

Автоматический вывод индуктивных инвариантов программ с алгебраическими типами данных

Костюков Юрий Олегович

Научный руководитель:
д. т. н., доцент Кознов Дмитрий Владимирович

2024



Санкт-Петербургский
государственный
университет

Содержание

Обзор предметной области

Постановка задачи

Результаты

Научная новизна

Публикации и выступления

Верификация программ путём вывода индуктивных инвариантов

```
x, y := 0, 0
```

```
while * do
```

```
    y := y + x
```

```
    x := x + 1
```

```
assert( $y \geq 0$ )
```

Верификация программ путём вывода индуктивных инвариантов

$\{x = 0 \wedge y = 0\}$

while * **do**

$y := y + x$

$x := x + 1$

$\{y \geq 0\}$

Верификация программ путём вывода индуктивных инвариантов

$\{x = 0 \wedge y = 0\}$

while * **do**

$y := y + x$

$x := x + 1$

$\{y \geq 0\}$

Как доказать корректность этой тройки Хоара?

Верификация программ путём вывода индуктивных инвариантов

$\{x = 0 \wedge y = 0\}$

while * **do**

$y := y + x$

$x := x + 1$

$\{y \geq 0\}$

Как доказать корректность этой тройки Хоара?

При помощи *пользовательского* индуктивного инварианта φ

$$x = 0 \wedge y = 0 \rightarrow \varphi(x, y)$$

$$\varphi(x, y) \wedge x' = x + 1 \wedge y' = y + x \rightarrow \varphi(x', y')$$

$$\varphi(x, y) \rightarrow y \geq 0$$

Верификация программ путём вывода индуктивных инвариантов

$\{x = 0 \wedge y = 0\}$

while * **do**

$y := y + x$

$x := x + 1$

$\{y \geq 0\}$

Как доказать корректность этой тройки Хоара?

При помощи *пользовательского индуктивного инварианта* φ

Пользователь: $y \geq 0$ — индуктивный инвариант?

$$x = 0 \wedge y = 0 \rightarrow \varphi(x, y)$$

$$\varphi(x, y) \wedge x' = x + 1 \wedge y' = y + x \rightarrow \varphi(x', y')$$

$$\varphi(x, y) \rightarrow y \geq 0$$

Верификация программ путём вывода индуктивных инвариантов

$\{x = 0 \wedge y = 0\}$

while * **do**

$y := y + x$

$x := x + 1$

$\{y \geq 0\}$

Как доказать корректность этой тройки Хоара?

При помощи *пользовательского индуктивного инварианта* φ

Пользователь: $y \geq 0$ — индуктивный инвариант?

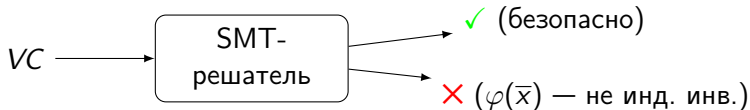
$$VC := \left\{ \begin{array}{l} \forall x, y. (x = 0 \wedge y = 0 \rightarrow y \geq 0) \wedge \\ \forall x, y, x', y'. (y \geq 0 \wedge x' = x + 1 \wedge y' = y + x \rightarrow y' \geq 0) \wedge \\ \forall x, y. (y \geq 0 \rightarrow y \geq 0) \end{array} \right.$$

Верификация программ путём вывода индуктивных инвариантов

Как доказать корректность этой тройки Хоара?

При помощи *пользовательского индуктивного инварианта* φ

Пользователь: $y \geq 0$ — индуктивный инвариант?



$$VC := \left\{ \begin{array}{l} \forall x, y. (x = 0 \wedge y = 0 \rightarrow y \geq 0) \wedge \\ \forall x, y, x', y'. (y \geq 0 \wedge x' = x + 1 \wedge y' = y + x \rightarrow y' \geq 0) \wedge \\ \forall x, y. (y \geq 0 \rightarrow y \geq 0) \end{array} \right.$$

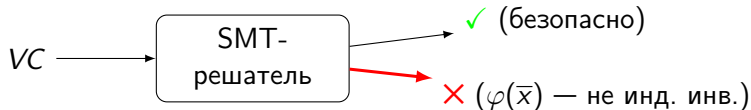
Верификация программ путём вывода индуктивных инвариантов

Как доказать корректность этой тройки Хоара?

При помощи *пользовательского индуктивного инварианта* φ

Пользователь: $y \geq 0$ — индуктивный инвариант?

