Символьные и квантовые вычисления в решении задачи выполнимости булевых формул

Гердт Владимир Петрович

Лаборатория Информационных Технологий
Объединённый Институт Ядерных Исследований, Дубна

МСКФ-2013

Булевы функции и формулы

Булева (логическая) переменная $x \in \{0,1\}$ может принимать два значения 1 (“истинно”) или 0 (“ложно”).

Булева функция $f(x_1,\ldots,x_n)$ ($f : \{0,1\}^n \mapsto \{0,1\}$) отображает множество из 2^n всевозможных значений своих переменных в $\{0,1\}$, т.е. может принимать те же логические значения 0 и 1, что и каждая из переменных.

Булева формула задает явную зависимость булевой функции от ее переменных посредством логических операций.

Основные логические операции

1. **отрицание** $\bar{x}$: $\bar{0} = 1,\ \bar{1} = 0$
2. **конъюнкция** $x \wedge y$: $1 \wedge 1 = 1,\ 0 \wedge 0 = 1 \wedge 0 = 0 \wedge 1 = 0$
3. **дизъюнкция** $x \vee y$: $0 \vee 0 = 0,\ 0 \vee 0 = 1 \vee 0 = 0 \vee 1 = 0$

Через эти операции выражаются другие логические операции

- **импликация** $x \Longrightarrow y$: $\bar{x} \vee y$
- **эквивалентность** $x \Longleftrightarrow y$: $(\bar{x} \vee y) \wedge (y \vee \bar{x})$
- **исключающее или** (сложение по модулю 2) $x \oplus y$: $(x \wedge \bar{y}) \vee \bar{x} \wedge y)$

Булева переменная или ее отрицание называется литералом.

Задача выполнимости булевых формул/функций

Данная задача, сокращенно ВЫП или SAT(satisfiability) заключается в следующем: для заданной формулы, выраженной через основные логические операции, можно ли добиться ее истинности (т.е. получить значение 1 для соответствующей булевой функции) предписывая определенные значения (0 или 1) входящим в нее переменным.

Подзадача: $k-$выполнимость КНФ ($k-$SAT)

Выполнимость булевой формулы в конъюктивной нормальной форме (КНФ) вида

$$(y_{i_1} \vee \cdots \vee y_{i_k}) \wedge (y_{j_1} \vee \cdots \vee y_{j_k}) \wedge \cdots \wedge (y_{t_1} \vee \cdots \vee y_{t_k})$$

где каждая скобка содержит дизъюнкции ровно k литералов.

Оптимизационная версия: максимальная $k-$выполнимость (MAX $k-$SAT)

Найти значения переменных, для которых в $k-$КНФ форме ($k \geq 2$) выполняется максимальное число скобок.

Простой пример задачи ВЫП/SAT

Рассмотрим две булевы функции F_1 и F_2 от трёх переменных a, b, c

$$F_1 = (\bar{a} \vee b) \wedge (\bar{b} \vee c), \quad F_2 = F_1 \wedge a \wedge \bar{c}$$

Функция F_1 - выполнима, а F_2 - нет. Почему?

Пусть булевы переменные (литералы) обозначают следующие мои убеждения:

a : я люблю пиво
b : я должен съездить в Германию
c : я должен выпить немецкого пива

Дизъюнкции в F_1 можно интерпретировать как импликации:

- $a \Longrightarrow b$: я люблю пиво, поэтому должен поехать в Германию
- $b \Longrightarrow c$: если поеду в Германию, то должен выпить там немецкого пива

А что будет, если я люблю пиво но не пью немецкого пива? В этом случае должна должна выполняться функция $(a \wedge \bar{c})$, что противоречит моим убеждениям, т.к.

