Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра системного программирования

Программная инженерия

Суханова Анжела Кирилловна

Реализация элиминации кванторов для арифметики Пресбургера, обогащённой функцией возведения двойки в степень

Производственная практика

Научный руководитель:
ассистент кафедры ИАС Смирнов К. К.

Санкт-Петербург
2020

Оглавление

Введение

Современный этап развития индустрии программных систем характеризуется значительным усложнением процесса их разработки, что приводит к увеличению числа ошибок, возникающих при проектировании программного обеспечения. В таких условиях крайне важными оказываются проверка и доказательство корректности разрабатываемой программы, ведь они необходимы для контроля соответствия поведения программы ожидаемому и обеспечения её безопасности. Существуют разные методы, направленные на разработку качественного программного обеспечения, отвечающего поставленным требованиям: одним из них является формальная верификация.

Формальная верификация — это доказательство соответствия или несоответствия программы её формальному описанию или, что более распространено на практике, проверка формализованных условий, описывающих ожидаемое или недопустимое поведение программы (часто у разработчиков нет формального описания программы, но есть представление о "запрещённых" состояниях, в которые она не должна попадать). В основу этой проверки нередко ложится решение задачи выполнимости формул в теориях[1] (SMT[2]), так как условия программы и ограничения, накладываемые на неё требованиями, могут быть сведены к определению выполнимости логических формул. Существуют SMT-решатели (SMT-солверы), автоматически определяющие выполнимость формулы в теориях. SMT-решатели используются инструментами формальной верификации, такими как Frama-C, BLAST, Java Pathfinder и другие.

Обычно легче определить, выполнима ли формула без кванторов, поэтому полезно уметь преобразовывать формулы, содержащие кванторы, в семантически эквивалентные бескванторные. Процесс такого преобразования называется элиминацией кванторов (quantifier elimination,

[1]Теория — множество предложений (формул без свободных переменных).

[2]SMT — задача выполнимости формул в логике первого порядка, где функциональные и предикатные символы интерпретируются согласно конкретным теориям. В дальнейшем будет использоваться эта аббревиатура.

QE).

Так как в основе архитектуры компьютера лежат операции с битовыми векторами, то необходимо уметь решать SMT в теории битовых векторов, что вызывает интерес к упрощающим это решение алгоритмам исключения кванторов из формул над двоичными векторами. Операции над последними можно свести к вычислениям в арифметике Пресбургера[3], обогащённой функцией 2^x. Доказательство того, что эта теория допускает элиминацию кванторов, а также алгоритм элиминации, сопутствующий построению доказательства, впервые были представлены А. Л. Семёновым [6], а затем более подробно описаны в других работах [4, 5]. Однако реализации данного алгоритма пока нет, чему и посвящена эта работа.

В настоящее время существует несколько поддерживаемых, конкурентоспособных SMT-решателей, работающих с двоичными векторами: Boolector, Z3, CVC4 и другие. В рамках этой курсовой будет осуществляться работа с Boolector, так как он специализируется на теории битовых векторов, а также в течение многих лет побеждал в The SMT Competition[4] (за исключением 2020-го года, в котором актуальная версия Boolector-а не принимала участие в соревновании).

[3]Арифметика Пресбургера — это теория первого порядка, описывающая натуральные числа со сложением и названная в честь предложившего её Мойжеша Пресбургера.

[4]The SMT Competition или SMT-COMP — ежегодное соревнование между SMT-солверами: https://smt-comp.github.io/2020/index.html

1. Постановка задачи

Целью данной работы является реализация элиминации кванторов для арифметики Пресбургера, расширенной функцией возведения двойки в степень[5]. Для её достижения были поставлены следующие задачи.

- Изучение алгоритма элиминации кванторов для расширенной арифметики Пресбургера.
- Реализация изученного алгоритма.
- Внедрение реализаций в SMT-решатель Boolector.

[5]Далее она иногда будет упоминаться как "расширенная арифметика Пресбургера", но имеется в виду, что теория расширяется функцией 2^x.

2. Теоретический обзор и описание реализации

2.1. Сигнатура

Говоря далее о расширенной арифметике Пресбургера, мы будем иметь в виду арифметику со следующей сигнатурой: $< +, 2^x, \leq, 0, 1, >$. Также воспользуемся некоторыми обозначениями, введёнными в [5].