SMT-решатель: Нет, индуктивность нарушается при $x \mapsto -1$



$$VC := \left\{ \begin{array}{l} \forall x, y. (x = 0 \wedge y = 0 \rightarrow y \geq 0) \wedge \\ \forall x, y, x', y'. (y \geq 0 \wedge x' = x + 1 \wedge y' = y + x \rightarrow y' \geq 0) \wedge \\ \forall x, y. (y \geq 0 \rightarrow y \geq 0) \end{array} \right.$$

Верификация программ путём вывода индуктивных инвариантов

Как доказать корректность этой тройки Хоара?

При помощи *пользовательского* индуктивного инварианта φ

Пользователь: $y \geq 0$ — индуктивный инвариант?

SMT-решатель: Нет, индуктивность нарушается при $x \mapsto -1$

Пользователь: А усиленная формула: $x \geq 0 \wedge y \geq 0$?

$$x = 0 \wedge y = 0 \rightarrow \varphi(x, y)$$

$$\varphi(x, y) \wedge x' = x + 1 \wedge y' = y + x \rightarrow \varphi(x', y')$$

$$\varphi(x, y) \rightarrow y \geq 0$$

Верификация программ путём вывода индуктивных инвариантов

Как доказать корректность этой тройки Хоара?

При помощи *пользовательского* индуктивного инварианта φ

Пользователь: $y \geq 0$ — индуктивный инвариант?

SMT-решатель: Нет, индуктивность нарушается при $x \mapsto -1$

Пользователь: А усиленная формула: $x \geq 0 \wedge y \geq 0$?

SMT-решатель: Да, эта формула является индуктивным инвариантом

$$x = 0 \wedge y = 0 \rightarrow \varphi(x, y)$$

$$\varphi(x, y) \wedge x' = x + 1 \wedge y' = y + x \rightarrow \varphi(x', y')$$

$$\varphi(x, y) \rightarrow y \geq 0$$

Дизъюнкты Хорна с ограничениями

Как автоматизировать вывод индуктивных инвариантов?

$$x = 0 \wedge y = 0 \rightarrow I(x, y)$$

$$I(x, y) \wedge x' = x + 1 \wedge y' = y + x \rightarrow I(x', y')$$

$$I(x, y) \rightarrow y \geq 0$$

Дизъюнкты Хорна с ограничениями

Как автоматизировать вывод индуктивных инвариантов?

Заменить пользовательскую формулу на неинтерпретированный символ I


$$\begin{aligned}x = 0 \wedge y = 0 &\rightarrow I(x, y) \\ I(x, y) \wedge x' = x + 1 \wedge y' = y + x &\rightarrow I(x', y') \\ I(x, y) &\rightarrow y \geq 0\end{aligned}$$

Дизъюнкты Хорна с ограничениями

Как автоматизировать вывод индуктивных инвариантов?

Заменить пользовательскую формулу на неинтерпретированный символ I

Дизъюнкты Хорна с ограничениями



The diagram shows a red box containing the text "Дизъюнкты Хорна" and a green box containing the text "с ограничениями". Both boxes have arrows pointing to the expression $x = 0 \wedge y = 0$ in the first line of the formula below.

$$\begin{aligned} & x = 0 \wedge y = 0 \rightarrow I(x, y) \\ I(x, y) \wedge x' = x + 1 \wedge y' = y + x & \rightarrow I(x', y') \\ & I(x, y) \rightarrow y \geq 0 \end{aligned}$$

Дизъюнкты Хорна формально

Дизъюнкт Хорна C — это формула первого порядка следующего вида:

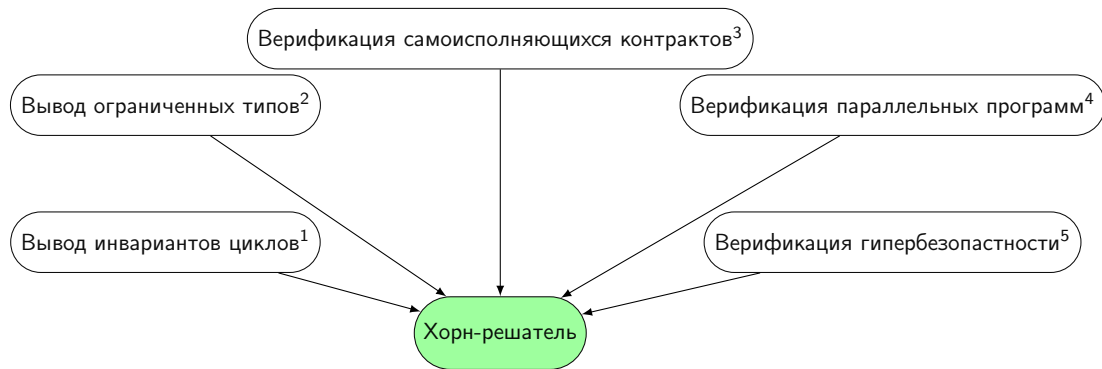
$$\varphi \wedge P_1(\bar{x}_1) \wedge \dots \wedge P_n(\bar{x}_n) \rightarrow H$$

