

Вестник

Томского государственного университета

ПРИЛОЖЕНИЕ

№ 18

АВГУСТ 2006

*Материалы международных,
всероссийских и региональных
научных конференций,
симпозиумов, школ,
проводимых в ТГУ*



каждый из которых реализует одну или несколько булевых функций. Достаточно часто комбинационные схемы реализуют отображение булевых векторов в булевы вектора и описываются посредством систем булевых функций, поэтому в данной работе мы будем говорить о декомпозиции системы булевых функций.

Вектор значений системы булевых функций $\Psi = \{\Psi_1, \Psi_2, \dots, \Psi_k\}$, где каждая функция $\Psi_i, i = 1, \dots, k$, зависит от n переменных; x_1, x_2, \dots, x_n , обозначается Ψ и называется далее *функциональным* вектором.

Введём отношение *уточнения* [1] для двух систем булевых функций, которое далее будем использовать. Пусть системы функций ϕ и Φ определены на множестве B^n булевых векторов длины n . Будем говорить, что функциональный вектор ϕ уточняет функциональный вектор Φ (над булевым пространством B^n), и писать $\phi \sqsupseteq \Phi$, если для любых $a, b \in B^n$ справедливо следующее: если $\Phi(a) \neq \Phi(b)$, то $\phi(a) \neq \phi(b)$.

Будем говорить, что множество ϕ *t-уточняет* функцию $f(\phi \sqsupseteq_t f)$, если $\exists \tau \subseteq \phi$ ($|\tau| \leq t$ и $\tau \sqsupseteq f$).

Функцию $f_i \in \Psi, i = 1, \dots, |\Psi|$, будем называть *ядерной* (относительно системы Ψ), если функция f_i не уточняется множеством $\Psi \setminus \{f_i\}$.

Далее через $Supp(f)$ обозначается множество существенных переменных булевой функции $f \in \Psi$. $Range(\Psi)$ обозначает множество значений системы Ψ .

Следующие две теоремы сформулированы в [1] и содержат необходимые и достаточные условия уточнения одного функционального вектора другим.

Теорема 1 (необходимое условие). Для двух данных функциональных векторов Φ и $\phi, \phi \sqsupseteq \Phi$, только если $Supp(\Phi) \subseteq Supp(\phi)$.

Теорема 2 (необходимое и достаточное условие). Функциональный вектор $\phi \sqsupseteq \Phi$, если и только если выполняется условие $|Range(\phi)| = |Range(\langle \phi, \Phi \rangle)|$.

Обе теоремы, вообще говоря, не содержат достаточно конструктивного алгоритма для проверки, уточняет ли один функциональный вектор другой функциональный вектор.

2. Алгоритмическая реализация декомпозиции методом функциональной редукции

Общий алгоритм декомпозиции системы булевых функций на основе отношения уточнения состоит из следующих шагов:

1. Нахождение минимального подмножества функций $\phi \subseteq \Psi$, которое уточняет все функции множества Ψ .

2. Построение для каждой функции Ψ_j : из множества $\Psi \setminus \phi$ функции зависимости $\theta_j: B^p \rightarrow B$, где $p = |\phi|$.

Для решения задачи, поставленной на первом шаге алгоритма, можно предложить следующий алгоритм. В множестве Ψ определяется множество ядерных функций Δ , которое и объявляется множеством ϕ . Далее рассматриваются функции из множества $\Psi \setminus \phi$. На каждом шаге дополняем множество ϕ функцией f , которая не уточняется текущим множеством ϕ .

Описанный выше алгоритм при числе переменных и функций порядка $25 \div 30$ становится практически нереализуемым по причине большого перебора, который необходимо выполнить для выявления отношения уточнения среди функций множества Ψ .

По указанной выше причине мы модифицируем алгоритм, рассматривая вместо отношения уточнения отношение *t-уточнения* для некоторого заданного (обычно небольшого) числа t .

В модифицированном алгоритме для каждой функции из множества Ψ рассматриваются только подмножества, мощность которых не превышает t .

После того как для каждой функции найдены все минимальные подмножества функций мощности не более t , которые уточняют данную функцию, задача нахождения минимального подмножества ϕ функций, которое уточняет каждую функцию множества Ψ , сводится к нахождению минимального покрытия булевой матрицы. На последнем шаге строятся функции зависимости для каждой функции множества $\Psi \setminus \phi$.