- Записи вида $1 + 1 + \ldots + 1$ обозначим соответствующим натуральным числом.
- $x < y \leftrightarrow x + 1 \leq y$ и $x \geq (>) y \leftrightarrow y \leq (<) x$.
- $x - y = 0$, если $x < y$, и $x - y = x - y$, если $y \leq x$.
- $n \cdot x$, где $n \in \mathbb{N}^*$ — обозначение $x + x + \ldots + x$ (n раз).
- $t_1 + z \cdot x \leq (=, \geq) t_2$, где $z \in \mathbb{Z}<$ означает, что $t_1 \leq (=, \geq) t_2 + (-z) \cdot x$ при $z < 0$.
- $\frac{x}{n} = y \leftrightarrow x = n \cdot y + z$, где $0 \leq z < n$.
- $l_2(x) = y \leftrightarrow 2^y \leq x < 2^{y+1}$.
- $P_2(x) \leftrightarrow x = 2^{l_2(x)}$ (то есть x — степень двойки).
- $\lambda(x) = y \leftrightarrow ((y \leq x < 2 \cdot y) \wedge P_2(y))$.

2.2. Алгоритм элиминации кванторов для расширенной арифметики Пресбургера

Заметим, что формулу, содержащую квантор всеобщности, $\forall x\ F$ можно заменить эквивалентной ей формулой $\neg\exists x\ \neg F$, а все кванторы существования можно элиминировать последовательно от самого внутреннего к внешнему. Также любая формула первого порядка может быть представлена в конъюктивной нормальной форме. Таким образом, алгоритм элиминации сводится к устранению квантора существования из формулы вида $\exists x\ \theta(x, \overline{y})$, где $\overline{y}$ — свободные переменные, а сама функция $\theta(x, \overline{y})$ — конъюнкт литералов. Это устранение квантора будет построено по материалу, изложенному в [5] (именование согласовано с этой работой).

Первым шагом в элиминации кванторов для рассматриваемой арифметики является преобразование формулы под квантором в дизъюнкт конъюнктов неравенств между термами следующего вида:

- $\sum\limits_{i=0}^{n} a_i \cdot 2^{d \cdot x_i} + \sum\limits_{i=0}^{n} b_i \cdot x_i + c$, где a_i, b_i, $c \in \mathbb{Z}$, $d \in \mathbb{N}$, а x_i — связанные переменные, причём x_0 — обозначение x (будем называть их S-термами);

- термы сигнатуры арифметики Пресбургера, обогащённой функцией возведения двойки в степень, без связанных переменных (будем называть их L-термами).

Это преобразование осуществляется посредством введения новых переменных, связанных кванторами существования — отсюда и возникают x_i (подробнее об этом шаге можно прочитать в [5]).

Существует два варианта вида преобразованной формулы.

- x появляется во всех неравенствах линейно. Тогда можно считать, что формула под кванторами выглядит следующим образом:
  $\bigwedge\limits_{1 \le i \le p, 1 \le j \le q, 1 \le k \le s} f_j(\overline{x}) + g_j(\overline{y}) \le d_k \cdot x \le f_i(\overline{x}) + g_i(\overline{y})$,
  где $d_k \in \mathbb{Z}$ и зависит от i и j, $\overline{y}$ — свободные переменные, а $\overline{x} = (x_1, x_2, \ldots, x_n)$, то есть $g_j(\overline{y})$, $g_i(\overline{y})$ — L-термы, а $f_j(\overline{x})$, $f_i(\overline{x})$ — S-термы.

- Хотя бы в одно неравенство x входит в экспоненциальном терме, то есть существует неравенство, имеющее вид:
  $a_0 \cdot 2^{d \cdot x_0} + \sum\limits_{i=1}^{n} a_i \cdot 2^{d \cdot x_i} + \sum\limits_{j=0}^{n} b_j \cdot x_j + c \le t(\overline{y})$, $t(\overline{y})$ — L-терм, a_i, b_i, $c \in \mathbb{Z}$, $a_0 \ne 0$, $d \in \mathbb{N}^*$.

Таким образом, реализация подразумевает разбор двух случаев: линейного и экспоненциального вхождения x.

2.3. Реализация случая линейного вхождения связанной переменной

Работа над этим случаем была начата с упрощённого варианта, а именно с элиминации кванторов для формул с единственной связанной переменной: $\exists x \bigwedge\limits_{1\le i\le p, 1\le j\le q, 1\le k\le s} g_j(\overline{y}) \le d_k \cdot x \le g_i(\overline{y})$.

