));$$

$$P_6=y_1P_7;$$

$$P_7=\underset{x_3}{(}\underset{x_4}{(}y_8P_5\vee y_{10}P_2)\vee P_7).$$

Графическая схема алгоритма, по которой получена эта ДФ, взята из работы [6]. В соответствии с алгоритмом 1:

$$P_1=\underset{x_1\vee x_6x_2}{(}y_2P_2\vee\underset{x_4\vee x_6}{(}P_{10}\vee P_5));$$

$$P_2=\underset{x_2}{(}\underset{x_3\vee x_7}{(}y_5P_4\vee y_{10}P_2)\vee\underset{x_2}{(}y_3P_3\vee P_8));$$

$$P_2=\underset{x_2}{(}\underset{x_3\vee x_7}{(}y_5P_4\vee y_{10}P_2)\vee\underset{x_2}{(}y_3P_3\vee P_8));$$

$$P_3=\underset{x_6}{(}P_9\vee P_{11});$$

$$P_4=\underset{x_8}{(}P_5\vee P_8);$$

$$P_5=\underset{x_1}{(}P_{11}\vee P_9);$$

$$P_6=y_1P_7;$$

$$P_7=\underset{x_3}{(}\underset{x_4}{(}y_8P_5\vee P_{11})\vee P_7);\tag{5}$$

$$P_8=\underset{x_4}{(}P_{10}\vee\underset{x_3}{(}y_9P_6\vee P_5));$$

$$P_9=\underset{x_7}{(}y_7P_1\vee y_6P_1);$$

$$P_{10}=y_4P_5;$$

$$P_{11}=y_{10}P_2.$$

Из примера следует, что проводимые на основе операции $\underset{r}{\cap}$ преобразования затрагивают лишь отдельные фрагменты структуры ДФ, поскольку они представлены в скобочной форме. Узкоцелевой является также стратегия приведения подобных членов при локальной минимизации.

Формальная подстановка задачи *минимизации алгоритмов по числу операторов и логических условий с заданным распределением сдвигов* состоит в следующем. Пусть $B(Y)=r(\overline{S}_L)$ - отношение распределения сдвигов, задаваемое для совокупности операторов *Y* ДФ алгоритма $\overline{S}_L$. Требуется построить процедуру минимизации для приведения $\overline{S}_L$ к такому виду $\overline{S}_L{}^{\min}$, при котором

$$C^*=\min_{x,y}C(n_1(x),n_2(y))\Bigg|_{\substack{B(Y)=r(\overline{S}_L);\\ \nu(\overline{S}_L{}^{\min})=\nu(\overline{S}_L)}},\tag{6}$$

где $n_1(x)$, $n_2(y)$, $\nu(\overline{S}_L)$, C соответствуют (1).

В МСАА учет распределения сдвигов производится на основе соотношения

$$R\underline{\alpha}\to\underline{\alpha}\cdot R.\tag{7}$$

Иными словами, операторы *R* и α коммутативны, если выполняется условие $\overline{\alpha}\in B(R)$, где $B(R)$ - распределение сдвигов оператора *R*.

При решении данной задачи применяются следующие тождества:

$$\underline{\alpha}\underline{\overline{\alpha}}=N;\ \ \underline{\alpha}\underline{\beta}=\underline{\beta}\underline{\alpha};\ \ \underline{\alpha}\underline{\alpha}=\underline{\alpha};\ \ \underline{\alpha}(\underline{\alpha}\vee\underline{\beta})=\underline{\alpha};$$

$$NQ = QN = N; \;\; Q \vee N = Q; \;\; EQ = QE = Q; \qquad (8)$$

$$\underset{\alpha}{\{P\}} = \underline{\alpha}\, \underset{\alpha}{\{P\}}^{+} \vee \underline{\overline{\alpha}}.$$

Методика решения задачи минимизации структур алгоритмов по числу операторов и логических условий с произвольным распределением сдвигов состоит в следующем.

Алгоритм 2. Минимизация алгоритмов в соответствии с критерием (6).