Алгоритмы определения минимального подмножества независимых функций и построения функций зависимости связаны с большим перебором, что накладывает ограничение на размерности декомпозируемых схем. Эти алгоритмы были реализованы программно, и на основе созданного программного обеспечения были проведены компьютерные эксперименты для того, чтобы оценить, как меняются параметры декомпозиции в зависимости от размерности системы булевых функций и параметра t , который используется в отношении *t-уточнения*.

3. Экспериментальные данные

На основе созданного программного обеспечения были проведены компьютерные эксперименты по декомпозиции систем булевых функций с заданными параметрами, которые генерировались случайным образом. В качестве параметров использовались число переменных, число функций и число t , которое используется в отношении *t-уточнения*. Для каждой строки таблицы генерировалось по 10 систем с заданными параметрами, время рассчитывалось как среднее арифметическое от времен каждой из 10 декомпозиций. Значение числа t в экспериментах равно 2, 3, 4. Для представления системы булевых функций использовалось представление Закревского [4]. Каждая функция системы существенно зависела не более чем от 7 переменных.

Результаты экспериментов, представленные в таблице, показывают, что процент систем функций, которые можно декомпозировать описанным выше методом, при возрастании числа переменных и функций в системе существенно зависит от параметра t .

Экспериментальные результаты по декомпозиции систем булевых функций

Число переменных	Число функций	Время, с при $t = 3$	Процент декомпозиции от общего числа функций, %			Средняя мощность множества независимых функций $ Ф $		
			при $t = 2$	при $t = 3$	при $t = 4$	при $t = 2$	при $t = 3$	при $t = 4$
5	5	2	30	30	30	4	4	4
10		9	0	0	0	5	5	5
15		17	0	0	0	5	5	5
20		23	0	0	0	5	5	5
25		41	0	0	0	5	5	5
5	10	2	80	80	80	9	8	6
10		13	20	20	20	9	9	9
15		27	0	0	0	10	10	10
20		35	0	0	0	10	10	10
25		58	0	0	0	10	10	10
5	15	2	90	100	100	11	10	8
10		27	50	50	70	14	14	14
15		55	10	10	10	14	14	14
20		71	0	20	20	15	14	14
25		117	0	0	20	15	15	14
5	20	3	100	100	100	16	13	12
10		47	80	80	100	18	18	17
15		87	40	60	60	19	19	17
20		218	30	30	30	19	19	18
25		470	0	0	10	20	20	19

Из полученных экспериментальных данных можно сделать следующие выводы:

- Зависимость от параметра t становится более существенной при увеличении числа функций в системе.
- Несмотря на то, что при возрастании числа t количество систем, для которых возможна декомпозиция, увеличивается, тем не менее число независимых функций в системе остается близким к общему числу функций, т.е. эффективность такой декомпозиции получается низкой.
- Оценка размерностей системы булевых функций, декомпозиция которых изложенным выше методом производится в реальном времени, следующая: число функций ≤ 25 , число существенных переменных для каждой функции ≤ 7 . При этом, если число переменных > 25 , то декомпозиция оказалась практически невозможной даже при небольшом числе булевых функций.

Заключение

В работе алгоритмизирован и программно реализован метод декомпозиции систем булевых функций (комбинационных схем). С использованием разработанного программного обеспечения были проведены компьютерные эксперименты, которые показали, что возможность декомпозиции существенно зависит от числа t в отношении t -уточнения, которое в экспериментах равно 2, 3, 4. В большинстве случаев эффективность декомпозиции оказалась достаточно низкой. Можно ожидать, что эффективность декомпозиции возрастет при увеличении числа t , но в этом случае необходимо использовать другое представление булевых функций, возможно, представление функций в виде BDD .

ЛИТЕРАТУРА

1. Jie-Hong R. Jiang, Robert K. Brayton. Functional Dependency for Verification Reduction // Work of International Conference. IWLS'2004. – 2004. – P. 184 – 191.
2. Агibalов Г.П., Орaнов А.М. Лекции по теории конечных автоматов. – Томск: Изд-во Том. ун-та, 1984. – 184 с.
3. Баранов С.И., Баркалов А.А. Применение программируемых логических матриц в цифровой технике // Зарубежная радиоэлектроника. – 1982. – № 6.
4. Закревский А.Д., Василькова И.В. Решение больших систем логических уравнений: метод минимизации дерева поиска // Вестник ТГУ. Приложение. – 2002. – № 12. – С. 260 – 265.