- ▶ **ограничение** φ — это формула теории
- ▶ **голова** H — это либо ложь \perp , либо атом $P(\bar{x})$
- ▶ P_1, \dots, P_n, P — это неинтерпретированные символы
- ▶ все переменные (неявно) универсально квантифицированы

Система дизъюнктов Хорна — это конъюнкция дизъюнктов Хорна

Хорн-решатель — программа, проверяющая выполнимость системы дизъюнктов

Применения Хорн-решателей



¹ Gurfinkel и др. The SeaHorn Verification Framework. CAV'15

² Tan и др. SolType: refinement types for arithmetic overflow in solidity. POPL'22

³ Alt и др. SolCMC: Solidity Compiler's Model Checker. CAV'22

⁴ Hoenicke и др. Thread Modularity at Many Levels. POPL'17

⁵ Shemer и др. Property Directed Self Composition. CAV'19

Дизъюнкты Хорна над алгебраическими типами данных (АТД)

Пример программы на языке `HASKELL`:

```
data Nat = Z | S Nat
data List = nil | cons Nat List
drop Z xs = xs
drop _ nil = nil
drop (S n) (cons(_, xs)) = drop n xs
assert (¬∃ n xs . xs /= nil && drop n xs == drop (S n) xs)
```

Условия верификации в виде дизъюнктов Хорна над АТД:

$$\top \rightarrow \text{drop}(Z, xs, xs)$$

$$\top \rightarrow \text{drop}(S(n), nil, nil)$$

$$\text{drop}(n, xs, rs) \rightarrow \text{drop}(S(n), \text{cons}(x, xs), rs)$$

$$\neg(xs = nil) \wedge \text{drop}(n, xs, ys) \wedge \text{drop}(S(n), xs, ys) \rightarrow \perp$$

Индуктивный инвариант

Пусть \mathcal{H} — модель теории АД, \mathcal{S} — система дизъюнктов Хорна.

Индуктивный инвариант \mathcal{I} — расширение модели $\mathcal{I} = \langle \mathcal{H}, \mathcal{R} \rangle$, такое что $\mathcal{I} \models \mathcal{S}$.

Индуктивный инвариант

Пусть \mathcal{H} — модель теории АТД, \mathcal{S} — система дизъюнктов Хорна.

Индуктивный инвариант \mathcal{I} — расширение модели $\mathcal{I} = \langle \mathcal{H}, \mathcal{R} \rangle$, такое что $\mathcal{I} \models \mathcal{S}$.

$$x = Z \wedge y = S(Z) \rightarrow inc(x, y)$$

$$x' = S(x) \wedge y' = S(y) \wedge inc(x, y) \rightarrow inc(x', y')$$

$$x = y \wedge inc(x, y) \rightarrow \perp$$

$$\mathcal{I}_1 = \mathcal{H} \left\{ inc \mapsto \{(x, y) \mid y = S(x)\} \right\}$$

$$\mathcal{I}_2 = \mathcal{H} \left\{ inc \mapsto \{(x, y) \mid x \neq y\} \right\}$$

$$\mathcal{I}_3 = \dots$$

Индуктивные инварианты составляют решётку

Индуктивный инвариант

Пусть \mathcal{H} — модель теории АТД, \mathcal{S} — система дизъюнктов Хорна.

Индуктивный инвариант \mathcal{I} — расширение модели $\mathcal{I} = \langle \mathcal{H}, \mathcal{R} \rangle$, такое что $\mathcal{I} \models \mathcal{S}$.

$$x = Z \wedge y = S(Z) \rightarrow inc(x, y)$$

$$x' = S(x) \wedge y' = S(y) \wedge inc(x, y) \rightarrow inc(x', y')$$

$$x = y \wedge inc(x, y) \rightarrow \perp$$

$$\mathcal{I}_1 = \mathcal{H} \left\{ inc \mapsto \{(x, y) \mid y = S(x)\} \right\}$$

$$\mathcal{I}_2 = \mathcal{H} \left\{ inc \mapsto \{(x, y) \mid x \neq y\} \right\}$$

$$\mathcal{I}_3 = \dots$$

Как представлять эти бесконечные множества?

