УДК 681.322

Я.В. ИЛЮШКО

Национальный аэрокосмический университет им Н.Е. Жуковского "ХАИ", Украина

СИНТЕЗ МИНИМИЗИРОВАННОЙ СИСТЕМНОЙ МОДЕЛИ

Рассмотрен метод построения модели минимального обобщенного алгоритма из частных моделей. Частные модели алгоритмов записываются в языке регулярных схем системных моделей. Используя систему аксиом тождественно-эквивалентных преобразований алгоритмов в регулярных схемах системных моделей, выносят за скобки общие эквивалентные части алгоритмов и получаем минимизированную системную модель. Такие операции позволяют строить системные модели из частных моделей структурного и событийного характера. Системные модели позволяют произвести анализ проектируемой технической системы методом системного моделирования.

Ключевые слова: *системная модель, минимизация алгоритмов, системное проектирование*

Постановка задачи

При проектировании сложных технических систем часто возникают задачи построения системных моделей. Построение системных моделей верхнего уровня иерархии из нижнего требует разработки метода синтеза обобщенной системной модели из частных системных моделей. Возможны несколько вариантов такого построения, которые отличаются в основном избыточностью записей в формализованных языках. Требуется решение задачи оптимизации обобщенной модели или в частном случае минимизации алгоритмов в обобщённой модели. Такие задачи предлагается решать с помощью метода синтеза системной алгоритмической модели в языке регулярных схем системных моделей.

Решение задачи

Минимизация обобщенного алгоритма осуществляется с использованием аксиомы №7 и №8 из алгебры операторов системы аксиом А [1]

$$(P \vee Q)R = (PR \vee QR) \text{ или}$$
$$P(Q \vee R) = PQ \vee PR,$$

т.е. вынесение за скобки одинаковых операторов. Однако в реальных ситуациях построения системных структурных и алгоритмических моделях возникают ситуации, когда производится вынос за скобки не одного оператора, а группы и связанных отношениями в алгоритмические цепочки. Усложняет процесс минимизации и различное размещение одинаковых цепочек в самом алгоритме. Ну и самый сложный случай, когда в группе алгоритмов существует несколько типов одинаковых цепочек, и они пересекаются друг с другом. Рассмотрим пример построения обобщенного алгоритма из частных. Существуют частные алгоритмы:

$$R_1 = Y_{об}Y_{гп}Y_{хп}Y_{тр}Y_{гп1}Y_{хп1}Y_{осв}Y_{к1}Y_{к2}\wedge$$
$$\wedge Y_{ан}Y_{к1}Y_{к2}Y_{укл}Y_{с};$$
$$R_2 = Y_{об}Y_{гп}Y_{хп}Y_{тр}Y_{гп1}Y_{хп1}Y_{осв}Y_{к1}Y_{к2}Y_{ан}Y_{к1}\wedge$$
$$\wedge Y_{к2}Y_{окр}Y_{гп}Y_{хп}Y_{укл}Y_{с};$$
$$R_3 = Y_{об}Y_{гп}Y_{хп}Y_{тр}Y_{гп1}Y_{хп1}Y_{осв}Y_{к1}Y_{к2}\wedge$$
$$\wedge Y_{ан}Y_{к1}Y_{к2}Y_{кр}Y_{гп}Y_{хп}Y_{с};$$
$$R_4 = Y_{об}Y_{гп}Y_{хп}Y_{тр}Y_{гп1}Y_{хп1}Y_{дек}Y_{хром}Y_{уп}Y_{гп}Y_{хр}Y_{с};$$
$$R_5 = Y_{об}Y_{гп}Y_{хп}Y_{тр}Y_{гп1}Y_{хп1}Y_{дек}Y_{хром}Y_{ник}Y_{пром}Y_{с};$$
$$R_6 = Y_{об}Y_{гп}Y_{хп}Y_{тр}Y_{гп1}Y_{хп1}Y_{дек}Y_{хром}Y_{мед}Y_{пром}Y_{с};$$
$$R_7 = Y_{об}Y_{гп}Y_{хп}Y_{тр}Y_{гп1}Y_{хп1}Y_{дек}Y_{пром}Y_{ян}Y_{гп}Y_{хп}Y_{с}.$$

Из них необходимо построить обобщенный алгоритм с минимальным числом повторения одинаковых цепочек операторов и одинаковых операторов. Исследуем цепочки операторов в частных алгоритмах $R_1 - R_7$ и подчеркнем одинаковые, произведем замены:

$$Y_{об}Y_{гп}Y_{хп}Y_{тр}Y_{гп1}Y_{хп1} = A;$$
$$Y_{к1}Y_{к2} = B;$$
$$Y_{гп}Y_{хп} = C;$$
$$Y_{пром}Y_{с} = D.$$