При $d_k = 1\ \forall k$ искомая формула без квантора — $\bigvee\limits_{\rho\in S_p}\bigvee\limits_{\tau\in S_q} g_{\tau(q)}(\overline{y}) \le \ldots \le g_{\tau(1)}(\overline{y}) \le g_{\rho(1)}(\overline{y}) \le \ldots \le g_{\rho(p)}(\overline{y})$, где S_p — группа перестановок элементов множества $\{1, \ldots, p\}$, а S_q — группа перестановок элементов множества $\{1, \ldots, q\}$.

Если существует $d_k \neq 1$, то ситуация усложняется. По алгоритму все неравенства домножаются так, чтобы x встречался в них с одинаковым коэффициентом d (d — НОК всех d_k). Однако в отличие от арифметики Пресбургера, позволяющей работать с натуральными числами без ограничения сверху, арифметика битовых векторов работает с векторами фиксированного размера, то есть накладывает ограничение на значения переменных и величину констант, а также поддерживает переполнение. Из-за этого результирующая формула, полученная по алгоритму, может быть не эквивалентна исходной. Решение этого затруднения пока не найдено.

2.4. Формат ввода формул

Изначально в качестве формата ввода рассматривались стандарты Btor [2] и Btor2 [3]. Оба формата не поддерживают работу с кванторами, и хотя это неудобство может быть преодолено (например, договорённостью, что первые введённые переменные связаны кванторами существования), было принято решение работать со стандартом SMT-LIB v.2 [1], в котором можно записывать формулы с кванторами. Парсеры всех вышеупомянутых стандартов записи встроены в Boolector.

Работая с Boolector-ом, можно задать любую формулу над битовыми векторами (даже с кванторами) програмно, используя функции решателя, но такой ввод не очень удобен. К тому же важно, чтобы программа могла работать с общепринятым, распространённым форматом, ведь в нём записывались и пишутся условия и ограничения, накладываемые на реальное программное обеспечение. Идея написать собственный парсер выражений над битовыми векторами также была отброшена из-за неоправданной трудоёмкости.

3. Текущие результаты

- Изучен описанный в [5] алгоритм элиминации кванторов для расширенной арифметики Пресбургера.
- Изучен исходный код Boolector-a.
- Рассмотрены различные форматы ввода формул над битовыми векторами (Btor, Btor2, SMT-LIB, SMT-LIB v.2), и, как следствие, выбран один из них, а именно стандарт SMT-LIB v.2.
- Реализована элиминация кванторов для формул вида $\exists x \bigwedge_{1 \le i \le p, 1 \le j \le q} g_j(\overline{y}) \le x \le g_i(\overline{y})$, где $g_j(\overline{y})$, $g_i(\overline{y})$ — линейные комбинации свободных переменных $\overline{y}$ и констант.
- Начата реализация алгоритма для случая, когда связанная переменная встречается в экспоненциальном терме.

Список литературы

[1] The SMT-LIB Standard: Version 2.0 : Rep. / Department of Computer Science, The University of Iowa ; Executor: Clark Barrett, Aaron Stump, Cesare Tinelli : 2010. — Available at www.SMT-LIB.org.

[2] Brummayer Robert, Biere Armin, Lonsing Florian. BTOR: Bit-Precise Modelling of Word-Level Problems for Model Checking // Proceedings of the Joint Workshops of the 6th International Workshop on Satisfiability Modulo Theories and 1st International Workshop on Bit-Precise Reasoning. — SMT '08/BPR '08. — New York, NY, USA : Association for Computing Machinery, 2008. — P. 33–38. — Access mode: https://doi.org/10.1145/1512464.1512472.

[3] Btor2 , BtorMC and Boolector 3.0 / Aina Niemetz, Mathias Preiner, Clifford Wolf, Armin Biere // Computer Aided Verification / Ed. by Hana Chockler, Georg Weissenbacher. — Cham : Springer International Publishing, 2018. — P. 587–595.

[4] Cherlin Gregory, Point Françoise. On extensions of Presburger arithmetic // Proc. 4th Easter Model Theory conference, Gross Köris. — 1986. — P. 17–34.

[5] Point Françoise. On the expansion $(\mathrm{N},+,2^x)$ of Presburger arithmetic. — 2007. — 01.

[6] Semenov A. L. Logical theories of one-place functions on the set of natural numbers // Math. USSR, Izv. — 1984. — Vol. 22. — P. 587–618.