Следствие 1. *Пусть $d \geqslant 1$ — заданное число. Задача поиска базиса пространства всех периодов функции алгебры логики степени не выше d по ее многочлену Жегалкина может быть решена полиномиальным алгоритмом относительно числа переменных этой функции.*

Работа поддержана РФФИ в рамках научного проекта № 19-01-00200-а и Минобрнауки РФ в рамках выполнения программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2019-1621.

Список литературы

[1] Логачев О. А., Сальников А. А., Смышляев С. В., Ященко В. В. Булевы функции в теории кодирования и криптологии. — М.: МЦНМО, 2012. — 584 с.

[2] Селезнева С. Н. О сложности распознавания полноты множеств булевых функций, реализованных полиномами Жегалкина // Дискретная математика. — 1997. — Т. 9, вып. 4. — С. 24–31.

[3] Dawson E., Wu C.-K. On the linear structure of symmetric Boolean functions // Australasian Journal of Combinatorics. — 1997. — V. 16. — P. 239–243.

[4] Леонтьев В. К. О некоторых задачах, связанных с булевыми полиномами // Журнал вычислительной математики и математической физики. — 1999. — Т. 39, вып. 6. — С. 1045–1054.

[5] Charpin P., Kyureghyan G. M. On a class of permutation polynomials over F_{2^n} // Lecture Notes of Computer Science. — 2008. — V. 5203. — P. 368–376.

[6] Бухман А. В. О свойствах полиномов периодических функций и сложности распознавания периодичности по полиному булевой функции // Дискретная математика. — 2014. — Т. 26, вып. 1. — С. 21–31.

[7] Yang L., Li H.-W. Investigating the linear structure of Boolean functions based on Simon's period-finding quantum algorithm // https://arxiv.org/pdf/1306.2008.pdf

Программы с запаздыванием

Сергеев Игорь Сергеевич

ФГУП «НИИ «Квант», e-mail: isserg@gmail.com

В современной электронике многие преобразования выполняются конвейерными схемами. Конвейерная схема считывает значения входов и обновляет значения выходов на каждом такте рабочей частоты, но соответствующий входному набору аргументов результат вычисляется с задержкой (латентностью) в несколько тактов. Это побуждает рассмотреть следующую модель вычислений, которую мы назовем *программами с запаздыванием.* Программа с запаздыванием t над базисом B и множеством входных переменных X определяется как разновидность неветвящейся программы:

$$g_1 = f_1(Y_1), \quad g_2 = f_2(Y_2), \quad \ldots, \quad g_k = f_k(Y_k),$$

где $f_i \in B$ и $Y_i \subset X \cup \bigcup_{j \le i-t}\{g_j\}$. При $t = 1$ имеем обычную неветвящуюся программу. При $t > 1$ выбор аргументов для выполнения очередной операции ограничен результатами, полученными не менее t шагов назад.

Как обычно, k (длина последовательности) называется сложностью программы. Программа реализует оператор F, если каждая компонента оператора функционально эквивалентна некоторой функции g_i. Через $C_B^{(t)}(F)$ обозначим *сложность* оператора F — минимальную длину реализующей его программы с запаздыванием t.

Основной вопрос: насколько вычисление конкретного оператора сложнее в модели программ с запаздыванием относительно модели обычных неветвящихся программ. Заметим, что в модели с запаздыванием любые t последовательных функций $g_{i+1}, g_{i+2}, \ldots, g_{i+t}$ вычисляются независимо. Таким образом, модели присуща локальная параллельность.

Очевидно,

$$C_B^{(1)}(F) \le C_B^{(t)}(F) \le t \cdot C_B^{(1)}(F). \tag{1}$$

Рассмотрим несколько простых примеров вычислений над базисом из одной операции умножения $B = \{*\}$. Первые два примера демонстрируют достижимость как оптимистической, так и пессимистической границ (1). Параметр t далее считаем постоянным.

$\boxed{\text{Пример 1.}}$ Возведение в N-ю степень.

Утверждение 1. $C_B^{(t)}\left(x^N\right) \sim t \log_2 N$.

Доказательство. Верхнюю оценку дает метод Брауэра [2] и (1). Для доказательства нижней достаточно заметить, что за серию из t шагов (напомним, что они выполняются независимо) максимум вычисленных показателей степени x^M может быть не более чем удвоен. □

$\boxed{\text{Пример 2.}}$ Вычисление последовательных степеней.

Утверждение 2. $C_B^{(t)}\left(x, x^2, \ldots, x^N\right) \sim N$.

