ПАРАЛЛЕЛЬНЫЕ ГИБРИДНЫЕ (SAT+ROBDD) АЛГОРИТМЫ В ЗАДАЧАХ ОБРАЩЕНИЯ ДИСКРЕТНЫХ ФУНКЦИЙ

А.С. Игнатьев
Институт динамики систем и теории управления СО РАН
Россия, 664033, Иркутск, ул. Лермонтова, 134
alexey.ignatiev@gmail.com

А.А. Семенов
Институт динамики систем и теории управления СО РАН
Россия, 664033, Иркутск, ул. Лермонтова, 134
biclop@rambler.ru

В статье описана стратегия гибридного логического вывода, в котором нехронологический DPLL комбинируется с техникой модификации баз накапливаемых ограничений-дизъюнктов при помощи двоичных диаграмм решений (точнее, ROBDD). Данный подход был протестирован на задачах обращения некоторых криптографических функций.

PARALLEL (SAT+ROBDD) ALGORITHMS IN PROBLEMS OF DISCRETE FUNCTIONS INVERSION / A.S. Ignatiev (Institute for system dynamics and control theory SB RAS, 144 Lermontova, Irkutsk 664033, Russia, E-mail: alexey.ignatiev@gmail.com), A.A. Semenov (Institute for system dynamics and control theory SB RAS, 144 Lermontova, Irkutsk 664033, Russia, E-mail: biclop@rambler.ru).

The paper describes a strategy of a hybrid derivation combining DPLL derivation method (Davis-Putnam-Logemann-Loveland algorithm) and a derivation technique based on binary decision diagrams (or more precisely ROBDDs). This approach can be applied to the inversion of some cryptographic functions.

1. Введение

В последние годы заметен рост интереса к символьным алгоритмам, показывающим высокую эффективность на важных в практическом отношении классах булевых уравнений. Особенно впечатляет прогресс в разработке таких алгоритмов для решения SAT-задач [1]. Этим вопросам посвящены многочисленные конференции и специализированные издания [2]. SAT-задачи находят широкое применение в различных разделах прикладной дискретной математики и кибернетики: синтез и верификация дискретных автоматов, верификация программных логик, криптоанализ, задачи биоинформатики и во многих других областях.

Наибольшую эффективность в решении SAT-задач (по результатам специализированных конкурсов [1]) демонстрируют алгоритмы, использующие в своей основе DPLL [3] и последующие его модернизации [4]–[6]. Главное отличие поздних версий от классического DPLL в том, что в них информация о ходе вывода хранится в форме булевых ограничений-дизъюнктов, запрещающих конфликтные присвоения. В связи с этим одной из центральных в современных SAT-решателях является проблема переполнения памяти генерируемыми в процессе поиска булевыми ограничениями. На практике данная проблема решается при помощи процедур чистки баз ограничений, в результате которых признаваемые нерелевантными ограничения попросту отбрасываются. Однако все заключения о релевантности носят характер эвристик и, как следствие, отбрасывание ограничений в общем случае приводит к потере алгоритмом полноты ввиду отсутствия гарантии конечности его работы на произвольных КНФ. Эффект потери полноты наблюдался на некоторых

криптографических тестах и приводил при крупноблочном распараллеливании соответствующих SAT-задач к «сверхлинейному» ускорению [7].

В статье [8] для решения проблемы переполнения памяти на задачах обращения полиномиально вычислимых дискретных функций был предложен гибридный подход, в котором нехронологический DPLL-вывод сочетается с выводом на двоичных диаграммах решений (BDD), а точнее, на сокращенных упорядоченных BDD или ROBDD. В статье [9] были приведены алгоритмы работы с ROBDD как с базами булевых ограничений (алгоритмы подстановки и вывода значений логических переменных).

В настоящей работе будет описана архитектура и результаты тестирования нового параллельного гибридного (SAT+ROBDD) решателя булевых уравнений, ориентированного на решение задач обращения полиномиально вычислимых дискретных функций.

