



## Реализация элиминации кванторов для арифметики Пресбургера, обогащённой функцией $2^x$

**Автор:** Суханова Анжела Кирилловна, 371 группа (18.Б11-мм)

**Научный руководитель:** ассистент кафедры ИАС К. К. Смирнов

Санкт-Петербургский государственный университет  
Кафедра системного программирования

12 июня 2021г.

- Проверка и доказательство корректности разрабатываемой программы очень важны, ведь они необходимы для контроля соответствия поведения программы ожидаемому и обеспечения её безопасности
- Одним из методов формальной верификации является решение задачи выполнимости формул в теориях (SMT)
- Так как в основе архитектуры компьютера лежат операции с битовыми векторами, то необходимо уметь решать SMT в теории битовых векторов

- Можно ввести пропозициональные переменные для всех битов исходных термов и решать задачу выполнимости булевых формул
- Это работает, но такой подход очень трудоёмкий

- Легче определить, выполнимы ли формулы без кванторов
- Элиминация кванторов<sup>1</sup> — это процесс преобразования формулы, содержащей кванторы, в эквивалентную бескванторную формулу
- Пример  
Пусть есть формула  $\exists x : 3 \leq x \leq z$ . Такой  $x$  найдётся, если  $3 \leq z$  (например,  $x = 3$ )

---

<sup>1</sup>теория  $T$  допускает элиминацию кванторов, если для любой формулы этой теории  $\phi$  существует формула  $\psi$  без кванторов, такая что  $T \models \forall y. \phi(y) \leftrightarrow \psi(y)$

## Обогащенная арифметика Пресбургера ( $PA + 2^x$ )

- Операции над битовыми векторами можно свести к вычислениям в арифметике Пресбургера, обогащённой функцией  $2^x$ :  $\langle 0, 1, +, 2^x, \leq \rangle$
- Доказательство того, что эта теория допускает элиминацию кванторов, и идея элиминации впервые были представлены А. Л. Семёновым (1979). Алгоритм элиминации для  $PA + 2^x$  подробно описала Франсуаза Пуан (2007)
- Арифметику битовых векторов с кванторами поддерживают несколько SMT-решателей. Они не используют элиминацию кванторов для  $PA + 2^x$

**Целью** данной работы является реализация элиминации кванторов для арифметики битовых векторов на основе элиминации кванторов в обогащённой арифметике Пресбургера

**Задачи:**

- Выбор SMT-решателя
- Изучение алгоритма элиминации кванторов для расширенной арифметики Пресбургера и реализация его в рамках арифметики битовых векторов и выбранного SMT-решателя
- Экспериментальное исследование элиминации: сравнение времени работы и длины результирующей формулы реализации и поддерживающего элиминацию кванторов SMT-решателя

## Выбор SMT-решателя и формата ввода

- Существует несколько поддерживаемых, конкурентоспособных SMT-решателей, работающих с двоичными векторами: Boolector, Z3, CVC4 и другие
- Boolector специализируется на теории битовых векторов, а также в течение многих лет побеждал в ежегодном соревновании между SMT-солверами — The SMT Competition
- Из различных форматов ввода формул над битовыми векторами (Btor, Btor2, SMT-LIB, SMT-LIB v.2) выбран стандарт SMT-LIB v.2

## Сравнение теорий

Обогащенная арифметика Пресбургера	Теория битовых векторов (синтаксис SMT-LIB)
Носитель: $\mathbb{N}$	Носитель: битовые векторы фикс. размеров ( <code>_ BitVec n</code> )
$t_1 \leq t_2$	<code>bvslte t<sub>1</sub> t<sub>2</sub> (bvulte t<sub>1</sub> t<sub>2</sub>)</code>
$t_1 + t_2$	<code>bvadd t<sub>1</sub> t<sub>2</sub></code>
$2^t$	<code>bvshl 1 t</code>
	<code>bvand, bvor, bvnot, bvslt/bvult,</code> <code>bvsgt(e)/bvugt(e)</code> и другие

- По алгоритму, предложенному Франсуазой Пуан, можно проэлиминировать только ограниченное подмножество формул в арифметике битовых векторов без сведения к арифметике натуральных чисел



## Преобразование формул из $BV_n$ в $PA + 2^x$

$$Tr(\varphi \wedge \psi) = Tr(\varphi) \wedge Tr(\psi)$$

$$Tr(\varphi \vee \psi) = Tr(\varphi) \vee Tr(\psi)$$

$$Tr(\varphi \rightarrow \psi) = Tr(\varphi) \rightarrow Tr(\psi)$$

$$Tr(\neg\varphi) = \neg Tr(\varphi)$$

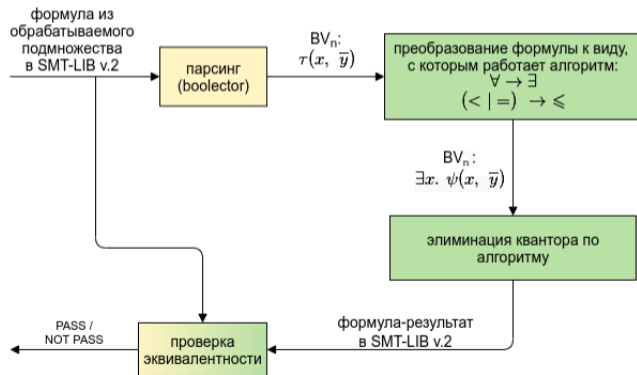
$$Tr(t_1 \text{ op } t_2) = (Tr(t_1) \text{ op } Tr(t_2)) \text{ mod } 2^n$$

$$Tr(x) = x$$

- При работе над реализацией этих преобразований была обнаружена неточность в предложенных Франсуазой Пуан действиях, ставящая под сомнение их завершаемость
- В настоящий момент по этой проблеме ведётся переписка с Франсуазой Пуан