Тогда частные алгоритмы $R_1 - R_7$ запишутся в виде:

$$R_1 = AY_{осв}BY_{ан}BY_{укл}Y_{с};$$
$$R_2 = AY_{осв}BY_{ан}BY_{окр}CY_{с};$$
$$R_3 = AY_{осв}BY_{ан}BY_{хр}CY_{с};$$
$$R_4 = AY_{дек}Y_{укл}CY_{с};$$
$$R_5 = AY_{дек}Y_{хром}Y_{ник}Y_{пром}Y_{с};$$

$$R_6 = AY_{дек}Y_{хром}Y_{мед}Y_{пром}Y_c;$$
$$R_7 = AY_{дек}Y_{пром}Y_{ян}CY_c.$$

Очевидно, что хорошо склеятся R_1, R_2, R_3 и отдельно R_4, R_5, R_6, R_7:

$$R_{общ1} = \begin{cases} R_1, \text{ если } r_1 = 1, \\ R_2, \text{ если } r_1 = 0, r_2 = 1, \\ R_3, \text{ если } r_1 = 0,\ r_2 = 1, \end{cases}$$

$$R_{общ1} = AY_{осв}BY_{ан}B \underset{r_1}{(} Y_{укл} \vee \underset{r_2}{(} Y_{опр} \vee Y_{хр} \overset{r_2}{)} C \overset{r_1}{)} Y_c;$$

$$R_{общ2} = AY_{дек}Y_{хр} \underset{r_4}{(} \underset{r_3}{(} Y_{укл} \vee Y_{ян} \overset{r_3}{)} C \vee$$

$$\vee \underset{r_5}{(} Y_{ник} \vee Y_{мед} \overset{r_5}{)} Y_{пром} \overset{r_4}{)} Y_c;$$

$$R_{общ2} = \begin{cases} R_4, \text{ если } r_4 r_3 = 1, \\ R_7, \text{ если } r_4 = 1,\ r_3 = 0, \\ R_5, \text{ если } r_4 = 0,\ r_5 = 1, \\ R_6, \text{ если } r_4 = 0,\ r_5 = 0. \end{cases}$$

Теперь можно построить обобщенный алгоритм из двух предыдущих:

$$R_{общ} = \begin{cases} R_{общ}, \text{ если } r_6 = 1, \\ R_{общ}, \text{ если } r_6 = 0. \end{cases}$$

$$R_{общ} = A \underset{r_6}{(} Y_{осв}BY_{ан}B \underset{r_1}{(} Y_{осв} \vee \underset{r_2}{(} Y_{окр} \vee Y_{mp} \overset{r_2}{)} C \overset{r_1}{)} \vee$$

$$\vee Y_{дек}Y_{хр} \underset{r_4}{(} \underset{r_3}{(} Y_{укл} \vee Y_{ян} \overset{r_3}{)} C \vee$$

$$\vee \underset{r_5}{(} Y_{ник} \vee Y_{мед} \overset{r_5}{)} Y_{пром} \overset{r_4}{)} \overset{r_6}{)} Y_c,$$

что при раскрытии операторов имеет вид:

$$R_{общ} = Y_{об}Y_{гп}Y_{хп}Y_{mp}Y_{гп1}Y_{хп2} \underset{r_6}{(} Y_{осв}Y_{к1}Y_{к2}Y_{ан} \wedge$$

$$\wedge Y_{к1}Y_{к2} \underset{r_1}{(} Y_{укл} \vee \underset{r_2}{(} Y_{опр} \vee Y_{хр} \overset{r_2}{)} Y_{гп}Y_{хп} \overset{r_1}{)} \vee$$

$$\vee Y_{дек}Y_{хр} \underset{r_4}{(} \underset{r_3}{(} Y_{упл} \vee Y_{ян} \overset{r_3}{)} Y_{гп} \wedge$$

$$\wedge Y_{хп} \vee \underset{r_5}{(} Y_{ник} \vee Y_{мед} \overset{r_5}{)} Y_{пром} \overset{r_4}{)} \overset{r_6}{)} Y_c.$$

Если сравнивать с исходными R_1 – R_7, то увидим, что в $R_{общ}$ операторов – 27, условий – 6, а в исходных – 93 оператора, 0 условий. Приняв во внимание, что каждый оператор – это технологическая операция в гальванической ванне, то, построив $R_{общ}$, получим структуру многопроцессорной гальванической линии, сократив 66 технологических агрегатов.

В данном примере мы не получили 100% минимизации повторяемости операторов в $R_{общ}$. Повторяются два раза цепочки B и C и вытащить их с помощью аксиом алгебры операторов практически невозможно.