$$\overline{a \wedge \bar{c}} = (\bar{a} \vee c) \equiv (a \Longrightarrow c)$$

Теоретическая значимость задачи ВЫП/SAT

- $k-$SAT
  1. $k = 1$ легко решается за время $O(n)$ (n-число переменных).
  2. $k = 2$ полиномиально разрешима и существует алгоритм решаемый эту задачу за время $O(n)$.
  3. $k \geq 3$ *NP−полная задача* (Cook'71,Levin'73). Один из наиболее эффективных алгоритмов решения этой задачи - вероятностный алгоритм PPSZ (Paturi,Pudlák,Saks,Zane'98) - имеет временную сложность

$$O(2^{n-\mu_k/k}), \quad \lim_{k\to\infty} \mu_k = \pi^2/6$$

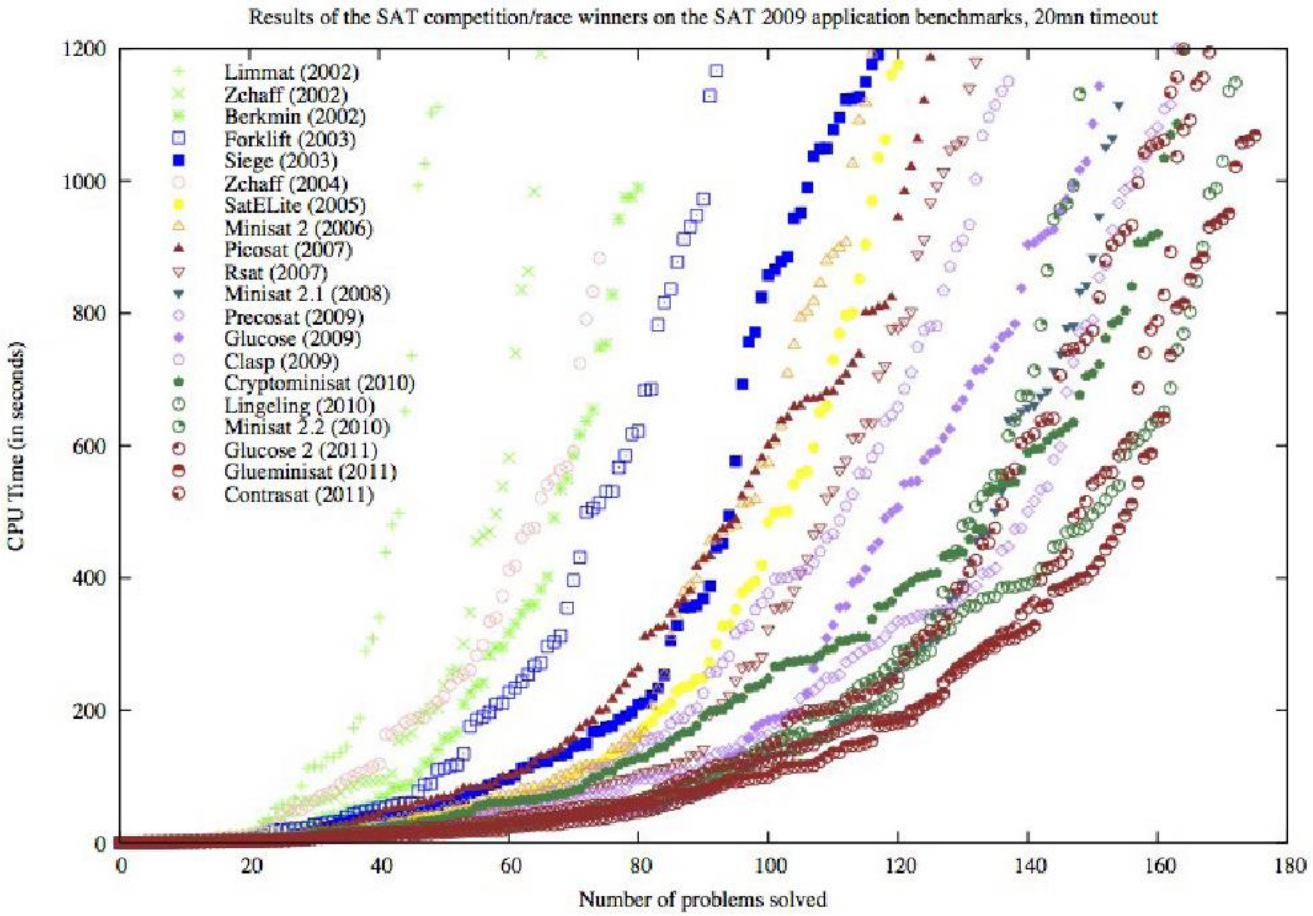
     а лучшие на сегодняшний день оценки алгоритмической сложности для $k = 3$ и $k = 4$ равны 1.308^n и 1.469^n, соответственно (Hertli, 2011).

