

5. 6. 2006 Каноническая декомпозиция

Статья посвящена канонической декомпозиции, восходит которая к древней истории проектирования динамических логических цепей, но до сих пор состоянчески корректно не обоснованной.

Йозеф БОКР

КАНОНИЧЕСКАЯ ДЕКОМПОЗИЦИЯ

Йозеф Бокр, г. Пльзень, Чехия

АННОТАЦИЯ

Имеются два подхода к канонической декомпозиции, «утверждают» которые, что она регламентирована с одной стороны функцией переходов [1,2] заданного конечного автомата (прототипа), и с другой функцией возбуждения [3], не обращая внимание на то, что каноническая декомпозиция прежде всего декомпозиция, хотя с предопределенной блочной структурой. Поэтому в заметке показывается несостоятельность имеющихся пониманий канонической декомпозиции и используя понятие декомпозиции [1] прототипа, дается состоянчески корректное её определение. Приведем также модификации канонической декомпозиции и покажем как подобрать элементы «памяти», чтобы минимизировать число аргументов функции возбуждения.

1. Введение

Пусть задан, ради простоты, детерминированный, динамический полуавтомат

$$A = \langle X, S, \delta \rangle$$

где X и S – алфавит соответственно входной и состояний и δ - функция переходов

$$\delta : S \times X \mapsto S : \langle s, x \rangle \mapsto s' (pred s')$$

где s' - состояние – приемник (или же его предикация) исходного состояния s .

Тогда [1,2] каноническая декомпозиция заданного A , регламентирует которую функция переходов, имеет вид – рис. 1. а), где блок задержки объект с предвосхищением. Но, так как будущее s' в настоящий момент времени ни документируемо, ни манипулируемо, придется обойтись её предсказанием $pred s'$. Но, так как $pred s'$ находится в памяти лица, идентифицирующего каноническую декомпозицию A или в его внешней вспомогательной памяти (таблица, диаграмма, матрица и т.п. переходов), то вряд ли её встретят в схеме по рис. 1. а). И ещё, что делать, если вместо элементов задержки применить триггера?

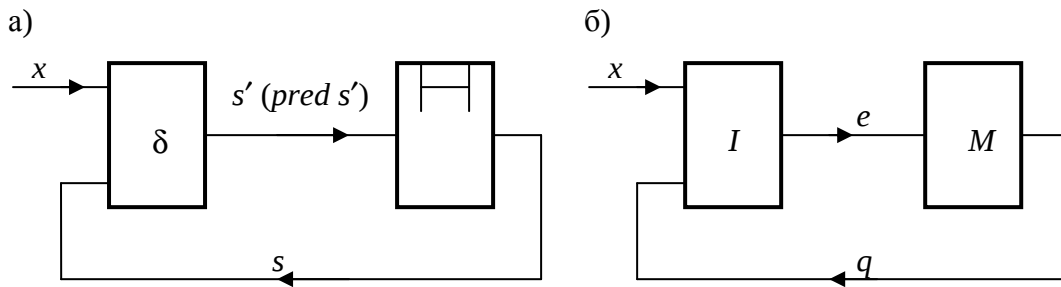


Рис. 1. Имеющиеся концепции канонической декомпозиции.

Займемся канонической декомпозицией [3], регламентированной функцией возбуждения

$$\mu : Q \times Q \rightarrow E : \langle q, q' \rangle \mapsto e$$

где E и Q – алфавит соответственно возбуждений и состояний блока памяти M , при заданном кодировании k состояний $k : S \rightarrow Q : s \mapsto q$. Отсюда $e = \mu(q, q') = \mu(q, \delta(q, x)) = \rho(q, x)$. Заметим, что блок памяти M , как ни странно, не располагает стиранием, но что гораздо важнее – приведенное понимание не поддается эффективному анализу своей работы. Ведь конечно – полуавтоматная модель M имеет вид

$$M = \langle E, Q, \delta_M \rangle$$

где δ_M - функция переходов

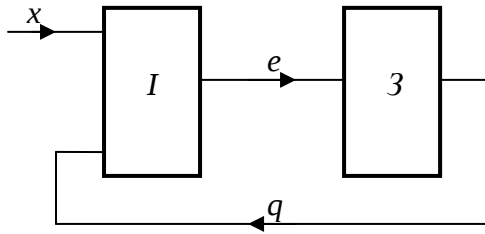
$$\delta_M : Q \times E \rightarrow Q : \langle q, e \rangle \mapsto q'$$

и подставляя $q' = \delta_M(q, e)$ в функцию возбуждения, получают $e = \mu(q, \delta_M(q, e)) = \mu(q, (\delta_M(q, \mu(q, \delta_M(q, e)))))) = \dots$ регресс в бесконечность.