Вход. ДФ алгоритма, для которой задано отношение $B(Y) = r(\overline{S}_L)$.

Выход. Минимальная квазирегулярная структура алгоритма (КСА).

Метод. Решение системы уравнений в МСАА с проведением тождественных преобразований.

Шаг 1. Если в уравнениях P_i имеются циклы, применить к ним правила

$$\frac{LRP = L(\alpha SR \vee Q)P}{LRP = L\,(\underset{\alpha}{\{S\}^{+}} \vee Q)P}, \qquad \frac{LRP = L(S\underline{\alpha}R \vee Q)P}{LRP = L(\underset{\alpha}{\{S\}^{+}} \vee Q)P}. \qquad (9)$$

Шаг 2. Положить $d = 1$, где d - индекс уравнения.

Шаг 3. Подсчитать в ДФ ранги правых нетерминалов.

Шаг 4. Выбрать нетерминал P_i, содержащий $V_i^{\min}$. Если в ДФ ранги различных нетерминалов совпадают, то выбор нетерминала P_i из них осуществляется произвольно.

Шаг 5. Подставить в правые части системы уравнений вместо нетерминала P_i соответствующее ему выражение.

Шаг 6. Произвести в задействованных на шаге 5 уравнениях тождественные преобразования, учитывая распределение сдвигов, и применить к ним при наличии циклов одно из правил (9).

Шаг 7. Положить $d = d + 1$. Если $d < g$, то перейти к шагу 3, в противном случае – к шагу 8.

Шаг 8. Конец.

<u>Пример 2.</u> Найти минимальную КСА для ДФ Мура [4]. Записываем:

$$P_1 = \underline{x_1}\underline{x_2}y_1P_2 \vee \underline{x_1}y_3P_4 \vee \underline{\overline{x_1}}\underline{x_2}; \;\; P_2 = y_2P_3;$$

$$P_3 = \underline{\overline{x_1}}y_3P_4 \vee \underline{x_1}; \;\; P_4 = \underline{x_1}y_2P_3 \vee \underline{\overline{x_1}}.$$

При заданном распределении сдвигов имеем

$$B(y_1) = \{x_2\}; \, B(y_2) = \{x_2\}; \, B(y_3) = \{x_2\}.$$

На основании алгоритма 2 получаем

$$P_1 = \underset{x_1}{(}\, y_3 \vee \underset{x_2}{(}\, y_1y_2 \vee E)).$$

По аналогии с алгоритмами 1, 2 могут быть разработаны правила *минимизации по другим критериям.* Например, пусть даны ДФ алгоритма $\overline{S}_L$ и множество неиспользуемых наборов значений логических переменных $H = \{\Delta f \,/\, f = \overline{1, F}\}$. Требуется привести S_L к такой минимальной форме $\overline{S}_L^{\min}$, при которой

$$C^{*} = \min_{x,y} C(n_1(x), n_2(y))\Big|_{\substack{\nu(\overline{S}_L^{\min}) = \nu(\overline{S}_L); \\ H = \{\Delta f / f = \overline{1,F}\}}}, \qquad (10)$$

где $n_1(x)$, $n_2(y)$, $\nu(\overline{S}_L)$, C соответствуют (1).

При использовании ДФ алгоритмов для минимизации КСА исходим из того, что $\alpha \in \{0,1\}$. Следовательно, если условие α в операторе $\underset{\alpha}{(}\, R \vee Q)$ не определено на наборе $\Delta_1 = \langle 0 \rangle$, то $\underset{\alpha}{(}\, R \vee Q) = \underset{1}{(}\, R \vee N) = R$.

Аналогично, если в этом же операторе условие α не определено на наборе $\Delta_2 = \langle 1 \rangle$, то $\underset{\alpha}{(}\, R \vee Q) = \underset{0}{(}\, N \vee Q) = Q$.

Наконец, возможен случай, когда условие α в операторе $\underset{\alpha}{(}\, R \vee Q)$ не определено ни на одном из наборов $\Delta_1 = \langle 0 \rangle$, $\Delta_2 = \langle 1 \rangle$. Тогда $\underset{\alpha}{(} R \vee Q) = \underset{\mu}{(} R \vee Q) = N$.