Доказательство. За $t-1$ серий по t шагов легко получить степени $x^2, \ldots, x^t$ (на самом деле, достаточно $\log_2 t$ серий). Далее, в каждой очередной серии из t шагов можно последовательно вычислять степени $x^{(k+1)t}$, $x^{(k+1)t-1}, \ldots,$ x^{kt+1}. Длина описанной программы не превосходит $N + t^2$. □

Из общих соображений ясно, что запаздывание не замедляет существенно вычисление операторов, допускающих эффективные параллельные реализации. Напомним, что *глубина* программы определяется как глубина изображающего ее графа (схемы), т. е. как длина максимального ориентированного пути от входа к выходу.

Лемма 1. *Если оператор F вычисляется неветвящейся программой над базисом B со сложностью C и глубиной D, то $C_B^{(t)}(F) \le C + tD$.*

Доказательство. Программу можно разбить на D подпрограмм (слоев), состоящих из независимых (параллельно выполняемых) шагов. Подпрограмма из k независимых шагов тривиально реализуется $\lceil k/t \rceil$ сериями из t шагов в модели с запаздыванием t, откуда следует требуемая оценка. $\square$

Утверждение 2 выводится из леммы 1, поскольку последовательные N степеней переменной x легко вычислить программой сложности $N-1$ и глубины $\lceil \log_2 N \rceil$.

В свете сказанного вопрос о замедлении вычислений в модели программ с запаздыванием содержателен для операторов, программы оптимальной сложности для которых имеют глубину, по порядку совпадающую со сложностью. Следующие два примера иллюстрируют эту ситуацию.

$\boxed{\text{Пример 3.}}$ Вычисление префиксов $\pi_i = x_1 \cdot \ldots \cdot x_i$, $1 \le i \le N$.

Утверждение 3. $C_B^{(t)}(\pi_1, \ldots, \pi_N) \sim 2tN/(t+1)$.

Доказательство. Докажем нижнюю оценку. Разобьем программу на серии из t операций. Предположим, что вычисление укладывается в k серий. Пусть программа вычисляет максимальное произведение $x_1 \cdot \ldots \cdot x_N$ как $x_1 \cdot p_1 \cdot \ldots \cdot p_k$, где p_i — множитель, добавляемый в i-й серии. Допускается, что некоторые p_i могут быть равны 1.

Вычисление всех p_i требует не менее $N-1-k$ шагов программы. Учитывая, что еще минимум $N-1$ шагов требуется для вычисления собственно префиксов, длина программы оценивается как $2(N-1)-k$. Получаем оценку $kt \ge 2(N-1)-k$, откуда следует $k \ge 2(N-1)/(t+1)$.

Иначе, нижнюю оценку можно получить, отталкиваясь от известного соотношения $C + D \ge 2N - 2$ для сложности C и глубины D программ, реализующих префиксы N переменных, см. [3]. Поскольку для программы с запаздыванием t выполнено $D \le C/t + 1$, то $(1 + 1/t)C \gtrsim 2N$.

Покажем, что оценка достижима. Пусть для простоты $N = k(t+1)$. Максимальный префикс вычисляется по формуле

$$\pi_N = x_1 \cdot p_1 \cdot x_{t+2} \cdot p_2 \cdot x_{2(t+1)+1} \cdot \ldots \cdot p_k, \qquad p_i = x_{(i-1)(t+1)+2} \cdot \ldots \cdot x_{i(t+1)}.$$

Обозначим $p_{i,j} = x_{(i-1)(t+1)+2} \cdot \ldots \cdot x_{(i-1)(t+1)+j+1}$ — промежуточные стадии вычисления p_i (при этом $p_i = p_{i,t}$). Остальные префиксы получаются как

$$\pi_{i(t+1)+1} = \pi_{i(t+1)} \cdot x_{i(t+1)+1}, \quad \pi_{i(t+1)+j} = \pi_{i(t+1)+1} \cdot p_{i+1,j-1}, \quad j = 2, \ldots, t+1.$$

Программу можно составить, последовательно объединяя серии следующего вида при i от $2-t$ до $k-1$:

$$p_{i+t-1,2},\ p_{i+t-2,3},\ \ldots,\ p_{i+1,t},\ \pi_{i(t+1)+1},$$
$$\pi_{i(t+1)-1},\ \pi_{i(t+1)-2},\ \ldots,\ \pi_{i(t+1)-t+1},\ \pi_{(i+1)(t+1)}.$$

Неопределенные величины из программы исключаются (заменяются чем угодно). Длина программы равна $2t(k+t-2) = 2N/(t+1) + O(t^2)$. $\square$

Сделаем еще одно элементарное наблюдение. Пусть $X_1, \ldots, X_t$ — равномощные группы переменных. Очевидно,

$$C_B^{(t)}(F(X_1), \ldots, F(X_t)) \le t \cdot C_B^{(1)}(F). \tag{2}$$

В следующем примере (несмотря на схожесть с предыдущим) свойство (2) позволяет избежать существенных потерь во времени при переходе к модели программ с запаздыванием.

