УДК 510.62

В. А. ЧИКИНА, канд. техн. наук, *Ю. В. КОВАЛЕВ*

ДЕКОМПОЗИЦИЯ УРАВНЕНИЙ АЛГЕБРЫ КОНЕЧНЫХ ПРЕДИКАТОВ МЕТОДОМ ВСПОМОГАТЕЛЬНЫХ ПЕРЕМЕННЫХ

Простой и эффективный метод декомпозиции уравнений алгебры конечных предикатов (АКП) путем введения вспомогательных переменных предусматривает обозначение выделенных участков формул произвольных переменных и запись полученных предикатов в виде системы уравнений [1]. При схемном решении [2] возникает самостоятельная задача построения схем, соответствующих заданным уравнениям из обратимых элементов, которые могут реализовывать отношения на две и более переменные.

Построение обратимых элементов на четыре и более переменных затруднено из-за высокой сложности получаемых подсхем. Реализация же их на два и три входа достаточно проста.

В данной работе предлагается процедура декомпозиции на бинарные и тернарные отношения, которые могут быть реализованы простейшими обратимыми элементами на два или три входа.

Полагаем, что декомпозируемое уравнение задается произвольной скобочной формой. Никаких иных ограничений на структуру формулы не налагается. Пусть исходное уравнение представлено строкой символов из алфавита σ и имеет произвольную длину, выходные данные — также строки символов. В каждой такой строке будет записано одно-, двух- или трехместное уравнение.

Процесс декомпозиции разделим на три этапа [3]: лексический анализ исходной строки с целью построения структуры декомпозируемого уравнения; синтаксический анализ правильности построения структуры уравнения; вычленение простейших отношений и их запись в выходной регистр.

Остановимся подробно на каждом из этих этапов.

Результатом этапа лексического анализа является список лексем. Введем переменные:

входной регистр $S=\{S_1, S_2, \ldots, S_i, \ldots, S_{k_S}\}$, где S_i — i-я пе-

ременная регистра с областью определения σ, k_S — длина входной строки или же число переменных в регистре;

регистр лексем $L = \{L_1, L_2, ..., L_l, ..., L_{k_L}\}$, $L_i = \{L_{i1}, ..., L_{ij}, ..., L_{ik_{L_i}}\}$, где L_i — i-я переменная, определенная на множестве лексем σ_L, k_L — число лексем, L_{ij} — j-я буква лексем, определенная на множестве букв σ;

вспомогательный регистр $R = \{R_1, R_2, ..., R_l, ..., R_{k_R}\}$, где R_i — i-й разряд регистра, k_R — число разрядов вспомогательного регистра

$$k_R = \max(k_{L_i}),\ i \in \{1, 2, ..., k_L\};$$

вспомогательная булева переменная $FL = \bigvee_{i=1}^{k_R} \overline{R\bar{i}}$;

дискретное время, определенное на множестве целых чисел t.

Введем вспомогательные предикаты:

$$z(x) = \begin{cases} 1, \text{ если } x \text{ есть знак операции,} \\ 0 — \text{в противном случае,} \end{cases}$$

$$z(x) = x^{\neg} \vee x^{\vee} \vee x^{\&} \vee x^{\supset} \vee x^{\sim}.$$

Знаки операций, стоящие в показателях узнаваний, следует понимать как символы.

$$P_3(x, y) = \bigwedge_{i+1}^{k_X} x^{i+1} y^i,$$

где k_x — число значений x. Если $P_3(x, y) = 1$, то переменная x на единицу больше y.

Предикат $P_1(i, j) = 1$, если значение 1-го разряда j-го регистра лексем равно значению i-й переменной входного регистра S:

$$\begin{cases} \bar{L}_{t=1,1} i^{\varnothing}_{t=1} i^{\varnothing}_{t=\varnothing} = 1, \\ L_{t=1,1,j} \sim S_t; \text{ при } t > 1, \\ P_3(i_{t+1}, i_t) = 1, \\ P_3(j_{t+1}, j_t) = 1. \end{cases}$$

Предикат $P_2(k) = 1$, когда значение k-го регистра лексем равно значению вспомогательного регистра R:

$$\begin{cases} L_{t+1,k} \sim R_t, \\ P_3(k_{t+1}, K_t) FL_{t+1}. \end{cases}$$

Предикат $P_5(i)$ описывает операцию конкатенации:

$$\begin{cases} \overline{R\bar{i}_{,j}} \supset (R_{t+1,j} \sim R_{t,j}), \\ R_{t,j} \supset (R_{t+1,j} \sim S_t), \end{cases}$$

где $j = \overline{1, k_R}$, k_R — число переменных в регистре R.