Основной момент новизны составляет технология обмена булевыми ограничениями в распределенных вычислительных средах (РВС). Отличие данного решателя от имеющихся параллельных SAT-решателей (например, [10]) заключается в том, что по сети передаются массивы накапливаемых на вычислительных узлах булевых ограничений-дизъюнктов, представленные в виде ROBDD. При этом переход от КНФ, образованной накопленными конфликтными дизъюнктами, к ROBDD-представлению соответствующей булевой функции позволяет уменьшить объем используемой памяти в тысячи раз. Малый размер баз конфликтных ограничений, представленных в виде ROBDD, делает реальным обмен этими ограничениями по сети РВС.

Описанный решатель был реализован с применением стандарта MPI и протестирован на задачах обращения некоторых криптографических функций. На данных тестах новый решатель превзошел ряд решателей, с которыми проводилось сравнение (в том числе оказался эффективнее [10] в среднем более чем в три раза).

Следует особо отметить принципиальную возможность функционирования нового решателя в средах с произвольным числом вычислительных единиц (например, в [10] такая возможность лишь задекларирована, однако соответствующие программные реализации не представлены в открытых источниках).

2. Алгоритмика обращения некоторых дискретных функций

2.1. Проблема обращения полиномиально вычислимых дискретных функций. Обозначим через $\Im$ класс, образованный натуральными семействами вычислимых за полиномиальное время дискретных функций вида $f:\{0,1\}^n \to \{0,1\}^*$. Проблема обращения $f \in \Im$ в произвольной точке $y \in range\ f$ ставится следующим образом: зная y и алгоритм вычисления f, найти $x \in \{0,1\}^n : f(x) = y$. Используя фундаментальную идею С.Кука (см. [11]), можно свести данную задачу к задаче поиска выполняющего набора выполнимой КНФ $C(f)$ над множеством булевых переменных $\tilde{X} = \{x_1,...,x_{q(n)}\}$, $q(\cdot)$– некоторый полином. Именно этот факт лежит в основе многочисленных примеров применения SAT-решателей к задачам обращения дискретных функций. Про КНФ $C(f)$ (которую также будем обозначать через $C(x_1,...,x_{q(n)})$) будем говорить, что она кодирует задачу обращения функции f в точке $y \in range\ f$.

2.2 SAT-задачи. К SAT-задачам относятся задачи поиска решений булевых уравнений вида

$$C(x_1,...,x_k) = 1,$$

где $C(x_1,...,x_k)$– формула исчисления высказываний (ИВ), имеющая вид конъюнктивной нормальной формы (КНФ), $x_1,...,x_k$– булевы переменные. В общей постановке SAT-задачи NP-трудны, однако к ним сводится настолько обширное множество практически важных задач, что построение эвристических алгоритмов, эффективных на тех или иных классах SAT-задач, является одной из актуальных областей современной компьютерной алгебры.

Наиболее эффективные на сегодняшний день SAT-решатели используют в своей основе алгоритм DPLL [3] и дальнейшие его модификации. Кратко перечислим их основные алгоритмические компоненты: алгоритм DPLL и его составляющие (правило единичного дизъюнкта, BCP – стратегия распространения булевых ограничений); графы анализа вывода, процедура «Clause Learning» (далее CL-процедура), смысл которой заключается в конъюнктивном приписывании к текущей КНФ новых ограничений-дизъюнктов, запрещающих приведшие к конфликтам присвоения [4]; процедуры анализа конфликтов и построения конфликтных дизъюнктов [4], [12], их предназначение – синтез новых булевых ограничений-дизъюнктов; механизмы отсроченных вычислений (типа «watched literals», [13]), с их помощью сокращается время, затрачиваемое на подстановку в КНФ значений переменных (в некоторые дизъюнкты, не удовлетворяющие определенным признакам, подстановка не осуществляется).

Использование SAT-решателей с перечисленными компонентами оказалось оправданным даже на таких аргументированно трудных задачах как задачи обращения некоторых криптографических функций [14]. При этом следует отметить выгодные свойства перечисленных алгоритмов при их крупноблочном распараллеливании [15-16].