- MAX $k-$SAT

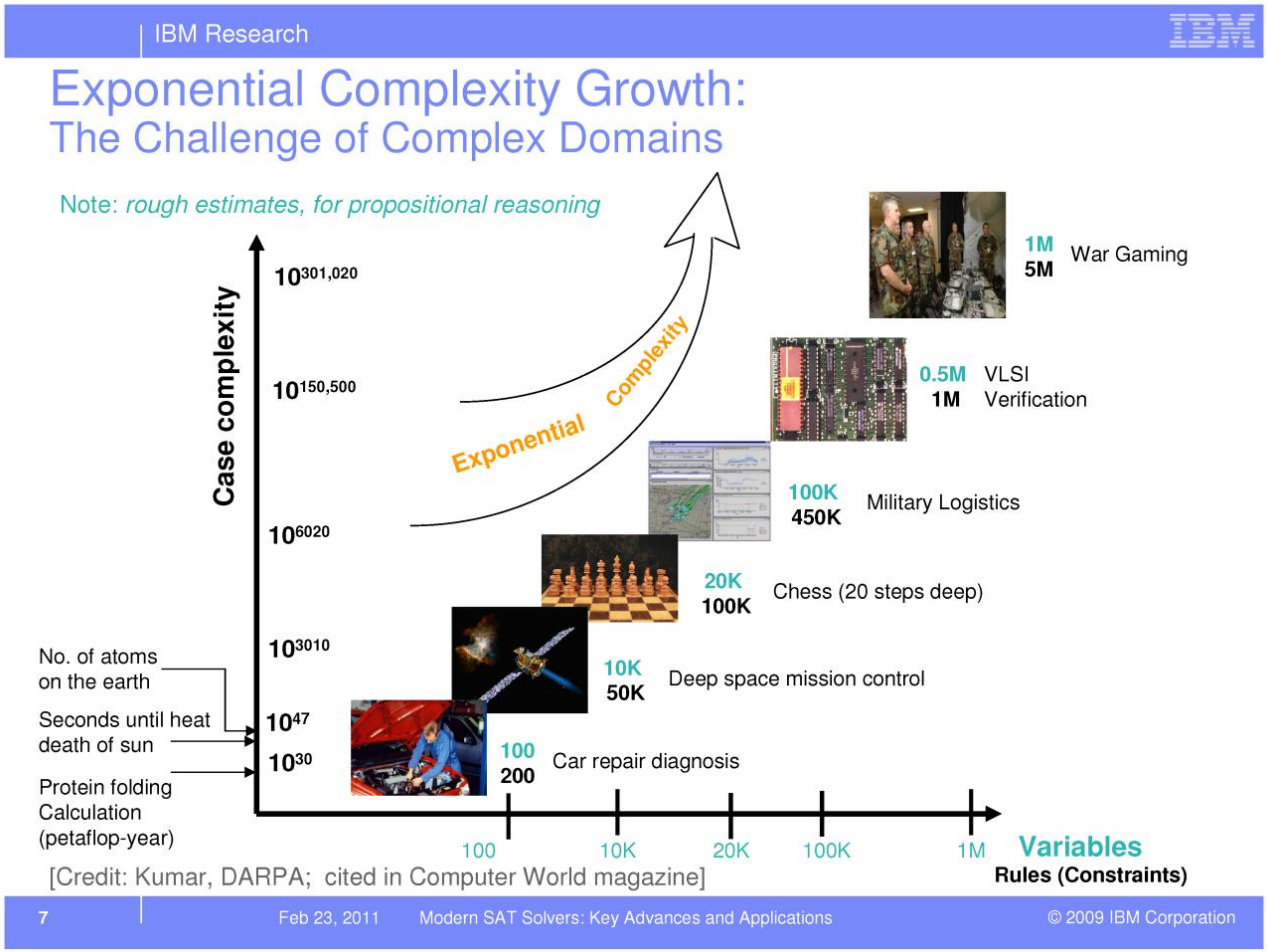
  NP−полная задача. Один из наиболее эффективных алгоритмов ее решения имеет временную сложность $O(b2^n)$ по n (Zhang, Shen, Manyá, 2003), где b-максимальное число появлений булевой переменной во входной формуле. Если длина формулы - L, то типично $b \simeq L/n$.