Предикат $P_6(k)=1$, если вспомогательная переменная установлена в единицу ($FL=1$), а значение первой буквы вспомогательного регистра равно значению k-го разряда входного регистра:

$$P_6(k)=FL_{t+1}(R_{t+1,1}\sim S_k)\left(\bigwedge_{j=1}^{k_R} R_{t+1,j}\right);$$

$$\text{Б}(x,\ y)=\begin{cases}1, & \text{если } x>y,\\ 0 & \text{— в противном случае;}\end{cases}$$

$$\text{Б}(x,\ y)=\bigvee_{i=1}^{k_x}\bigvee_{j=1}^{i-1}x^i y^j,$$

где k_x — число возможных значений переменной x.

Предикат $P_8(i, x, y)$ обращается в единицу, если $x_j=y$, а переменная i получает значение j:

$$P_8(i,\ x,\ y)=\bigwedge_{j=1}^{k_x}(x_j^y\supset i_{t+1}^j);$$

$$P_7(x,\ y,\ z)=\exists i P_8(i,\ x,\ y)\,\text{Б}(i,\ z);$$

$$P_9(x,\ y,\ z)=\exists i P_8(i,\ x,\ y)\,\overline{\text{Б}(i,\ z)}(i\oplus z).$$

Приведем правила, описывающие закономерности образования лексем.

1. Поскольку лексема может состоять из многих символов, то необходима вспомогательная переменная для определения, к какой лексеме следует отнести очередной символ входной строки. Положим $FL_{t=0}=\varnothing$.

2. Считаем содержимым регистров лексем пробелы

$$L_{t=0,j}^{-}=1;\ (j=\overline{1,\ k_L}).$$

3. Для всех моментов дискретного времени $t>0$ справедливы следующие правила выделения лексем: если очередной символ входной строки — знак операции ($\urcorner$ — знак отрицания, $\vee$ — знак дизъюнкции, $\&$ — конъюнкции, $\supset$ — импликации, $\oplus$ — знак суммы по модулю; 2, $\sim$ — знак эквиваленции) и вспомогательная переменная равна нулю, что равносильно пустому вспомогательному регистру (предыдущая последовательность символов распределена по лексемам), то очередная лексема есть выделенный символ

$$z(S^i)\,\overline{FL}\supset P_1(i,\ j),$$

где i — номер буквы входного регистра, j — номер очередной лексемы.

4. Если очередной i-й символ — знак операции и предыдущая последовательность символов могла образовать лексему (н/п X: $:A\&$), то необходимо эту последовательность символов обозначить лексемой. Увеличим счетчик лексем на единицу. Очередной

лексемой будет i-й символ, а значение вспомогательной переменной следует установить в ноль, так как текущая последовательность символов уже распределена по регистрам лексем

$$z(S_i)\,FL \supset P_2(i)\,P_1(i,\ j).$$

5. Если текущий символ — пробел, то он указывает лишь на то, что предыдущая последовательность символов должна быть распределена по лексемам

$$(\text{н/п } A\&X:A\_),\quad \overline{S_i}\,FL \supset P_2(i).$$

6. Если некоторая предыдущая последовательность символов включена в состав очередной лексемы и текущий символ — скобка, а в текущей последовательности уже есть и открывающаяся и закрывающаяся скобки (н/п $X(A)$)), то текущий символ есть операторная скобка. Она служит для указания порядка выполнения операций в транслируемой формуле. Реакцией системы должно быть выделение такой скобки в отдельную лексему

$$S_i^{\vee}(FLP_7(R_t,\ (,\ 1)\,P_7(R_t,\ ),\ 1) \supset P_2(i)\,P_1(i,\ j).$$

7. Если предыдущая последовательность символов может составлять лексему ($FL=1$), очередной символ — открывающаяся скобка и в последовательности есть двоеточие — указание на степень узнавания, то реакция системы аналогична рассмотренной в п. 2

$$S_i^{\vee}FLP_7(R_t,\ :,\ 1) \supset P_2(i)\,P_1(i,\ j).$$

8. Если очередной символ во входном регистре — открывающаяся скобка, $FL=1$ и отсутствует двоеточие в регистре R (н/п′X(′), эту скобку необходимо включить в R.

