OpenMP, которая позволяет задействовать все ядра центрального процессора (в нашем случае 4 ядра). В таблице показано время работы (в секундах) соответствующих реализаций.

	CPU		GPU
n	Посл. алг.	OpenMP	CUDA
15	1,02	0,25	0,01
16	4,13	1,03	0,04
17	16,9	4,22	0,15
18	69,9	17,4	0,65

Таким образом, реализация для видеокарты с использованием технологии CUDA оказалась быстрее, чем последовательный алгоритм, примерно в 106 раз, а в сравнении с параллельной реализацией на центральном процессоре — примерно в 26,5 раз.

ЛИТЕРАТУРА

1. *Агибалов Г. П.* Избранные теоремы начального курса криптографии. Томск: Изд-во НТЛ, 2005.
2. `http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf`
3. `http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_BestPractices.pdf`

УДК 519.7

О СХОДИМОСТИ ГИБРИДНОГО SAT+ROBDD-ЛОГИЧЕСКОГО ВЫВОДА[1]

А. А. Семенов, А. С. Игнатьев

Проблема обращения дискретных функций возникает во многих теоретических и прикладных областях современной кибернетики. Пусть $f : \{0,1\}^* \to \{0,1\}^*$ — вычислимая детерминированным образом за полиномиальное от длины входа время дискретная функция. Если $A(f)$ — некоторый полиномиальный алгоритм, вычисляющий f, то $A(f)$ задает семейство функций вида $f_n : \{0,1\}^n \to \{0,1\}^*, n \in N$. Задача обращения произвольной функции f_n из данного семейства состоит в следующем: известно $y \in \mathrm{Range}\, f_n$, требуется, зная текст программы $A(f)$, найти произвольный $x \in \{0,1\}^n$, такой, что $f_n(x) = y$. Описанная задача является вычислительно трудной в общей постановке — она не может быть решена за полиномиальное время в предположении, что $P \neq NP$. Поскольку различные практические задачи могут рассматриваться как частные случаи сформулированной, разработка вычислительных алгоритмов для её решения является актуальной областью.

Описанную проблему можно сводить к SAT-задачам [1], используя эффективные алгоритмы трансляции программ в булевы уравнения (см., например, [2]). Для решения получаемых SAT-задач можно использовать различные методы, лучшие из которых базируются на алгоритме DPLL. В КНФ, получаемой в процессе трансляции алгоритма $A(f_n)$, можно выделить множество X_n, состоящее из n так называемых «переменных входа» рассматриваемой функции. Мощность данного множества равна n. В [3] говорится о том, что X_n является «сильной формой» множества-лазейки для алгоритма DPLL, который применяется к КНФ $C(f_n)$, кодирующей задачу обращения f_n

[1]Работа выполнена при поддержке гранта РФФИ № 11-07-00377-а.

в произвольной точке (в [3] используется термин «*strong unit propagation backdoor set*», SUPBS). Данный факт в [3] полагается «интуитивно ясным» и приводится без доказательства. В работе [4] дано понятие ядра DPLL-вывода, которое является по сути аналогом понятия SUPBS, а также доказано, что множество переменных, кодирующих вход рассматриваемой функции f_n, образует ядро DPLL-вывода в применении к КНФ $C(f_n)$. Это означает, что ограниченный вариант нехронологического DPLL, который выбирает переменные только из X_n, является полным и может использоваться для решения задачи обращения рассматриваемой функции. Такой алгоритм получил название core-DPLL. В [4] отмечено, что core-DPLL должен порождать избыточные ограничения. Данный тезис целиком подтвердился вычислительными экспериментами. В связи с этим в работе [5] описан гибридный SAT+ROBDD-логический вывод, в котором core-DPLL используется для накопления баз конфликтных дизъюнктов, составленных из литералов только над переменными ядра. Каждая такая конфликтная база имеет вид КНФ и задает тем самым булеву функцию, для которой строится её ROBDD-представление. В дальнейшем вывод работает как на исходной КНФ, так и на конфликтной части, но представленной в форме ROBDD. Для этой цели в [5] введены аналоги основных механизмов вывода, используемых в современных SAT-решателях, основанных на DPLL. В [5] описан также новый SAT-решатель с параллельной архитектурой, в котором реализован гибридный SAT+ROBDD-вывод. Данный решатель существенно превзошёл по эффективности известные решатели на SAT-задачах, кодирующих обращение некоторых криптографических функций.

В докладе представлено новое свойство гибридного SAT+ROBDD-вывода, а именно его сходимость относительно числа путей в терминальную единицу в ROBDD, представляющей базу конфликтных ограничений.