2. Каноническая декомпозиция

Рассмотрим каноническую композицию K (рис. 2.) прототипа $A = \langle X, S, \delta \rangle$, где $\delta : S \times X \rightarrow S : \langle s, x \rangle \mapsto s'$, при заданном кодировании k состояний $k : S \rightarrow Q : s \mapsto q$ (кодирование – инъективная функция), где Z – заменитель прототипа A , т.е. полуавтомат, иммитирует который своими переходами между своими состояниями иницированными возбуждением e переходы между состояниями образца иницированными стимулом x .

а)



б)

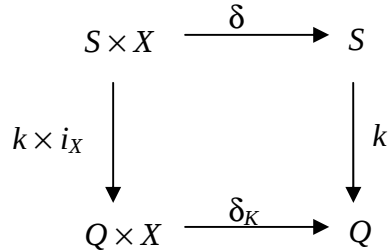


Рис. 2. а) Каноническая композиция K , б) коммутативная диграмма свертки функций.

Иными словами:

$$Z = \langle E, Q, \delta_3 \rangle$$

где E и Q – алфавит соответственно возбуждений и состояний при $|Q| \geq |S|$ и δ_3 – функция переходов $\delta_3 : Q \times E \rightarrow Q : \langle q, e \rangle \mapsto q'$ и

$$K = \langle X, Q, \delta_K \rangle$$

где для функции переходов $\delta_K : Q \times X \rightarrow Q : \langle q, x \rangle \mapsto q'$ имеет место равенство

$$\delta_3(q, e) = \delta_K(q, x)$$

Модель искомого входного статического блока I имеет формально вид

$$I = \langle Q \times X, E, \lambda \rangle$$

где искомая функция выходов λ есть

$$\lambda : Q \times X \rightarrow E : \langle q, x \rangle \mapsto e.$$

Заданный 3 является действительно заменителем образца A , имеет ли силу мноморфизм k образца A в заданный 3 [4], т.е. из коммутативной диаграммы свёртки функций (рис. 2. б)) вытекает

$$\delta * k = (k \times i_X) * \delta_K, \text{ т.е. } k(\delta(s, x)) = \delta_K(k(s), x)$$

где $i_X : X \rightarrow X : x \mapsto x$ - тождественная функция и $*$ - оператор свертки. Отсюда, так как

$$k(\delta(s, x)) = \delta_3(k(s), e) = \delta_3(k(s), \lambda(k(s), x)),$$

находят функцию λ .

На вопрос «Какова польза от приведенного определения канонической декомпозиции?» ответ проще простого: «Разработчики динамических схем волей неволей проектируют свои схемы интуитивно, фактически пользуясь предложенным подходом». Вот почему никто не страдает за «странных» концепций приведенных выше во введении.

Пример 1. Решить, служат ли автоматы A_1 и A_2 заданные таблицами переходов заменителями прототипа A (таб. 1. а)) при заданном кодировании состояний $k : \{1, 2, 3\} \rightarrow \{I, II, III, IV\} : 1 \mapsto I, 2 \mapsto III, 3 \mapsto IV$. Отсюда получают «рабочую» таблицу переходов образца A (таб. 1. б)) вместе с таблицами возбуждений возможных заменителей A_1, A_2 .

Таб. 1. Таблицы а) переходов прототипа A и потенциальных заметителей A_1, A_2 ; б) возбуждений автоматов A_1, A_2 из примера 1.

а)

A_2	s'		
$s \backslash x$	a	b	c
1	2	1	2
2	3	3	2
3	1	2	3

A_1	q'			
$q \backslash e$	α	β	γ	δ
I	II	III	IV	III
II	I	II	III	IV
III	IV	I	IV	II
IV	III	IV	II	I

A_2	q'			
$q \backslash e$	α	β	γ	δ
I	III	III	I	IV
II	I	IV	III	II
III	II	I	IV	III
IV	I	I	IV	II

б)

				A_1			A_2		
A	q'			e			e		
$q \backslash x$	a	b	c	a	b	c	a	b	c
I	III	I	III	β	?	β/δ	α/β	γ	α/β
III	IV	IV	III	α/γ	α/γ	?	γ	γ	δ
II	I	III	IV	α	γ	δ	α	γ	β