Специфика ДФ алгоритмов при решении данной задачи позволяет учитывать не весь используемый набор значений логических переменных, а лишь ту его часть, которая касается требуемого фрагмента ДФ.

Пусть, например, для уравнения [5]

$$\begin{aligned} P_2 = \underset{\bar{x}_1}{(} y_6 \vee \underset{x_2}{(} \underset{x_3}{(} y_2P_2 \vee y_3P_3) \vee \\ \vee \underset{x_3}{(} y_4P_4 \vee y_5P_5))) \end{aligned} \tag{11}$$

заданы неиспользуемые наборы значений логических условий, которым соответствуют неиспользуемые конъюнкции

$$\begin{aligned} \varphi_1 &= x_1\bar{x}_2x_3, \\ \varphi_2 &= x_1\bar{x}_2\bar{x}_3, \\ \varphi_3 &= x_1\bar{x}_2. \end{aligned} \tag{12}$$

В (10) оператор y_4P_4 должен выполняться на основе конъюнкции $\varphi_1 = x_1\bar{x}_2x_3$, оператор y_5P_5 - на основе конъюнкции $\varphi_2 = x_1\bar{x}_2\bar{x}_3$. Поскольку эти конъюнкции являются неиспользуемыми, условие x_3 относится к неопределенным.

Следовательно, $\underset{x_3}{(} y_4P_4 \vee y_5P_5) = N$.

В результате уравнение (10) принимает вид

$$P_2 = \underset{\bar{x}_1}{(} y_6 \vee \underset{x_2}{(} \underset{x_3}{(} y_2P_2 \vee y_3P_3) \vee N)),$$

где конъюнкция $\varphi_3 = x_1\bar{x}_2$ является неиспользуемой. Отсюда

$$\begin{aligned} P_2 = \underset{x_1}{(} y_6 \vee \underset{1}{(} \underset{x_3}{(} y_2P_2 \vee y_3P_3) \vee N)) = \\ = \underset{x_1}{(} y_6 \vee \underset{x_3}{(} y_2P_2 \vee y_3P_3)). \end{aligned} \tag{13}$$

Таким образом, если задана ДФ алгоритма с неиспользуемыми наборами значений логических переменных, то необходимо во всех уравнениях ДФ выявить наборы значений логических условий, проверить их соответствие неиспользуемым наборам и произвести минимизацию ДФ алгоритма с учетом рассмотренных особенностей.

Преобразование КСА в автоматную форму (АФ) алгоритма соответствует задаче *абстрактного синтеза дискретного преобразователя* (ДП) и состоит в следующем.

Пусть задана КСА R в алфавитах $Y = \{y_f \,/\, f = \overline{1,F}\}$ (базисные операторы) и $X = \{x_m \,/\, m = \overline{1,M}\}$ (базисные условия). Алфавиты входят в конечную систему образующих $\Sigma_2 = \langle X, Y \rangle$ МСАА. Требуется построить управляющий автомат A, который при любой интерпретации базисных переменных представлял бы собой оператор, заданный этой КСА. Искомый автомат должен действовать на информационном множестве B_1, его элементарные операторы должны совпадать с операторами $y_f \in Y$, а входные сигналы – с условиями $x_m \in X$ [5].

Задача абстрактного синтеза ДП соответствует реализации отображения

$$\varphi_1 : R(Gr) \to A(Gr), \tag{14}$$

где $R(Gr)$ - множество КСА; $A(Gr)$ - множество ДФ алгоритмов. После решения этой задачи легко организуется перевод АФ в языки низкого уровня.

Как известно, КСА в связи с наличием циклов имеют более сложный и развитой язык описания алгоритмов, чем язык операторной схематологии алгоритмов. Этим объясняется необходимость в разработке специальных процедур для решения задачи синтеза, которая возникает, например, на этапе отладки алгоритмов.