2.3. Двоичные диаграммы решений и алгоритмы манипулирования булевыми функциями на их основе. Двоичные диаграммы решений (BDD) были введены К. Ли в статье [17]. Фундаментальность этой структуры данных для дискретной математики была осознана после выхода работы Р. Брайанта [18], в которой он описал семейство алгоритмов манипулирования булевыми функциями при помощи BDD. Одним из главных теоретических результатов [18] является теорема о каноническом представлении булевых функций в виде ROBDD – сокращенных BDD, не содержащих повторяющихся фрагментов. Структура ROBDD может рассматриваться как некоторая «плотная упаковка» дерева решений рассматриваемой функции. Повторы вершин в ROBDD не допускаются – дубликаты уже существующих вершин заменяются ссылками на них.

Основным алгоритмом при работе ROBDD является *Apply*, описанный Р. Брайантом в [18]. Данный алгоритм позволяет на основе ROBDD-представлений $B(f_1)$ и $B(f_2)$ булевых функций f_1 и f_2 построить ROBDD-представление функции $f_1 * f_2$, где $*$–произвольная бинарная логическая связка. При совпадении порядка означивания переменных в $B(f_1)$ и $B(f_2)$ сложность *Apply* ограничена сверху величиной $O(|B(f_1)| \cdot |B(f_2)|)$ (здесь и далее через $|B|$ обозначается число вершин в ROBDD B).

2.4 Гибридный подход к обращению функций из $\Im$. Сложно отследить первое появление идеи совместного использования SAT и ROBDD. В качестве одной из наиболее ранних работ можно указать [19], в которой предлагалось строить ROBDD-представление остаточной КНФ после нескольких итераций BCP. Однако идея использовать ROBDD для представления баз конфликтных ограничений в применении к задачам обращения дискретных функций, по-видимому, впервые была предложена в [8]. Кратко остановимся на основных результатах данной работы.

Будем рассматривать задачу обращения произвольной функции $f:\{0,1\}^n \to \{0,1\}^*$, $f \in \Im$. Используя преобразования Цейтина, сведем данную к некоторой SAT-задаче. Нетрудно показать, что в соответствующей КНФ $C(f)$ можно выделить подмножество булевых переменных, от которых «функционально зависят» все остальные переменные, фигурирующие в данной КНФ. Это множество, обозначаемое далее через X, образовано булевыми переменными, кодирующими входное слово из $\{0,1\}^n$. Как правило, число n существенно меньше общего числа переменных в $C(f)$. Однако можно показать, что если угадывать переменные только из X, используя BCP, CL-процедуру и рестарты, то получаемая стратегия (далее «K-стратегия» от «Kernel») является полной (то есть за конечное число шагов по известному $y \in range\ f$ будет найден $x \in \{0,1\}^n : f(x) = y$).

Сказанное означает, что конфликтные ограничения, порождаемые в процессе K-стратегии, состоят только из литералов над X. В [8] рассматривались системы булевых уравнений вида

$$\begin{cases} C(f)=1 \\ D_1\left(x_1^1,...,\ x_{r_1}^1\right)=1 \\ ..... \ ..... \ .... \\ D_Q\left(x_1^Q,...,\ x_{r_Q}^Q\right)=1 \end{cases}$$

Здесь $C(f)$– КНФ, кодирующая задачу обращений f в точке $y \in range\ f$, а $D_j(...), j \in \{1,...,Q\}$– конфликтные дизъюнкты, накопленные за Q итераций K-стратегии, $\bigcup_{i=1}^{Q}\left\{x_1^i,...,x_{r_i}^i\right\} \subseteq X$. Предлагалось вместо системы ограничений

$$\begin{cases} D_1\left(x_1^1,...,x_{r_1}^1\right)=1 \\ ..... \\ D_Q\left(x_1^Q,...,x_{r_Q}^Q\right)=1 \end{cases}$$

рассматривать ROBDD-представление булевой функции δ_Q, заданной формулой

$$D_1\left(x_1^1,...,x_{r_1}^1\right)\cdot ... \cdot D_Q\left(x_1^Q,...,x_{r_Q}^Q\right). \qquad (1)$$

В [8] была доказана следующая теорема.