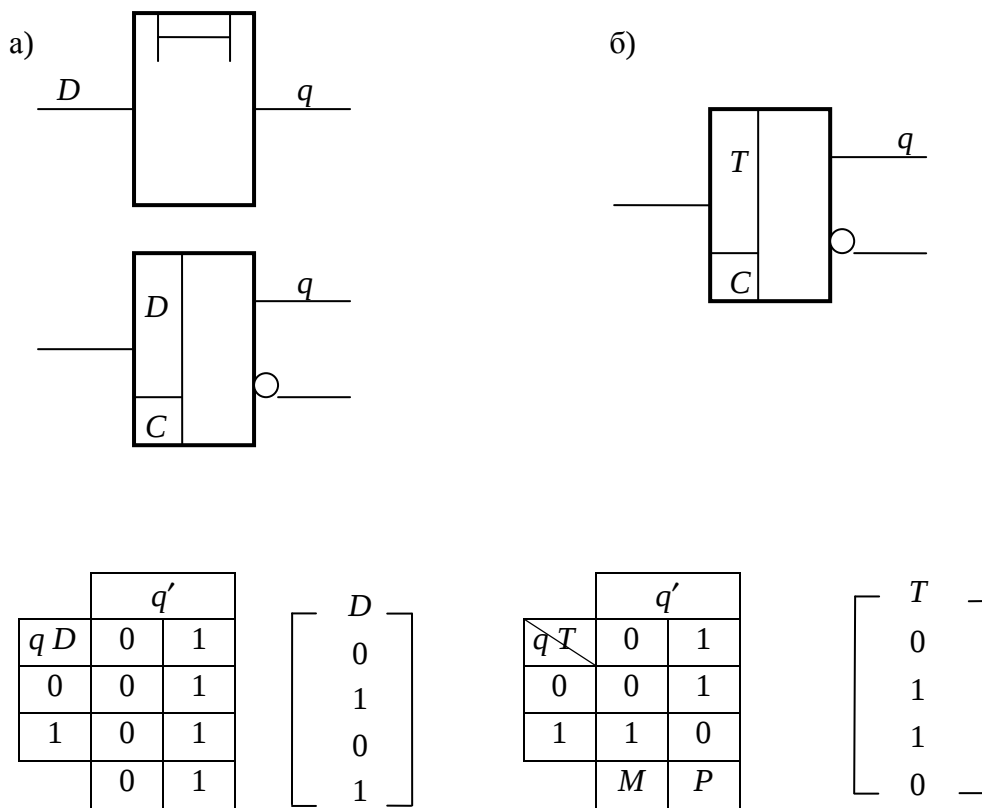
Поскольку нельзя найти возбуждения для переходов $\delta(I, b) = I$ и $\delta(III, c) = III$ образца A в автомате A_1 , то A_1 не является заменителем A , в отличие от A_2 . ■

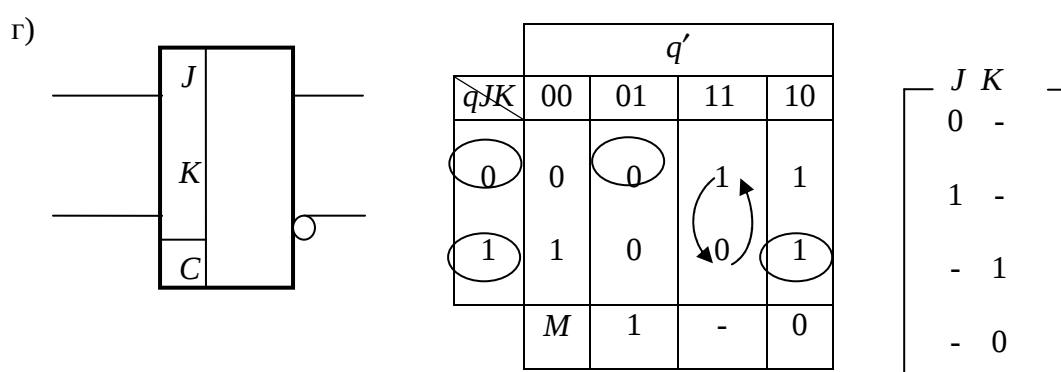
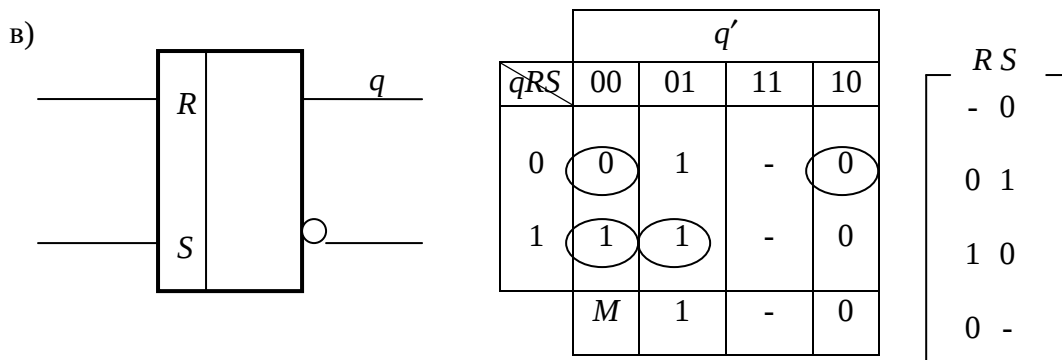
3. Модульные (элементарные) заменители