## Реализация (1/2)

- Преобразование формулы к виду, с которым работает алгоритм
- Элиминация кванторов по алгоритму
- Вывод результирующей формулы в теории битовых векторов



Boolector:

- Си
- Парсер преобразует полученную формулу
- Формулы хранятся не в виде AST, а в стеке подвыражений

Идея проверки формул на эквивалентность:

- Пусть  $\varphi$  — исходная формула, а  $\theta$  — результат, тогда  $\varphi \oplus \theta$  должна быть невыполнима

# Экспериментальные исследования

Тест	Boolector ( $PA + 2^x$ )			Z3		
	Среднее время (мс)	Стандартное отклонение	Длина формулы	Среднее время (мс)	Стандартное отклонение	Длина формулы
1	0.005	$< 10^{-3}$	1	1.115	0.025	1
2	0.010	0.005	1	4.345	0.038	1
3	0.028	0.001	49	4.814	0.040	2210
4	0.011	0.001	62	26077.601	232.020	287204
5	0.013	0.004	61	21.335	0.144	14431
6	0.012	0.001	66	127.824	0.462	62412
7	0.023	0.004	1	0.820	0.026	1
8	0.008	0.001	41	4.332	0.037	2361
9	0.026	0.006	1	4.831	0.040	1
10	0.009	0.001	51	18.155	0.398	14837
11	0.011	0.001	95	16.147	0.072	15171
12	0.031	0.002	76	126.020	0.700	88648
13	0.011	0.001	79	70.078	0.294	44904
14	0.050	0.006	1	4.760	0.041	1
15	0.057	0.009	395	20.058	0.094	716
16	0.107	0.013	1	122.765	0.424	1

- Выбран SMT-решатель, в рамках которого осуществлялась реализация описанного алгоритма
- Реализован<sup>2</sup> алгоритм элиминации кванторов для следующих формул (вместо  $\leq$  может быть  $<$  или  $=$ , а вместо  $\exists - \forall$ ):
  - 1)  $\exists x. \bigwedge_{i,j} (g_i(\bar{y}) \leq x \wedge x \leq g_j(\bar{y}))$ ,  
где  $g_i(\bar{y}), g_j(\bar{y})$  — термы в арифметике битовых векторов, представляющие из себя линейные комбинации констант, свободных переменных ( $\bar{y}$ ) и сдвигов  $1 \ll y_k$
  - 2)  $\exists x. \bigwedge_i ((1 \ll x) \leq g_i(\bar{y}) \vee g_i(\bar{y}) \leq (1 \ll x))$
- Проведено сравнение реализации с элиминацией кванторов SMT-решателем Z3 и выяснено, что на указанном подмножестве она выдаёт более короткую формулу и работает быстрее, чем Z3

<sup>2</sup>[https://github.com/AnzhelaSukhanova/QE\\_expPA](https://github.com/AnzhelaSukhanova/QE_expPA)

## Тесты (1/2)

- 1)  $\exists x. x \geq 9505 \ (n = 16)^3$
- 2)  $\exists x. y \leq x \wedge 2 \leq x \wedge z \leq x \ (n = 4)$
- 3)  $\forall x. 3 \cdot y \leq x \wedge x \leq 12 \cdot y \ (n = 4)$
- 4)  $\exists x. x \leq 997 \cdot y \wedge z \leq x \wedge x \leq t \ (n = 10)$
- 5)  $\exists x. x \leq 2 \cdot y + z \wedge 10 \cdot y \leq x \ (n = 6)$
- 6)  $\exists x. x \leq 5 \cdot y + 7 \wedge 8 \cdot (y + z) \leq x \ (n = 8)$
- 7)  $\exists x. y + 15 < x \wedge x < 1 \ (n = 4)$
- 8)  $\exists x. 3 \cdot (1 \ll y) \leq x \wedge x \leq 7 \cdot (1 \ll y) \ (n = 4)$

---

<sup>3</sup>тест из набора Benchmarks

- 9)  $\forall x. (1 \ll y) \leq x \wedge 2 \leq x \wedge z \leq x \quad (n = 4)$
- 10)  $\exists x. 3 \cdot (1 \ll y) \leq x \wedge x \leq 12 \cdot y \quad (n = 6)$
- 11)  $\exists x. x \leq 3 \cdot (1 \ll y) \wedge (1 \ll z) \leq x \wedge x \leq t \quad (n = 6)$
- 12)  $\forall x. x \leq 2 \cdot (1 \ll y) + (1 \ll z) \wedge 10 \cdot (1 \ll y) \leq x \quad (n = 8)$
- 13)  $\exists x. x \leq 5 \cdot (1 \ll y) + 7 \wedge 8 \cdot ((1 \ll y) + z) \leq x \quad (n = 8)$
- 14)  $\exists x. (1 \ll x) \leq (1 \ll y) + 11 \cdot y + 4 \quad (n = 4)$
- 15)  $\exists x. (1 \ll x) \leq y + 3 \cdot z + 8 \quad (n = 6)$
- 16)  $\exists x. (1 \ll x) \leq 7 \cdot y \wedge (1 \ll x) \leq z \wedge (1 \ll x) \leq (1 \ll t) \quad (n = 8)$

- А. Л. Семёнов, О некоторых расширениях арифметики сложения натуральных чисел, Изв. АН СССР. Сер. матем., 1979, том 43, выпуск 5, 1175–1195
- Point Française. On the expansion  $(\mathbb{N}, +, 2^x)$  of Presburger arithmetic. — 2007. — 01