Прогресс в эффективности решателей SAT

Источник: http://www.satcompetition.org/PoS12/

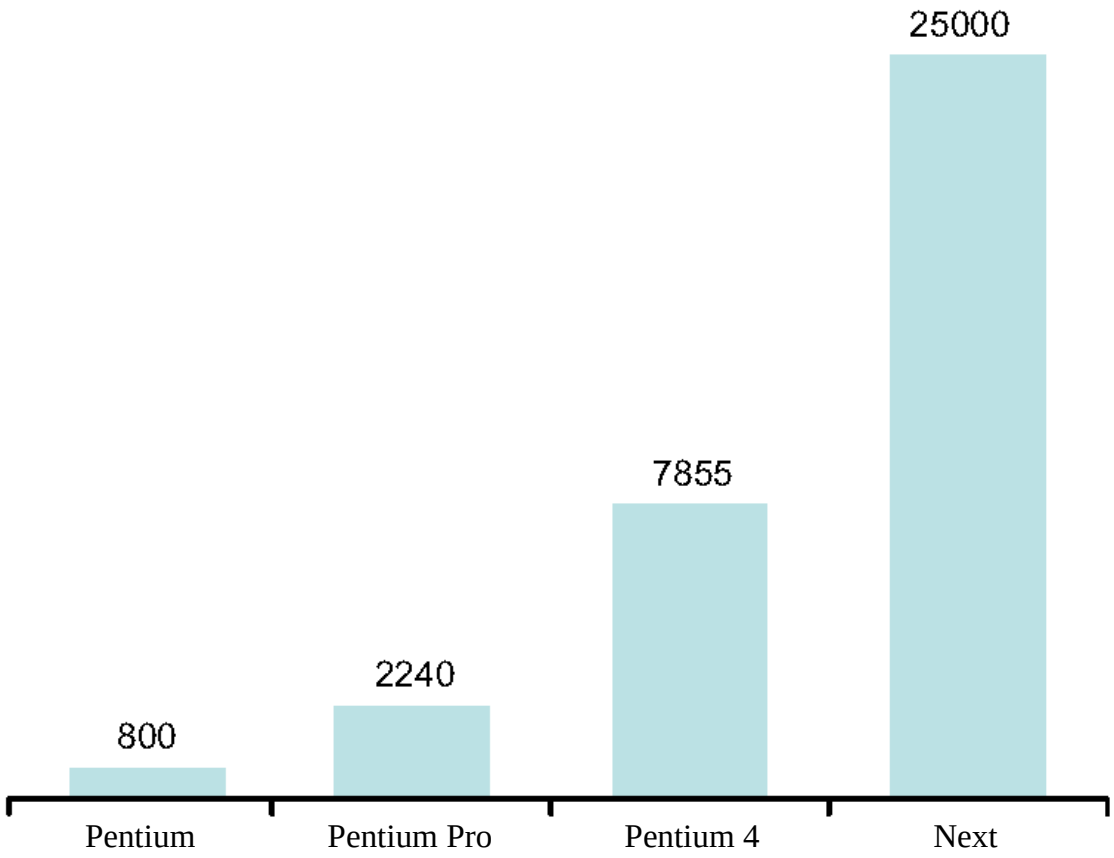


Практическая важность задачи ВЫП/SAT



Динамика роста числа логических ошибок

Источник: Proc. of Design Automation and Test in Europe, pp.1240-1245, 2006



Число логических ошибок на этапе проектирования процессоров, предшествующего реализации в кремнии (pre-silicon) по данным корпорации Intel

Верификация процессоров на этапе pre-silicon

По данным корпорации ИНТЕЛ (LNCS 5643, pp. 414-429, 2009) разработка процессора на основе новой микроархитектуры типично требует работы более 500 инженеров в течении 2-3 лет. Основная часть работ (до 70%) связана с функциональной верификацией.

Пример: Intel® Core™ i7 процессор.

В EXE кластере процессора Intel® Core™ i7 реализовано 2700 различных микроинструкций и поддержка многопотоковости (SMT). А число комбинаций данных для одной инструкции в арифметике с плавающей запятой с расширенной точностью (80 бит) составляет 2^{160}.

Все работы по верификации процессора Intel® Core™ i7 были выполнены на основе SAT технологии.

Статистика по процессору Intel XScale с использованием SAT решателя BerkMin

Источник: Proc. of MEMOCODE'03, June 2003, pp.65-74

Processor	Boolean Variables			CNF Variables	CNF Clauses	Formal Verification Time [sec]			
	e_{ij}	Control	Total			TLSim	EVC	BerkMin	Total
XS-7	450	102	552	4,347	39,459	0.11	1	3	4