Пример 4. Вычисление дополняющих произведений $c_i = \prod_{j \ne i} x_j$, $1 \le i \le N$.

Утверждение 4. $C_B^{(t)}(c_1, \ldots, c_N) \sim 3N$.

Доказательство. Сложность системы в модели без запаздывания равна $3N-6$, см. [1]. Схема, доставляющая оптимальную оценку в этой модели, состоит из последовательной части (вычисление префиксов $\pi_i = x_1 \cdot \ldots \cdot x_i$ и суффиксов $\sigma_i = x_i \cdot \ldots \cdot x_N$) и параллельной части (произведения суффиксов и префиксов). Но эту схему можно перестроить в более параллельную.

Пусть $N = tk$. Разобьем множество переменных на t групп $X_j = (x_{(j-1)k+1}, \ldots, x_{jk})$, $j = 1, \ldots, t$, мощности k. Обозначим префиксные и суффиксные произведения в каждой из групп через $\pi_{i,j} = \pi_i(X_j)$ и $\sigma_{i,j} = \sigma_i(X_j)$. Положим формально $\pi_{0,j} = \sigma_{k+1,j} = 1$. Через $p_j = \pi_{k,j} = \sigma_{1,j}$ обозначим произведение всех переменных группы X_j.

Если $i = (j-1)k + l$, где $1 \le l \le k$, то c_i можно вычислить по формуле

$$c_i(x_1, \ldots, x_N) = c_j(p_1, \ldots, p_t) \cdot \pi_{l-1,j} \cdot \sigma_{l+1,j}. \tag{3}$$

Множество всех суффиксов k переменных вычисляется тривиально неветвящейся программой длины $k-1$. Поэтому согласно (2) все $\sigma_{i,j}$, и среди них p_j, могут быть вычислены программой с запаздыванием t длины $t(k-1)$.

Далее, за $t-1$ серий по t шагов можно вычислить все дополняющие группы X_j произведения $u_j = c_j(p_1, \ldots, p_t)$.

При каждом j произведения $u_j \cdot \pi_{l-1,j}$, $1 \le l \le k$, образуют систему префиксов $\{\pi_i(u_j, X_j) \mid 1 \le i \le k\}$. Согласно (2), они вычисляются программой длины $t(k-1)$.

Теперь любое произведение (3) может быть получено одним умножением выражения $u_j \cdot \pi_{l-1,j}$, найденного на предыдущем шаге, и суффикса $\sigma_{l+1,j}$. Эти умножения независимы, поэтому могут быть выполнены за N шагов. Общая длина программы не превосходит $2t(k-1) + N + t^2 = 3N + O(t^2)$. $\square$

Иначе, этот результат можно получить как следствие из леммы 1: система дополняющих произведений N переменных вычисляется неветвящейся программой сложности $3N + o(N)$ и глубины $o(N)$, что по сути и доказано в утверждении 4.

Список литературы

[1] Чашкин А. В. О сложности булевых матриц, графов и соответствующих им булевых функций // Дискретная математика. — 1994. — Т. 6(2). — С. 43–73.

[2] Brauer A. On addition chains // Bull. AMS. — 1939. — V. 45. — P. 736–739.

[3] Snir M. Depth-size trade-offs for parallel prefix computation // J. Algorithms. — 1986. — V. 4. — P. 185–201.

О длине минимальных единичных проверяющих тестов относительно замен функциональных элементов на инверторы в произвольном полном базисе

Темербекова Гульгайша Габдуловна, Романов Дмитрий Сергеевич

Московский государственный университет имени М. В. Ломоносова, e-mail: gulgaisha93@mail.ru, romanov@cs.msu.ru

Рассмотрим тестирование схем из функциональных элементов (СФЭ), реализующих произвольные булевы функции. Пусть имеется СФЭ S, реализующая булеву функцию $f(\tilde{x}^n)$, где $x^n = (x_1, x_2, \ldots, x_n)$. Пусть на S воздействует источник неисправностей U так, что один или несколько элементов схемы S переходят в неисправное состояние. Тогда схема S вместо исходной функции $f(\tilde{x}^n)$ будет реализовывать некоторую, возможно, от-