Теорема 1 (см. [8]).

Пусть $f:\{0,1\}^n \to \{0,1\}^*$– произвольная функция из класса $\Im$, и $\alpha=(\alpha_1,...,\alpha_n)$ – произвольное решение задачи обращения f в некоторой точке $y \in range\ f$. Пусть $B(\delta_Q)$ – ROBDD-представление функции δ_Q системы (1) в контексте рассматриваемой задачи. Тогда существует такой путь π из корня B в терминальную вершину «1» , что $\alpha \in A(\pi)$, где через $A(\pi)$ обозначено множество векторов из $\{0,1\}^n$, определяемых путем π.

2.5 Алгоритмы работы с ROBDD как с базами булевых ограничений. Многочисленные вычислительные эксперименты показали, что генерируемые K-стратегией массивы конфликтных ограничений-дизъюнктов имеют очень компактные ROBDD-представления: сотни мегабайт ограничений «сжимались» в ROBDD на нескольких десятках вершин (несколько килобайт в памяти ЭВМ). В такой ситуации нет никакой необходимости отбрасывать ограничения. Этот факт дает основу для нового подхода к обращению функций из $\Im$. В рамках данного подхода используется гибридный логический вывод: в процессе вывода идет работа как с исходной КНФ, так и с ROBDD-представлениями баз конфликтных дизъюнктов, генерируемых DPLL-выводом (K-стратегией). Тем самым возникает задача описания и реализации алгоритмов работы с ROBDD, рассматриваемыми в роли баз булевых ограничений. Семейство алгоритмов логического вывода на ROBDD было приведено в статье [9].

В процессе вывода в ROBDD осуществляются подстановки значений булевых переменных. При этом могут возникать ситуации, когда значения ряда переменных определяются структурой текущей ROBDD однозначно. Таки ситуации называются ROBDD-следствиями. Вывод значения переменной в соответствии с ROBDD-следствием аналогично срабатыванию правила единичного дизъюнкта в DPLL-выводе. В [9] было показано, что если B– ROBDD, представляющая булеву функцию от n переменных, то подстановка в нее значения произвольной переменной и вывод всевозможных ROBDD-следствий требует времени $O(n \cdot |B|)$. В этой же работе были описаны алгоритмы организации на ROBDD «отсроченных» вычислений (по аналогии с «watched literals», [13]), а также приведен новый

алгоритм модификации построенной ROBDD в соответствии с новым порядком означивания переменных.

3. Параллельный гибридный (SAT+ROBDD)-решатель и его применение к задачам обращения криптографических функций

С практических позиций весьма привлекательно то, что базы дизъюнктов, получаемые при реализации K-стратегиии, имеют компактные ROBDD-представления. Это позволяет организовать обмен ограничениями, накапливаемыми на независимо работающих вычислительных узлах.

Первые параллельные SAT-решатели были представлены на конкурсе SAT-race 2008 года [20]. В этих решателях также реализован обмен ограничениями между вычислительными узлами, однако ограничения передаются непосредственно в виде дизъюнктов. Такой подход не позволяет узлам кластера обмениваться большими объемами информации.

Далее описывается архитектура параллельного гибридного (SAT+ROBDD) решателя, реализованного в соответствии со стандартом MPI и протестированного на задачах обращения некоторых криптографических функций.

Основу параллельного решателя составляет последовательный гибридный решатель, названный «hsat». Данный решатель функционирует в соответствии со схемой, представленной на следующем рисунке (логический вывод ведется как на исходной КНФ, так и на ROBDD, представляющей базу накопленных ограничений-дизъюнктов).