XS-7-8	566	151	717	7,216	73,179	0.15	3	9	12
XS-7-8-SB	565	155	720	7,253	73,306	0.22	3	11	14
XS-7-8-SB-PC1	1,666	210	1,876	16,386	214,163	0.18	7	21	28
XS-7-8-SB-PC2	1,667	216	1,883	19,215	244,916	0.26	8	32	40
XS-7-8-SB-PC1-MC	2,082	280	2,362	23,369	396,711	0.34	17	92	109
XS-7-8-SB-PC2-MC	2,078	291	2,369	26,651	435,140	0.38	17	127	144
XS-7-8-SB-PC1-MC-IMP	2,386	272	2,658	36,266	573,252	0.46	23	162	185
XS-7-8-SB-PC2-MC-IMP	2,384	281	2,665	40,369	616,368	0.47	22	244	266

Символьный подход

Алгоритмы, используемые в решателях SAT, являются переборными, например, алгоритм DPLL (Davis,Putnam'60; Logemann,Loveland'62) и его "потомки", и заключаются в оптимизации процедуры перебора с учетом полученной в его ходе информации.

Особенности символьного подхода

Переменные булевой функции рассматриваются как символы $x_1, \ldots, x_n$, и задача символьных алгоритмов состоит в преобразовании функции к такому выражению, эквивалентному исходному, из которого можно сделать вывод о его (не)выполнимости.

Одним из таких выражений является многочлен Жегалкина (Жегалкин, 1927), который имеет вид

$$a \oplus a_1 x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_n x_n \oplus a_{12} x_1 x_2 \oplus \cdots \oplus a_{1\ldots n} x_1 \cdots x_n, \quad a, \cdots, a_{1\ldots n} \in \{0, 1\}$$

Переход от КНФ к многочлену Жегалкина

Многочлен Жегалкина получается из КНФ следующим образом.