При исследовании объектов алгоритмической природы зачастую встречаются обратные задачи, называемые обратимыми. К их числу относятся задачи анализа и абстрактного синтеза ДП. Установление для таких задач обратимых процедур на основе соответствующих взаимосвязей позволяет существенно сократить трудоемкость решения задач, поскольку наряду с эвристическими рассуждениями при этом оказывается возможным привлечение элементов формализма.

Рассмотрим правила нахождения АФ Мили в классе МСАА, используя модель детерминированного ДП последовательного действия.

При употреблении для этих целей квазирегулярной формы алгоритма нужно предварительно преобразовать их в КСА либо воспользоваться аналогичными алгоритмами в МСАА, модифицировав их для квазирегулярных форм. Последняя задача не вызывает принципиальных затруднений, если учесть связь между термами в квазирегулярной форме и операторами в КСА.

В МСАА КСА имеет вид

$$R_j = \underset{j}{V} Q_j P_j \vee \underset{j}{V} S_i , \qquad (15)$$

где Q_j, S_i - комбинация линейной и распределительной структур; P_j - циклическая структура (возможно также соотношение $Q_j = E$); $i, j = 1,\ 2, \ldots$.

Разметка КСА при построении АФ Мили производится по следующим правилам:

$$R = \underset{j}{V} \underset{a_1}{|}\ Q\ \underset{a_j}{|}\ P_j\ \underset{a_j}{|} \vee \underset{i}{V} \underset{a_1}{|}\ S_i\ \underset{a_{g+1}}{|} , \qquad (16)$$

причем предполагается, что P_j является левой α-итерацией.

Рассмотрим следующий алгоритм.

Алгоритм 3. Построение АФ Мили по заданной КСА.

Вход. КСА.

Выход. АФ автомата Мили.

Метод. Разметка исходного алгоритма и построение системы уравнений.

Шаг 1. Поставить в КСА вида (15) отметки состояний по следующим правилам:

Шаг 1.1. Отметить символами a_1, a_{g+1} начальное и заключительное состояния КСА.

Шаг 1.2. Отметить символом a_j входы и выходы α-итераций.

Шаг 1.3. Отметить символами состояния КСА с учетом предыдущих шагов.

Шаг 2. Построить по отмеченной КСА АФ Мили путем выделения отдельных уравнений.

Шаг 3. Конец.

<u>Пример 3.</u> Построить АФ Мили по КСА

$$P_1 = \underset{x_1}{\{}\ y_1 \mathrm{A} \Pi_2\ \underset{x_2}{(}\ y_2)\}^{+} \vee \Pi_2\ \underset{x_2}{\{}\ y_3 \Pi_3\ \underset{x_3}{(}\ y_4)\}^{+} \vee$$

$$\vee\ \underline{x_1} y_6 \vee \Pi_3 \underline{\overline{x_3}} y_5 ,$$

где П - указатель перехода.

На основании алгоритма 3 осуществляем следующие операции:

1. Производим разметку состояний в исходной КСА:

$$P_1 = \underset{a_1}{|}\ \underset{x_1}{\{}\ y_1\ \underset{a_2}{|}\ \Pi_2\ \underset{a_2}{|}\ \underset{x_2}{(}\ y_2)\}\ \underset{a_1}{|}^{+} \vee$$

$$\vee\ \underset{a_2}{|}\ \Pi_2\ \underset{a_2}{|}\ \underset{x_2}{\{}\ y_3\ \underset{a_3}{|}\ \Pi_3\ \underset{a_3}{|}\ \underset{x_3}{(}\ y_4)\}\ \underset{a_2}{|}^{+} \vee .$$

$$\vee\ \underset{a_1}{|}\ x_1 y_6\ \underset{a_4}{|} \vee \underset{a_3}{|}\ \Pi_3\ \underset{a_3}{|}\ x_3 y_5\ \underset{a_4}{|} .$$

2. Строим АФ Мили:

$$P_1 = \underset{x_1}{(}\ y_6 \vee y_1 P_2);$$

$$P_2 = \Pi_2\ \underset{x_1}{(}\ y_2 P_1 \vee y_3 P_3);$$

$$P_3 = \Pi_3\ \underset{x_3}{(}\ y_4 P_2 \vee y_5).$$

В общем случае разметка состояний в КСА не всегда приводит к минимизации, так как при этом определяется АФ Мили с избыточным числом состояний. Цель не достигается тогда, когда не заботятся об эффективности разметки. Чтобы снизить трудоемкость последней, иногда ограничиваются идентификацией входов и выходов в циклах и в отдельных линейных и распределительных структурах. В то же время не учитываются особенности переходов в КСА, связанные с функционированием детерминированных ДП и эквивалентностью их состояний.