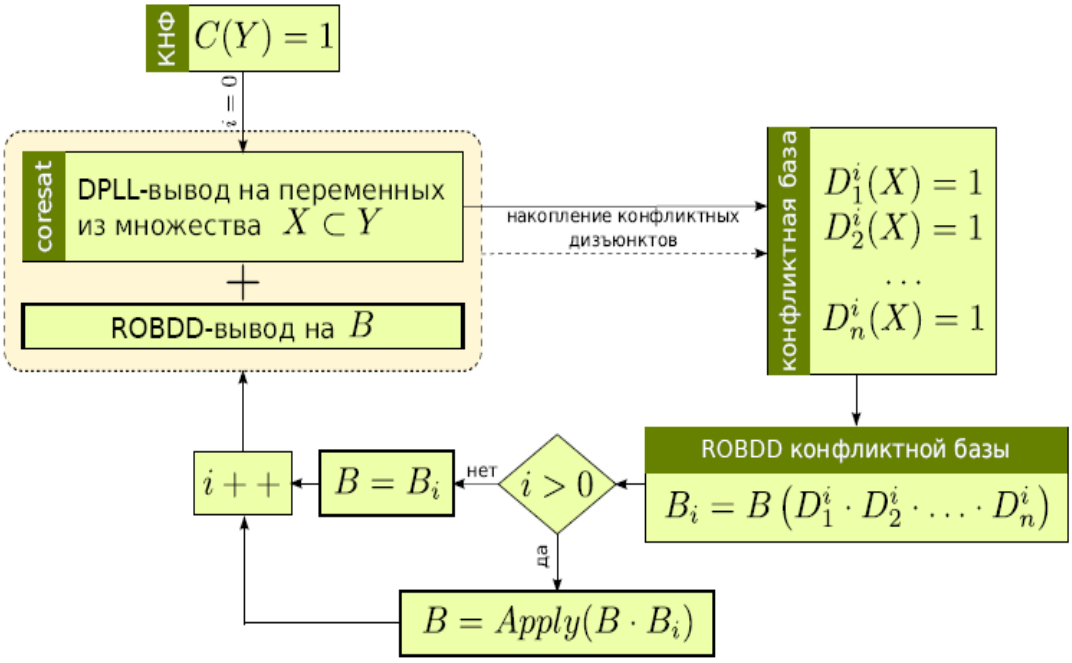

Рис. 1. Схема работы гибридного (SAT+ROBDD) решателя hsat

Параллельный гибридный (SAT+ROBDD) решатель получил название «mhsat». Далее описаны основные принципы его работы. На k вычислительных ядрах запускается k версий решателя hsat, получающих на входе, вообще говоря, произвольную КНФ C. Все версии hsat стартуют с различными начальными порядками угадывания переменных. Процесс вывода является итерационным. Каждая итерация разбивается на два этапа. На первом этапе все ядра работают независимо и на каждом происходит накопление конфликтных дизъюнктов, причем база конфликтных ограничений имеет вид ROBDD. На втором этапе все ядра обмениваются накопленными ограничениями (при этом возникает необходимость изменения

в некоторых ROBDD порядка означивания переменных). Обмен происходит в соответствии со схемой, представленной на рисунке 2. После обмена ограничениями решатель снова переходит в режим независимой работы ядер (следующая итерация). Работа продолжается до момента решения соответствующей SAT-задачи на некотором ядре.

Решатель mhsat был реализован с использованием стандарта MPI и протестирован на задаче обращения дискретной функции, задающей порождение ключевого потока в системе шифрования A5/1 (см., например [21]). Следует отметить, что задача поиска выполняющего набора КНФ $C(A5/1)$, непосредственно кодирующей криптоанализ A5/1, является очень сложной. Поэтому были рассмотрены ослабления этой задачи, а именно, рассматривались КНФ из декомпозиционного семейства $\Delta_d = \{C_1^{A5/1},\ldots,C_{2^d}^{A5/1}\}$, полученного в результате подстановок в $C(A5/1)$ всевозможных значений некоторых d булевых переменных, выбираемых специальным образом. Именно такой подход используется при крупноблочном распараллеливании SAT-задач [16], кодирующих задачи криптоанализа различных шифров (в том числе и A5/1).