Анализ этого примера и других показывает, что эффективность минимизации зависит от того, какие цепочки выбирать в качестве общих и какие алгоритмы склеивать первыми. Если бы мы начали склеивать первыми не R_1, R_2, R_3, а R_1 и R_6, то результат был бы другой. В связи с этим была разработана методика построения обобщенных минимизированных алгоритмов, заключающаяся в следующем.

Во-первых, производится упорядочивание частных алгоритмов для синтеза обобщенного на предмет последовательности склеивания.

Во-вторых, строятся попарные группы алгоритмов по критерию «большей» (совместимости) квазиэквивалентности.

В-третьих, строятся обобщения в наиболее «мощной» и «разнообразной» группе. Затем к обобщению подбирается более похожий частный алгоритм и также обобщается. Далее действуем по этому методу до обобщения последнего частного алгоритма в общую группу. К вопросу терминологии, «похожесть», «совместимость», «квазиэквивалентность»- это когда в двух частных алгоритмах есть тождественно-эквивалентные цепочки операторов Y_1, Y_2, … Y_i. «Большая» похожесть – это когда из группы $B_i \in B$ алгоритмов выбираются два, имеющих большое количество одинаковых цепочек Y_1, Y_i и одинаковых операторов. «Мощность цепочки», «мощность алгоритма» - это количество операторов в данном алгоритме. «Разнообразие группы» - это когда в алгоритме либо цепочке операторы Y_i не однозначны $Y_i \neq Y_{i+1}$.

Рассмотрим теоретико-множественное обоснование методики построения минимизированного обобщенного алгоритма. Запишем разноименные операторы, составляющие все частные алгоритмы, через элементы множества $y_i \in Y$ – это алфавит. Частный алгоритм R_i составляет векторное множество операторов $y_i \in B_i$. Обобщенный $R_{общ}$ алгоритм, состоящий из частных R_i, составит векторное множество $B_i \in A$, включающий в себя упорядоченные векторные множества B_i частных алгоритмов R_I. Для решения этой задачи поступаем в соответствии с предлагаемой методикой.

1. Анализируем каждое множество B_i частного алгоритма. Выбираем из B_i такой, который обладает наибольшей мощностью y_i (содержащий наиболь-

шее множество неповторяющихся y_i). Затем анализируем все остальные B_i и упорядочиваем их по степени мощности y_i. Строим очередь множества B_i по убывающей степени мощности y_i.

2. Анализируем бинарные цепочки, т.е. ($\overline{Y_i Y_{i+1}}$) в самом первом B_i и проверяем их нахождение во всех остальных B_i. Обозначим ее через подмножество D_i. Если D_i находится во всех остальных B_i более одного раза, то записывает D_i в множество $D_i \in D$. Если D_i встречается только один раз, то такую цепочку вычеркиваем из множества D. Затем формируем бинарные цепочки ($\overline{Y_{i+1} Y_{i+2}}$) и повторяем процесс по проверке на наличие во всех B_i. Присваиваем ему D_{i+1} и, в случае неоднократного нахождения в B_i, заносим в множество $D_{i+1} \in D$. Действуем таким образом до тех пор, пока не сформируем бинарные цепочки по всем сочетаниям в B_i. Формируем множество бинарных цепочек $D_i \in D$. Анализируем входимость цепочек D_i в B_i по количеству вхождений и ранжируем их (строим последовательность D_i, D_{i+1}, D_{i+2}, …) по количеству вхождений.

3. Строим на основе самого мощного D_i цепочку из трёх операторов

$$(D_i + y_{i+2}) \in F_i \text{ и } (y_{i-1} + D_i) \in F_i.$$

Повторяем все действия в соответствии с п.1 и составим множество $F_i \in F$ цепочек из трех операторов. Затем упорядочиваем множество F по большинству вхождений F_i в B_i.

4. Строим таким образом цепочки из 4, 5, 6, 7 и более операторов до тех пор, пока не выполнится условие, что i – я цепочка будет иметь вхождений в множество частных алгоритмов B_i не более одного раза. На этом процесс формирования цепочек общих частей хотя бы двух алгоритмов заканчивается. Таким образом, определяем все возможные общие части в частных алгоритмах R_i и общие части, имеющие наибольшую «мощность», а также ранжирование общих частей (цепочек) по количеству вхождений в различные частные алгоритмы. Это важно для дальнейшей методики построения минимизированного общего алгоритма.

5. Произведем замену имеющихся в частных алгоритмах R_i цепочки операторов (y_i, y_{i+1},…, y_{i+n}) на соответствующие им обозначения подмножеств C, D, F и далее по принципу поглощения большего подмножества меньшим.