1. Внутри скобок с дизъюнкциями применить подстановки вида:

$$\bar{x} \longrightarrow x \oplus 1, \quad y \vee z \longrightarrow y \oplus z \oplus y \cdot z$$

2. К полученному для каждой скобки многочлену прибавить 1 ($\oplus$).
3. После перемножения всех полученных выражений и приведения подобных членов прибавить к результату 1 ($\oplus$).

Замечание (Lindqvist, arXiv:1211.3398 [math.AC])

Это представление является каноническим (единственным). Функция $f(x_1, \ldots, x_n)$ выполнима тогда и только тогда, когда ее многочлен Жегалкина равен 1. Более того, если этот многочлен отличен от 1, по нему можно алгоритмически найти все его корни в $\{0,1\}^n$, т.е. определить все значения переменных, обеспечивающих выполнение функции.

Оптимизация вычислений

При большом числе переменных (на практике оно достигает 1 миллиона), преобразование всей булевой функции (КНФ) в многочлен Жегалкина оказывается вычислительно неподъемным.

Другая каноническая форма

Можно преобразовать в многочлен каждую скобку в КНФ и затем построить каноническую форму (также отличную от $\{1\}$ тогда и только когда функция выполнима) полученной системы многочленов (называемую в литературе базисом Грёбнера), используя для этого специальный (инволютивный) алгоритм (Гердт,Блинков,Зинин'2010). При этом надо позаботиться о максимально компактным по памяти внутреннем представлении промежуточных многочленов Жегалгина.

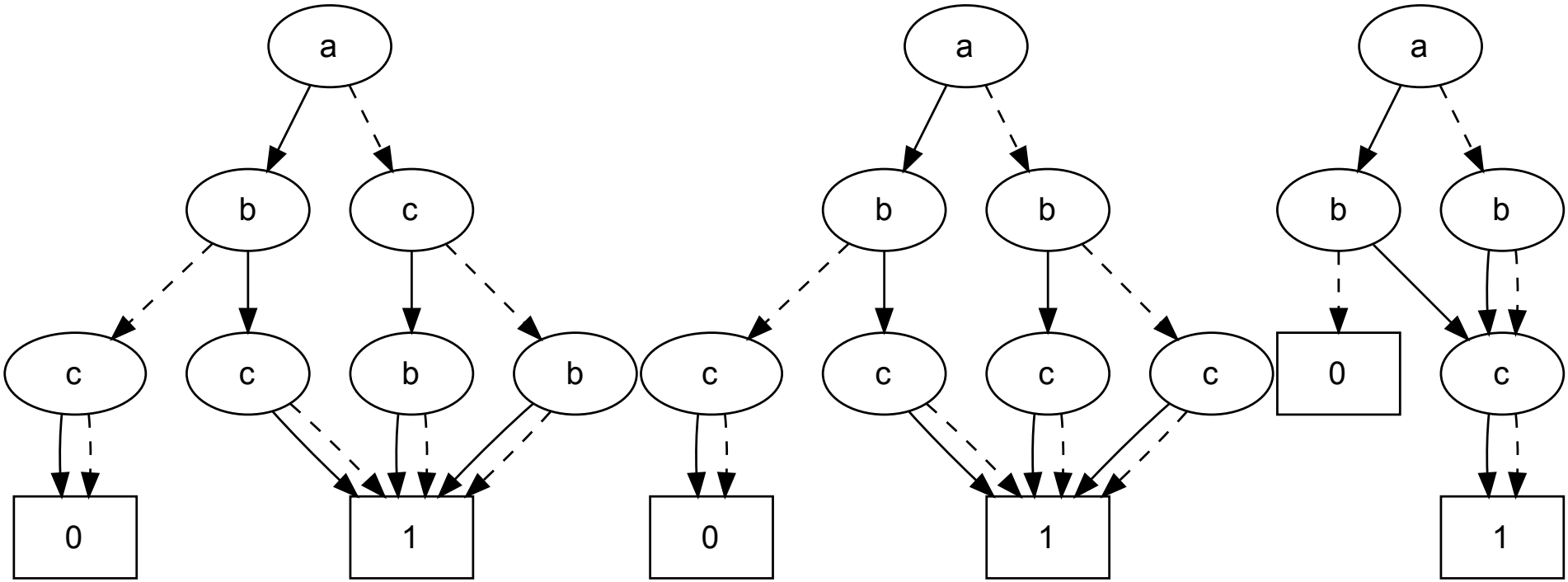
Рассмотрим опять наш пример: $F_1 = (\bar{a} \vee b) \wedge (\bar{b} \vee c), \quad F_2 = F_1 \wedge a \wedge \bar{c}.$