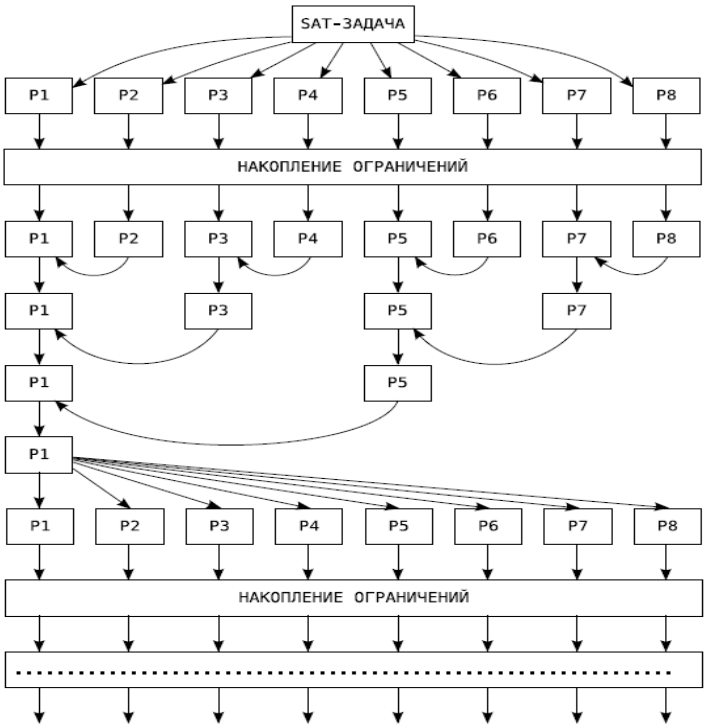

Рис. 2. Схема обмена ограничениями в решателе mhsat (рассмотрен случай 8 ядер: P1-P8)

		Решатель		
		mhsat	**dminisat**	**MiraXT**
Время	**Лучшее**	283	310	267
	Худшее	1762	9783	>4000
	Среднее	651,26	3336,46	>1875,7

Таблица 1. Результаты тестирования в секундах (кратко)

		Решатель	
		dminisat	**MiraXT**
Превосходство	**Лучшее**	15,45	11,05
	Худшее	1,63	0,61
	Среднее	5,12	>2,88

Таблица 2. Среднее превосходство mhsat перед другими решателями (в число раз)

В численных экспериментах были использованы КНФ из декомпозиционного семейства $\Delta_{20} = \{C_1^{A5/1}, \ldots, C_{2^{20}}^{A5/1}\}$ ($d = 20$). Решатель mhsat запускался на 4 ядрах процессора Intel® Xeon® E5345 с тактовой частотой 2,33 ГГц. Проводилось сравнение по эффективности mhsat, известного параллельного решателя MiraXT [11], а также решателя dminisat [16]. Было сгенерировано 50 тестов (КНФ, выбираемые случайным образом из семейства Δ_{20}). Как видно из таблицы 1, среднее время решения задачи решателем mhsat составило 651 секунду, решателем MiraXT – 1876 секунд, решателем dminisat – 3336 секунд. В среднем mhsat по эффективности на данном наборе тестов превзошел MiraXT в 2,88 раза, а dminisat – в 5,12 раза.

Заключение

В статье представлен новый подход к обращению полиномиально вычислимых дискретных функций. Такие функции важны своими многочисленными приложениями в математической и прикладной кибернетике. Основу предложенного подхода составляют алгоритмы логического вывода, использующие две структуры данных – булевы формулы в КНФ и двоичные диаграммы решений, а точнее, ROBDD. Логический вывод организуется как на КНФ, так и на ROBDD. Компактность ROBDD-представлений массивов конфликтных дизъюнктов позволяет эффективно обмениваться ими в распределенных вычислительных средах. Основным практическим результатом работы является новый решатель гибридной (SAT+ROBDD) архитектуры, функционирующий в MPI-среде. Данный решатель показал высокую эффективность на задачах обращения некоторых дискретных функций, используемых в криптографии.