9. Если очередной символ — скобка и $FL=0$ (н/п′X:$A\vee X$:B′), то скобку необходимо обозначить новой лексемой:

$$S_i^{\vee}(FL \supset P_1(i).$$

10. Если очередной символ — закрывающаяся скобка, предыдущие символы объединены в лексему и среди них имеется открывающаяся скобка (н/п′$X(A)$′), то эту скобку необходимо присоединить к имеющимся символам

$$S_i^{\vee}FLP_7(R_t,\ (,\ 1) \supset P_5(i).$$

11. Если очередной символ — не знак операции, не пробел, не скобка и $FL=1$, то этот символ необходимо присоединить к предыдущим символам:

$$P_{10}FL \supset P_5(i),$$

где вспомогательная переменная $P_{10}=\overline{z(S_i) \vee S_i^{\vee(0)}}$.

12. Если же $FL=0$, то этот символ будет первым символом очередной лексемы:

$$P_{10}\overline{FL} \supset P_6(i).$$

Перейдем к описанию этапа синтаксического анализа. Введем регистр типов лексем

$$TL = \{Tl_1,\ Tl_2,\ ...,\ Tl_{k_l}\},$$

где Tl_i — тип i-й лексемы, $i = \overline{1,\ k_l}$. В качестве типа лексемы может выступать переменная, скобка открывающаяся, скобка закрывающаяся, знаки операций:

$$TL_l^{(,),\text{п}} \vee z(L_i)(Tl_i \sim L_l) = 1.$$

Предикат $P_{11}(L, TL)$ формализует определение типа лексем:

$$(z(L_{t,t}) \vee L_{i,t}^{(,)}) \supset (Tl_{i,t+1} \sim L_{i,t});$$

$$L_{i,t}^{(,)} \vee z(L_{i,t}) \supset Tl_{i,t+1}^{\text{п}},$$

где $i = \overline{1, k_l}$.

Введем предикат синтаксически правильной формулы Син(x). Аргумент x — регистр типов лексем, x_i — $(i=1,\ n)$ — i-я переменная в регистре, n — число переменных в регистре:

$$x_1^{\text{п}} D_0(x_2) \vee x_1^{(}(\overline{x_2} \vee x_2^{\text{п}} \vee x_2^{)}) \vee \overline{x_1} x_2^{\text{п}} = 1;$$

$$\bigwedge_{i=2}^{n-1} [x_i^{\text{п}}(z(x_{i-1}) \vee x_{i-1}^{(})(x_{i+1}^{)} \vee D_0(x_{i+1})) \vee \overline{x_i}(x_{i-1}^{(} \vee$$

$$\vee D_0(x_{i-1}))\, x_{i+1}^{\text{п}} \vee D_0(x_i)(x_{i-1}^{\text{п}} \vee x_{i-1}^{)})\, x_{i+1}^{\text{п},(} \vee x_i^{(} \overline{x_{i-1}^{\text{п}}}\, \overline{D_0(x_{i+1})} \vee$$

$$\vee x_i^{)} \overline{x_{i+1}^{\text{п}}}\, \overline{D_0(x_{i-1})}\, \overline{x_{i+1}}] = 1;$$

$$x_n^{\text{п}} z(x_{n-1}) \vee x_n^{)}(D_0(x_{n-1}) \vee \overline{x_{n-1}}) = 1.$$

Предикат $P_{12}(TL, uw)$ формализует подсчет уровней вложенности лексем:

$$S_k(Tl,\ i,\ p,\ q) = 1, \quad Tl_{i,t}^{(} uw_{i,t+1}^{p-Q+1} \vee Tl_{i,t}^{)} uw_{i,t+1}^{p-Q} \vee \overline{Tl_{i,t}^{(,)}} uw_{i,t+1}^{\varnothing} = 1,$$

где

$$S_k(Tl,\ i,\ p,\ q) = \begin{cases} t = \overline{0,\ k_l};\ i = \overline{1,\ k_l}, \\ P_{t=0}^{\varnothing} q_{t=0}^{\varnothing} = 1, \\ Tl_{i,t}^{(} \supset \text{Сл}_1(P_{t+1},\ P_t), \\ Tl_{i,t}^{)} \supset \text{Сл}_1(q_{t+1},\ q_t), \\ \text{Сл}_1(x,\ y) = \bigvee\limits_{i=1}^{k_x} x^i y^{i-1}. \end{cases}$$

Регистр uw — уровни вложенности лексем:

$$uw = \{uw_1,\ ...,\ uw_{k_l}\};$$

$$uw_i^{\varnothing} \vee ... \vee uw_i^m = 1,$$

где uw_i — уровень вложенности i-й лексемы, m — максимально допустимый уровень вложенности лексемы.