Само собой разумеется, что при надлежаще подобраной мощности алфавита состояний $|Q|$ заменителя: $|Q| \geq |S|$, естественным требованием оказывается полнота системы всех его переходов между состояниями, т.е. существование для любого перехода заметителя соответствующего возбуждения. Если прототип, как правило, бинарный, то заменитель является параллельным регистром, представляет собой который самую простую по сложности структуру из модульных или элементарных двоичных заменителей, регламентирует которую двоичное кодирование состояний образца А.

В таб. 2. упоминаются преобладающие типы двоичных заменителей таблицами переходов $(\{0,1\} \times \{1,2,\dots,m\} \rightarrow \{0,1\} : \langle q, e_1 2^{m-1} + e_2 2^{m-2} + \dots + e_m 2^0 \rangle \mapsto q')$, оснащенные видами своих столбцов (0, 1, M, P), и матрицами возбуждений $(\{0,1,2,3\} \times \{1,2,\dots,m\} \rightarrow \{0,1\} : \langle q 2^1 + q' 2^0, e_1 2^{m-1} + e_2 2^{m-2} + \dots + e_m 2^0 \rangle \mapsto 0/1)$, где $m = \lceil \lg |E| \rceil$ и $\delta_3(q, e_1 e_2 \dots e) = q'$.

Таб. 2. Двоичные заменители: а) элемент задержки, D – триггер; б) T, в) RS, г) JK триггера.





Заметим, что функция переходов элемента задержки и D – триггера фиктивно зависит от аргумента q , а именно

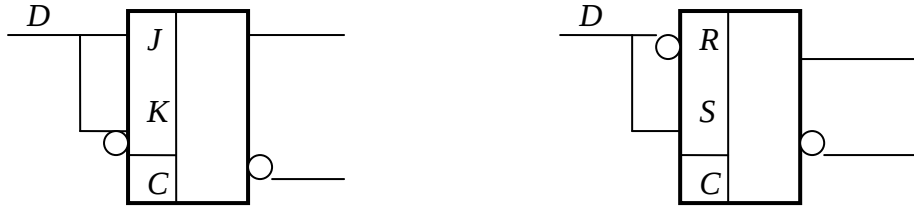
$$D = q' (\text{pred } q'),$$

что с одной стороны свидетельствует об элементарности задержки (но нет D – триггера), ибо её нельзя канонически декомпонировать, но с другой приводит к ошибочной записи q' в схемы вместо возбуждения D . Ясно, что из-за простоты формулы возбуждения D – триггера, проще при синтезе схем найти возбуждение D – триггера, заменив потом его имеющимся триггером.

Пример 2. Располагая возбуждением D – триггера, воспользуйтесь JK и RS двоичным заменителями (таб. 3.); отсюда $J = D$, $K = \overline{D}$ и $R = \overline{D}$, $S = D$. ■

Таб. 3. JK и RS – триггеры вставленные в схему вместо D – триггера из примера 2.

	q'		J		K		R		S	
$q \backslash D$	0	1	0	1	0	1	0	1	0	1
0	0	1	0	1	-	-	-	0	0	1
1	0	1	-	-	1	0	1	0	0	-



Займемся дальше выбором подходящего двоичного модуля заменителя – носителя состоянческой составляющей q_i так, чтобы его функция возбуждения существенно не зависела от им несенной q_i , считая, что задано двоичное кодирование состояний прототипа $A-k: S \rightarrow \{0,1\}^l: s \mapsto q_1 q_2 \dots q_l$, где $l \geq \lceil \lg |S| \rceil$, т.е. $k(\delta(s, x)) = \delta_k(q_1 q_2 \dots q_l, x) \dots = q'_1 q'_2 \dots q'_l = \langle \delta_{31}(q_1, e_1 e_2 \dots e_m), \delta_{32}(q_2, e_1 e_2 \dots e_m), \dots, \delta_{3l}(q_l, e_1 e_2 \dots e_m) \rangle$. Рассмотрим карты функций переходов $q'_i = \delta_{31}(q_i, \lambda_1(q_1 q_2 \dots q_l, x), \lambda_2(q_1 q_2 \dots q_l, x), \dots, \lambda_l(q_1 q_2 \dots q_l, x))$, содержащие множества разных типов столбцов (рис. 3.). Содержит ли функция переходов модуля заменителя то-же самое множество разных видов столбцов как и им несенная состоянческая составляющая q_i , то функция возбуждения модуля имеет аргумент q_i фиктивный (рис. 3.)[4]. Оказывается, что функция возбуждения JK – триггера никогда существенно не зависит от им несенной состоянческой составляющей.