СПИСОК ЛИТЕРАТУРЫ

[1] http://www.satlive.org
[2] www.isa.ewi.tudelft.nl/Jsat
[3] Davis M., Logemann G., Loveland D. A machine program for theorem proving // Communication of the ACM. 1962. vol. 5. P. 394-397.
[4] Marqeus-Silva J. P. Sakallah K.A. GRASP: A search algorithm for propositional satisfiability // IEEE Trans. on Computers. 1999. vol. 48. No. 5. P. 506-521.
[5] Moskewicz M, Madigan C., Zhao Y., Zhang L., Malik S. Chaff: Engineering an Efficient SAT Solver // Proc. Design Automation Conf. (DAC). 2001. P. 530-535.
[6] MiniSat [http://minisat.se/MiniSat.html]
[7] Семенов А.А., Заикин О.С. Неполные алгоритмы в крупноблочном параллелизме комбинаторных задач // Вычислительные методы и программирование. 2008. т.9. С. 108-118.
[8] Семенов А.А. Декомпозиционные представления логических уравнений в задачах обращения дискретных функций // Известия РАН. Теория и системы управления. 2009. №5. С. 47-61.

[9] Игнатьев А.С., Семенов А.А. Алгоритмы работы с ROBDD как с базами булевых ограничений // Прикладная дискретная математика. 2010. №1. С. 86-104.
[10] Schubert T., Lewis M., Becker B. PaMiraXT: Parallel SAT Solving with Threads and Message Passing //Journal on Satisfiability, Boolean Modeling and Computation. 2009. Vol.6. P. 203-222.
[11] Cook S.A. The complexity of theorem-proving procedures // Proc. 3rd Ann. ACM Symp. on Theory of Computing, ACM, 1971, pp. 151-159.
[12] Zhang L., Madigan C., Moskewicz M, Malik S. Efficient conflict driven learning in a boolean satisfiability solver // Proc. Intern. Conf. on Computer Aided Design (ICCAD). – 2001. pp. 279-285.
[13] Lynce I., Marques-Silva J.P. Efficient data structures for backtrack search SAT solvers // Annals of Mathematics and Artificial Intelligence.– 2005. vol. 43. pp. 137-152.
[14] Семенов А.А., Заикин О.С., Беспалов Д.В., Ушаков А.А. SAT-подход в криптоанализе некоторых систем поточного шифрования // Вычислительные технологии. – 2008. т. 13. №6. С. 134-150.
[15] Заикин О.С., Семенов А.А. Технология крупноблочного параллелизма в SAT-задачах // Проблемы управления.– 2008. №1. С. 43-50.
[16] Семенов А.А., Заикин О.С., Беспалов Д.В., Буров П.С., Хмельнов А.Е. Решение задач обращения дискретных функций на многопроцессорных вычислительных системах // Труды Четвертой Международной конференции PACO'2008 (Москва 26-29 октября 2008). С. 152-176.
[17] Lee C.Y. Representation of Switching Circuits by Binary-Decision Programs // Bell Systems Technical Journal. 1959. vol. 38. pp. 985-999.
[18] Bryant R.E. Graph-Based Algorithms for Boolean Function Manipulation // IEEE Transactions on Computers. 1986. vol. 35. No. 8. pp. 677-691.
[19] Gopalakrishnan S., Durairaj V., Kalla P. Integrating CNF and BDD Based SAT Solvers // In Proceedings of the Eighth IEEE International Workshop on High-Level Design Validation and Test Workshop. 2003. p. 51.
[20] SAT-Race 2008 [http://baldur.iti.uka.de/sat-race-2008/index.html]
[21] A. Biryukov, A. Shamir, D. Wagner Real Time Cryptanalysis of A5/1 on a PC // Fast Software Encryption Workshop. 2000. pp. 1-18.