Проверка на ошибки в следовании скобок осуществляется следующим образом: если уровень вложенности какой-либо лексемы меньше 0, то в выражении от первой лексемы до рассматриваемой число закрывающихся скобок больше числа открывающихся:

$$M(uw_{t,t},\ \varnothing) \supset ER^{\text{следование}},\ i = \overline{1,\ k_l},$$

где $M(x, y) = \overline{\text{Б}(x, y)}\,(x \otimes y)$, ER — переменная, указывающая на вид ошибки в выражении.

Для нахождения самой приоритетной операции [3] необходимо выделить фрагмент входного регистра, имеющий самую большую вложенность. Введем следующие переменные: muw — максимальный уровень вложенности, переменная определена на множестве целых чисел; sbo — номер лексемы типа открывающаяся скобка с самым большим уровнем вложенности:

$$i^1 \supset muw_{t+1} \sim uw_{t,i},$$

$$\overline{i^1} \supset (Tl_i^{\text{Б}}(uw_{t,t},\ muw_t \supset (muw_{t+1} \sim uw_{t,t})\, sbo^t.$$

Для формализации поиска закрывающейся скобки с самым большим уровнем вложенности введем переменную sbz — номер лексемы типа закрывающаяся скобка с самым большим уровнем вложенности. Правило определения значения sbz следующее. Первая скобка, которая стоит после открывающейся скобки с самым большим уровнем вложенности:

$$i^{sbo+1} \supset LV_t;\ \ (i = \overline{sbo+1,\ k_l});$$

$$LV_t\, Tl_{tl} \supset sbz^i \overline{LV_{t+1}},$$

где LV — вспомогательная булева переменная.

Определение самой приоритетной операции заключается в анализе выделенного ранее фрагмента на предмет наличия знака операции, а именно, отрицания, конъюнкции, дизъюнкции, импликации, неравнозначности, эквивалентности. Запишем соответствующую систему уравнений:

$$\begin{cases} P_{12}(p, q, z) = \overline{\bigvee_{i=p}^{q} Tl_t^z(i)},\ P_{13}(p, q, z) = \bigvee_{i=p}^{q} Tl_t^z(i) \supset S_p^i, \\ \left(\bigwedge_{z \in \{-,\,\&\}} P_{12}(sbo,\ sbz,\ z)\right) P_{13}(sbo,\ sbz,\ \vee), \\ \left(\bigwedge_{z \in \{-,\,\&,\,\vee\}} P_{12}(sbo,\ sbz,\ z)\right) P_{13}(sbo,\ sbz,\ \supset), \\ \left(\bigwedge_{z \in \{-,\,\&,\,\vee,\,\supset\}} P_{12}(sbo,\ sbz,\ z)\right) P_{13}(sbo,\ sbz,\ \oplus), \\ \left(\bigwedge_{z \in \{-,\,\&,\,\vee,\,\supset,\,\otimes\}} P_{12}(sbo,\ sbz,\ z)\right) P_{13}(sbo,\ sbz,\ =), \\ \left(\bigwedge_{z \in \{-,\,\&,\,\vee,\,\supset,\,\otimes\}} F_{13}(sbo,\ sbz,\ z)\right) us(L_t,\ L_{t+1},\ sbo,\ sbz,\ k_l). \end{cases}$$

Следующим этапом является выдача в регистр выходной информации очередного простейшего уравнения. Переменная W служит для описания входного регистра, W_i — i-я переменная регистра, определенная на множестве всех букв $(i=\overline{1, k_W})$, k_W — число переменных в регистре. Если выделенная операция — отрицание, то у такой операции всего один аргумент, у всех остальных — по два. Следует также учесть необходимость модификации исходного уравнения с учетом вычленяемого отношения.