а) $q'_i:$ $J_i:$ $K_i:$

q_i	0	1	0	1	0	1	0	1	-	-	-	-
	0	1	1	0	-	-	-	-	1	0	0	1
	0	1	M	P								

б) $q'_i:$ $R_i:$ $S_i:$

q_i	0	1	0	-	0	-	0	1	0
	0	1	1	1	0	0	0	-	-
	0	1	M						

в) $q'_i:$ $T_i:$ г) $q'_i:$ $D_i:$

q_i	0	1	0	1
	1	0	0	1
	M	P		

q_i	0	1	0	1
	0	1	0	1
	0	1		

Рис. 3. Функция возбуждения триггера а) $J_i K_i$, б) $R_i S_i$, в) T_i , г) D_i не зависящая от q_i .

Пример 3. Пусть задан автомат своей функцией переходов (таб. 4. а)). Построить карты функций переходов модулей заменителей (таб. 4. б.)). Построив карты функций переходов модулей – заменителей (таб. 4. б.)), видим, что подходящими заменителями

суть RS – триггера : $R_1 = \bar{x} \bar{q}_2$, $S_1 = x q_2$ и $R_2 = x q_1$, $S_2 = x \bar{q}_1$ против, например $D_1 = x q_2 \vee x q_1 \vee q_1 q_2$, $D_2 = x \bar{q}_1 \vee \bar{x} q_2$. ■

Таб. 4. а) Таблица переходов, б) карты функций переходов и возбуждений автомата из примера 3.

а)

		$q'_1 q'_2$	
$q_1 q_2$	x	0	1
0 0		0 0	0 1
0 1		0 0	1 0
1 0		0 1	1 1
1 1		1 0	1 1

б)

$q'_1:$				$R_i:$				$K_i:$					
q_2				q_2				q_2					
x				x				x					
q_1		0	0	1	0	-	-	0	-	0	0	1	0
		0	1	1	1	1	0	0	0	0	-	-	-
		0	M	1	M								

$q'_2:$				$R_2:$				$S_2:$					
q_1				q_1				q_1					
x				x				x					
q_2		0	1	0	0	-	0	-	-	0	1	0	0
		1	1	0	1	0	0	1	0	-	-	0	-
		M	1	0	M								

4. Модификации канонической декомпозиции

Автор считает, что нет особого смысла возвращаться к модификациям канонической декомпозиции, выделяются в которых из статического блока I с одной стороны дешифратор состояний, и с другой кроме дешифратора состояний также шифратор возбуждений (Wilkes, Stringer) поскольку обе они рассмотрены в [5]. Следует только добавить [4] что, так как возбуждение, вырабатывает которое шифратор возбуждений, строится по предикации состояния-преемника q'_i модуля заменителя и, таким образом, не входит в состав заменителя T – триггер, ибо для $T_i = 1$ получают $q'_i = 0/1$ и что ещё нужно доопределить матрицы возбуждений RS и JK – триггеров:

$$\begin{array}{cc}
 R & S \\
 \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}
 \end{array}
 \quad
 \begin{array}{cc}
 J & K \\
 \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}
 \end{array}
 .$$

5. Заключение

Автор подчеркивает принципиальную разность между блок – схемами канонической декомпозиции и системы автоматического логического управления, не смотря на их морфологическое сходство. Заменитель в канонической композиции – динамический объект, но управляемая технологическая установка потенциально – динамическая.

Список литературы

- [1] Баранов, С.И.: Синтез микропрограммных автоматов. – Л.: Энергия 1979
- [2] Шалыто, А.А.: Логическое управление. – С.П.: Наука 2000
- [3] Глушков, В.М.: Синтез цифровых автоматов. – М.: Физматгиз 1962
- [4] Bokr, J. – Jáneš, V.: Logické systémy. – Praha: изд. ČVUT 1999
- [5] Майоров, С.А.- Новыков. Г.И.: Структура цифровых вычислительных машин. – Л.: Машиностроение 1970