1. $F_1 \longrightarrow a \oplus b \oplus ab \oplus bc, \quad F_2 \longrightarrow 1$
2. $F_1 \longrightarrow \{a \oplus ab, a \oplus ac, b \oplus bc\}, \quad F_2 \longrightarrow \{1\}$

Первая версия нашего инволютивного алгоритма для булевых многочленов реализована в системах символьных вычислений Reduce и Macaulay 2.

Диаграммное представление многочленов Жегалгина

Пример: $p = abc + ab + bc + b + c + 1$



BDD OBDD ZDD

SAT и квантовые вычисления

Для решения задачи ВЫП/SAT на (будущих?) квантовых компьютерах уже разработан ряд квантовых алгоритмов.

Алгоритмы переборного типа

Стандартный подход состоит в использовании известного квантового алгоритма Гровера для поиска нужной элемента в неотсортированном массиве значений булевых переменных. Этот квантовый алгоритм ускоряет поиск записи в базе данных с N записями с $O(N)$ до $O(N^{1/2})$

Это, соответственно, улучшает оценку сложности алгоритма. Например, в случае алгоритма PPSZ для решения задачи $k-$SAT получается улучшение

$$O(2^{n-\mu_k/k}) \longrightarrow O(2^{\frac{n-\mu_k/k}{2}}), \quad \lim_{k\to\infty} \mu_k = \pi^2/6$$

Замечание

Из-за отсутствия квантовых компьютеров, подходящих для реализации подобных алгоритмов, такой результат пока носит чисто теоретический характер.

MAX 2-SAT на квантовом процессоре DW1

Первые экспериментальные результаты по решению оптимизационной задачи MAX 2-SAT на специализированном 128-кубитном квантовом процессоре DW1 (“Rainier”) известной канадской фирмы D-Wave, в сравнении с классическим решателем AKMAXSAT, представлены мировой общественности (arXiv:1307.393 [quant-ph]) около 3-х месяцев назад.