Определение 4 [7]. Состояния a_i, a_j ДП A называются эквивалентными, если они порождают одно и то же квазирегулярное выражение.

Определение 5 [8]. Состояния a_i, a_j ДП A называются k-эквивалентными, если реакции ДП в этих состояниях совпадают для всех слова длины k.

Определение 6 [8]. Если состояния a_i и a_j ДП A являются k-эквивалентными, то они называются k-неразличимыми, в противном случае – k-различимыми.

Таким образом, если в построенной АФ имеются два одинаковых уравнения с различными левыми нетерминалами P_i, P_j, то состояния a_i и a_j эквивалентны, в связи с чем к системе уравнений можно применить подстановку $t = \begin{pmatrix} P_i \\ P_j \end{pmatrix}$ и исключить после этого бывшее уравнение P_j из рассмотрения.

Наконец, если в уравнении ДФ Мили P_1 имеются состояния a_2, a_3, a_5, в которые осуществляются переходы под воздействием одинаковых входных сигналов с выдачей одних и тех же операторов, то эти состояния эквивалентны, в связи с чем к системе уравнений можно применить подстановку $t = \begin{pmatrix} P_2 & P_2 \\ P_3 & P_5 \end{pmatrix}$ с последующими преобразованиями.

Синтез АФ Мура и минимизации в них числа состояний производятся аналогичным образом с учетом особенностей разметки состояний в графической схеме алгоритма в муровской интерпретации.

Заключение

Алгоритмические алгебры являются эффективным инструментарием в области автоматизированной разработки алгоритмов и программ сложных проектов. Они обладают достаточно выразительными языками высокого уровня, которые сравнительно просто анализируются в соответствующих системах компиляции.

В дальнейших исследованиях предполагается разработать электронный компилятор для реализации любых алгоритмов управления сложными проектами.

Литература

1. Жихарев В.Я. Технологически ориентированное проектирование алгоритмов и программ // Радіоелектронні і комп'ютерні системи. – 2003. - № 1. – С. 73 – 77.
2. Мелешенко С.Ю., Шилова Т.В., Жихарева А.В. Принятие управляющих решений в промышленно-сбытовой системе // Технология приборостроения. – 2001. - № 1-2. – С. 3 – 8.
3. Жихарев В.Я., Илюшко В.М., Чумаченко И.В. Проектирование электронных компиляторов. - Х.: Факт, 1999. – 144 с.
4. Иванов П.М. Автоматные регулярные микропрограммы // Кибернетика. – 1973. - №1. - С. 21-30.
5. Торчило В.Н., Емад А.Р., Хилал М.Н. Абстрактный синтез конечных автоматов Мили в алгебре квазирегулярных событий // Открытые информационные и компьютерные интегрированные технологии. – Харьков: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т». – 2003. – Вып. 21. – С.185-191.
6. Баранов С.И. Синтез микропрограммных автоматов. - Л.: Энергия, 1974. – 79 с.
7. Лазарев В.Г., Пийль Е.И. Синтез управляющих автоматов. - М.: Энергия, 1970. – 490 с.
8. Дьяченко В.Ф., Лазарев В.Г. Способ упрощения логических схем алгоритмов, учитывающий неиспользуемые показы значений логических переменных // Проблемы передачи информации. – 1966. - Вып. 3. – С. 92-96.

Поступила в редакцию 9.01.04

Рецензент: д-р техн. наук, проф. В.Я. Жихарев, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», г. Харьков