В случае короткого уравнения (на две или три переменные), либо когда уже были выделены имеющиеся отношения, а это — последнее, тогда необходимо выдать заключительное уравнение (или единственное) и остановиться. Для этого введем признак окончания:

$$\begin{cases} \text{РО}^{\text{да}} \vee \text{РО}^{\text{нет}} = 1, \ \overline{i^{sp}z}\,(L_{t,i})\,\widetilde{L_{t,i}} \supset \text{РО}_t^{\text{нет}}, \\ \overline{i^{sp}z}\,(L_{t,i}) \supset \text{РО}_t^{\text{да}}, \ (i=\overline{1, k_l}), \\ L^{-}_{t,sp}\text{РО}_t^{\text{да}} \supset W^{-}_{t+1,1}\,(W_{t+1,2} \sim L_{t,sp+1})\,W^{1}_{t+1,3}W^{-}_{t+1,4}, \end{cases}$$

$$\begin{cases} z\,(L_{t,sp})\,L^{-}_{t,sp}\text{РО}^{\text{да}} \supset (W_{t+1,1} \sim L_{t,sp})(W_{t+1,2} \sim L_{t,sp-1})(W_{t+1,3} \sim \\ \qquad \sim L_{t,sp+1})\,W^{1}_{t+1,4}, \\ L^{-}_{t,sp}\text{РО}^{\text{нет}} \supset (W_{t+1,1} \sim L_{t,sp})(W_{t+1,2} \sim L_{t,sp+1})\,W^{t}_{t+1,3}W^{-}_{t+1,4}, \\ \overline{L^{-}_{t,sp}}\text{РО}^{\text{нет}} z\,(L_{t,sp}) \supset \bigvee_{i=1}^{3}\,(W_{t+1,i} \sim L_{t,sp-1+i})\,W^{t}_{t+1,4}. \end{cases}$$

t	L_t, W_t
0	L_0=″, W_0=″
1	L_1={X:A, &, (, Y:A, V, C(X),), V, Y:A, &, Z:B, ~, V(1)}. W_1=″
2	L_2={X:A, &, (M_1,), Y, Y:A, &, Z:B, ~, V(1)}, W_2 = {Y : A, V, C (X), M1}
3	L_3={M2, V, Y:A, &, Z:B, ~, V(1)}, W_3={X:A, &, M1. M2}
4	L_4={M2, V, M3, ~, V (1)}, W_4={Y:A, &, Z:B, M_3}
5	L_5={∅} W_5={M2, V, M3, V (1)}

Теперь произведем корректировку таблицы лексем, исключив имена, которые вошли в сгенерированное уравнение:

$$\text{РО}^{\text{нет}}L^{-}_{t,sp} \supset L^{t}_{t+1,sp} \bigwedge_{j=sp+1}^{k_l-1} (L_{t+1,j} \sim L_{t,j+1});$$

$$\text{РО}^{\text{чет}}\overline{L_{t,sp}} \supset L^{t}_{t+1,sp} \bigwedge_{j=sp}^{k_t-2} (L_{t+1,j} \sim L_{t,j+2}).$$

Может встретиться такой случай, когда в скобках останется всего один операнд

$$(\text{н/п}\, S_t = {}'A\,(B \vee C) \sim T' \Rightarrow W' = {}'B \vee C \sim M',$$
$$S_{t+1} = {}'A\,(M) \sim T'),$$

тогда необходимо удалить эти, ставшие ненужными, скобки:

$$\bigwedge_{t=1}^{sbo-1} (L_{t+1,t} \sim L_{t,t+1}) = 1;$$

$$\bigvee_{t=sbz-1}^{k_t-2} (L_{t+1,t} \sim L_{t,t+1}) = 1.$$

В заключении приведем небольшой пример декомпозиции уравнения:

$$x^a \,\&\, (y^a \vee c\,(x)) \vee y^a z^b \sim V^1;$$

$$S = {}'X : A \,\&\, (Y : A \vee C\,(X)) \vee Y : A \,\&\, Z : B \sim V\,(1).$$

Таким образом, в результате получим систему уравнений:

$$y^a \vee c\,(x) = M_1,$$
$$x^a M_1 = M_2,$$
$$y^a z^b = M_3,$$
$$M_2 \vee M_3 = V^2.$$

Список литературы: 1. *Шабанов-Кушнаренко Ю. П.* Теория интеллекта. Математические средства. Х., 1984. 144 с. 2. *Шабанов-Кушнаренко Ю. П.* Теория интеллекта. Технические средства. Х., 1986. 132 с. 3. *Вайнгартен Ф.* Трансляция языков программирования. М., 1977. 190 с.

Поступила в редколлегию 09.01.90