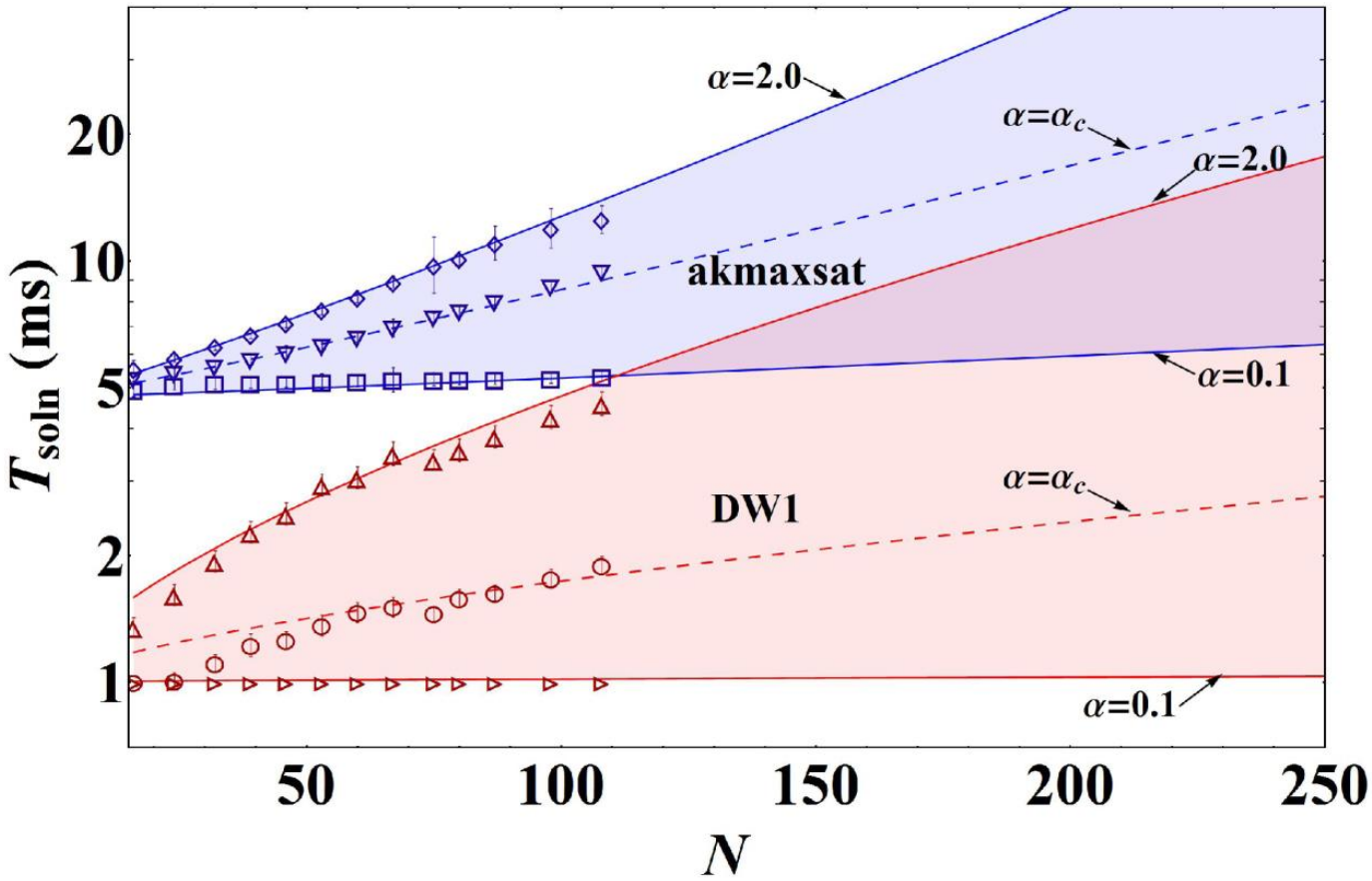
Этот процессор предоставляет пользователю до 108 кубит на проведение вычислений и ориентирован на решение оптимизационных задач, сводящихся к нахождению основного состояния (с минимальной энергией) в модели Изинга. Метод поиска минимума энергии осуществляется в рамках адиабатического квантового компьютинга и с использованием реализации на DW1 алгоритма квантового отжига.

Замечание

Для задачи MAX 2-SAT удается построить гамильтониан в форме модели Изинга, в которой сверхпроводящие кубиты процессора DW1 играют роль “спинов”, а значения этих спинов в основном состоянии системы соответствуют решению задачи MAX 2-SAT.

Сравнение DW1 с **akmaxsat** на задаче MAX 2-SAT

$\alpha = M/N$, $N-$число переменных, $M-$число скобок в КНФ



Заключение

- Развитие методов, алгоритмов и программ проверки выполнимости булевых функция является важной и актуальной задачей современной прикладной математики и информатики.
- Это обусловлено, в частности, необходимостью верификации логических элементов в современной копьютерной индустрии и большой трудоемкостью такой верификации.
- Несмотря на $NP-$полноту задачи и экспоненциальные оценки, на практике часто удается решать ее для тысяч и более переменных. Что это значит? $P = NP$?
- Символьные и квантовые вычисления являются новыми перспективными направлениями в методологии решения задачи.