Гибридная SAT+ROBDD-стратегия логического вывода в применении к $C(f_n)$ представляет собой процесс, разделенный на итерации [5]. На начальной итерации core-DPLL накапливает некоторое множество конфликтных дизъюнктов — $D_1^1, \ldots, D_{t_1}^1$. Далее строится ROBDD B_1, представляющая булеву функцию, заданную КНФ $D_1^1 \cdot \ldots \cdot D_{t_1}^1$. Дальнейший вывод является гибридным — задействуется как КНФ-часть (собственно $C(f_n)$), так и ROBDD-часть. Если B_r — ROBDD-представление конфликтной базы, построенной на итерации с номером r, а на итерации с номером $r+1$ гибридный вывод порождает дизъюнкты $D_1^{r+1}, \ldots, D_{t_{r+1}}^{r+1}$, то ROBDD B_{r+1} есть

$$B_{r+1} = \mathrm{Apply}(B_r \cdot D_1^{r+1} \cdot \ldots \cdot D_{t_{r+1}}^{r+1})$$

(используется алгоритм Apply [6]).

Через $\#_X B(g)$ обозначим число наборов значений истинности, на которых булева функция g над множеством переменных X, представляемая ROBDD $B(g)$, принимает значение 1. Справедлив следующий результат.

Теорема 1. Пусть C — произвольная КНФ над множеством X. Предположим, что на C осуществляется s итераций гибридного SAT+ROBDD-вывода и $B_1, \ldots, B_s$ — ROBDD-представления баз конфликтных дизъюнктов, построенные на соответствующих итерациях. Если для некоторого $k \in \{1, \ldots, s\}$ множество вершин B_k исчерпывается терминальным нулем, то C невыполнима. В противном случае для произвольного $k \in \{2, \ldots, s\}$ имеет место $\#_X B_k < \#_X B_{k-1}$.

ЛИТЕРАТУРА

1. *Biere A., Heule M., van Maaren H., and Walsh T.* Handbook of Satisfiability. IOS Press, 2009.

2. *Отпущенников И. В., Семенов А. А.* Технология трансляции комбинаторных проблем в булевы уравнения // Прикладная дискретная математика. 2011. № 1. С. 96–115.
3. *Jarvisalo M. and Junttila T.* Limitations of restricted branching in clause learning // Constraints. 2009. V. 14. No. 3. P. 325–356.
4. *Семенов А. А.* Декомпозиционные представления логических уравнений в задачах обращения дискретных функций // Изв. РАН. Теория и системы управления. 2009. № 5. С. 47–61.
5. *Ignatiev A. S. and Semenov A. A.* DPLL+ROBDD derivation applied to inversion of some cryptographic functions // LNCS. 2011. V. 6695. P. 76–89.
6. *Bryant R. E.* Graph-Based Algorithms for Boolean Function Manipulation // IEEE Trans. Comput. 1986. V. 35. No. 8. P. 677–691.

УДК 004.056.5:512.545

РЕАЛИЗАЦИЯ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ОРТОГОНАЛИЗАЦИИ В ЗАДАЧЕ ПОИСКА КРАТЧАЙШЕГО БАЗИСА ЦЕЛОЧИСЛЕННЫХ РЕШЁТОК

В. С. Усатюк

Целью работы является демонстрация увеличения производительности алгоритмов приведения базиса целочисленных решёток за счёт замены рекуррентного алгоритма Грама — Шмидта параллельными алгоритмами ортогонализации.

Для приведения базиса решётки с экспоненциальной точностью $\gamma = 2^{(n-1)/2}$ достаточно привести базис к $(\delta - LLL)$-редуцированному базису, применив полиномиальный по временной сложности алгоритм Ленстра — Ленстра — Ловаса (LLL) [1]. Для приведения базиса решётки с точностью $\gamma \in \left[2^{(n-1)/2} - \varepsilon, 1\right]$ достаточно привести подмножество базиса решётки, состоящее из $\beta \leqslant n$ векторов, к базису Коркина — Золотарева, применив блочный алгоритм Коркина — Золотарева (BKZ), чья временная сложность зависит от размера блока, изменяясь от полиномиальной до экспоненциальной [1].

Ключевой частью алгоритмов приведения базиса решёток является этап ортогонализации, осуществляемый при помощи алгоритмов, вычисляющих QR-разложение матрицы базисных векторов. Традиционно, в силу своей геометрической наглядности и простоты, для ортогонализации используется рекуррентный алгоритм Грама — Шмидта или его вычислительно устойчивый аналог — модифицированный алгоритм Грама — Шмидта [2]. Однако рекуррентная природа данного алгоритма препятствует его распараллеливанию и делает его «узким местом» процедуры приведения базиса решётки. Алгоритм Грама — Шмидта может быть заменён другими алгоритмами, осуществляющими QR-разложение, а именно алгоритмом отражения Хаусхолдера или алгоритмом вращения Гивенса [3]. Их применение теоретически позволяет ускорить процесс ортогонализации $(n \times n)$-матрицы базиса решётки в n раз.

В основе метода Гивенса лежит идея поворота векторов матрицы базиса с целью последовательного обнуления координат векторов ортогонализуемого базиса. Одной из ключевых особенностей алгоритма Гивенса является необходимость вычисления квадратного корня и в два раза большее число операций по сравнению с алгоритмом Хаусхолдера. Однако этот недостаток компенсируется отсутствием ветвления, что приводит к высокой эффективности исполнения данного алгоритма на векторных вычислительных устройствах, в частности на видеокартах. Последнее обстоятельство привело к реализации именно этого алгоритма в библиотеке CUBLAS [4].