6. Выбираем из ранжированного множества $B_i \in B$ два подмножества B_i, имеющие наибольшее количество одинаковых цепочек с наибольшей мощностью Y_i, и самих B_i, обладающих наибольшей мощностью. На основе аксиомы №7 и №8 из алгебры операторов эквивалентных преобразований алгоритмов [1], выносим общие цепочки этих частных алгоритмов R_i, которые соответствуют подмножествам B_i влево и вправо.

7. Производим в множестве R_i замену двух частных R_i, объединенных в общий, на $R_{общ1}$.

8. В полученном множестве R_i повторяем процесс, описанный в п.6 этой методики. Получим $R_{общ2}$ для 3-х частных алгоритмов.

9. Производим в множестве R_i замену $R_{общ1}$ и R_3 на $R_{общ2}$.

10. В полученном множестве R_i повторяем процесс, описанный в п. 6 этой методики. Получим $R_{общ3}$ для 4-х частных алгоритмов.

11. Повторяем дальнейшее объединение частных R_i в $R_{общ}$ до тех пор, пока в множестве R_i окажется один $R_{общ}$. Это и будет минимизированный $R_{общ}$, содержащий самое минимальное количество операторов. Возможен случай, когда в множестве R_i окажется две или больше группы подмножеств B_i, обладающих большими «мощностями» y_i и не имеющих общих цепочек между собой. Тогда в $R_{общ}$ они будут описаны через знак дизъюнкции $\vee$ без общих частей.

$$R_{общ} = \underset{r_n}{(} R_{общ1} \vee \underset{r_{n+1}}{(} R_{общ2} \vee R_{общ3} \overset{r_{n+1}}{)} \overset{r_n}{)},$$

т.е. $R_{общ}$ будет состоять из 3-х групп обобщенных алгоритмов и условий r_n между группами.

Выводы

Таким образом, следуя описанному выше алгоритму, можно склеить частные алгоритмы в обобщенный с минимальным количеством операторов и тождественно-эквивалентный каждому из его составляющих алгоритмов. Частным случаем этого алгоритма является методика сравнения двух и более алгоритмов. Общие цепочки в них укажут равные части алгоритмов, а пути алгоритмов по условиям r_n укажут на различные части алгоритмов. Этот факт важен при сравнении системных моделей.

Литература

1. *Илюшко, В.М. Системное моделирование в управлении проектами [Текст]: моногр. / В.М. Илюшко, М.А. Латкин; Нац. аэрокосм. ун-т им. Н.Е. Жуковского «Харьк. авиац. ин-т». – Х.: ХАИ, 2010. – 220 с.*

2. *Илюшко, В.М. Методы и модели создания метасистем [Текст]: дис. ... д-ра техн. наук: 05.13.06; защищена 26.06.1998; утв. 12.01.99 / Илюшко Виктор Михайлович. – Х., 1998. – 451 с.*

Поступила в редакцию 25.02.2013, рассмотрена на редколлегии 13.03.2013

Рецензент: д-р техн. наук, проф., зав. каф. информационных технологий проектирования ЛА Е.А. Дружинин, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.

СІНТЕЗ МІНІМІЗОВАНОЇ СИСТЕМНОЇ МОДЕЛІ

Я.В. Ілюшко

Розглянуто метод побудови моделі мінімального узагальненого алгоритму з приватних моделей. Приватні моделі алгоритмів записуються в мові регулярних схем системних моделей. Використовуючи систему аксіом тотожно-еквівалентних перетворень алгоритмів в регулярних схемах системних моделей, виносять за дужки загальні еквівалентні частини алгоритмів і отримуємо мінімізовану системну модель. Такі операції дозволяють будувати системні моделі з приватних моделей структурного і подієвого характеру. Системні моделі дозволяють зробити аналіз проектованої технічної системи методом системного моделювання.

Ключові слова: системна модель, мінімізація алгоритмів, системне проектування.

SYNTHESIS TO MINIMIZE SYSTEM MODEL

Ya.V. Ilyushko

The method of constructing a model of minimal generalized algorithm of particular models. Partial models of algorithms written in the language of regular patterns of system models. Using a system of axioms of identity-equivalent transformation algorithms in regular patterns of system models, introduce the brackets general equivalent of algorithms and we minimize system model. These operations allow us to construct a model system of partial models of structural and event-driven nature. System models allow an analysis of the projected technical system by system simulation.

Keywords: system model, minimization algorithms, system design.

Илюшко Ярослав Викторович – канд. техн. наук, доцент кафедры графического и компьютерного моделирования, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.