Индуктивный инвариант

Пусть \mathcal{H} — модель теории АТД, \mathcal{S} — система дизъюнктов Хорна.

Индуктивный инвариант \mathcal{I} — расширение модели $\mathcal{I} = \langle \mathcal{H}, \mathcal{R} \rangle$, такое что $\mathcal{I} \models \mathcal{S}$.

$$x = Z \wedge y = S(Z) \rightarrow inc(x, y)$$

$$x' = S(x) \wedge y' = S(y) \wedge inc(x, y) \rightarrow inc(x', y')$$

$$x = y \wedge inc(x, y) \rightarrow \perp$$

$$\mathcal{I}_1 = \mathcal{H} \left\{ inc \mapsto \boxed{y = S(x)} \right\}$$

$$\mathcal{I}_2 = \mathcal{H} \left\{ inc \mapsto \boxed{\neg(x = y)} \right\}$$

$$\mathcal{I}_3 = \dots$$

Как представлять эти **бесконечные** множества?

Инварианты обычно представляются в логике первого порядка (ЛПП)

ЛПП задаёт т.н. *класс элементарных инвариантов*

Проблема выразимости класса элементарных инвариантов

$$x = Z \rightarrow \textit{even}(x)$$

$$\textit{even}(y) \wedge x = S(S(y)) \rightarrow \textit{even}(x)$$

$$\textit{even}(x) \wedge \textit{even}(S(x)) \rightarrow \perp$$

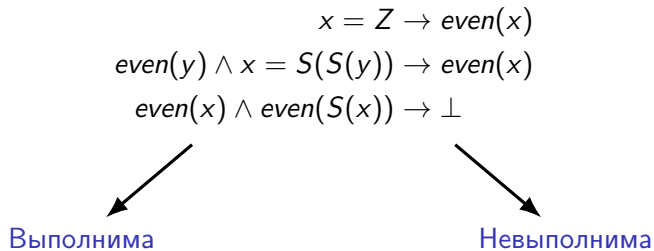
Проблема выразимости класса элементарных инвариантов

$$\begin{aligned}x = Z &\rightarrow \text{even}(x) \\ \text{even}(y) \wedge x = S(S(y)) &\rightarrow \text{even}(x) \\ \text{even}(x) \wedge \text{even}(S(x)) &\rightarrow \perp\end{aligned}$$



Невыполнима

Проблема выразимости класса элементарных инвариантов



Проблема выразимости класса элементарных инвариантов

$$x = Z \rightarrow \text{even}(x)$$

$$\text{even}(y) \wedge x = S(S(y)) \rightarrow \text{even}(x)$$

$$\text{even}(x) \wedge \text{even}(S(x)) \rightarrow \perp$$

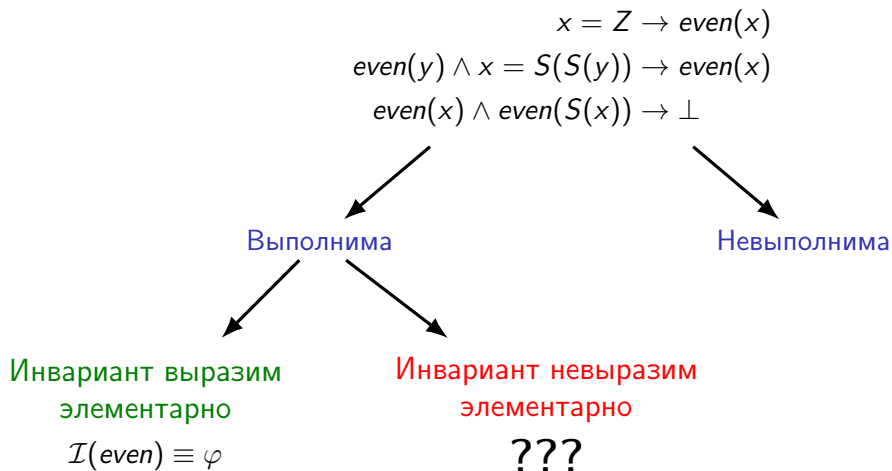
Выполнима

Невыполнима

Инвариант выразим
элементарно

$$\mathcal{I}(\text{even}) \equiv \varphi$$

Проблема выразимости класса элементарных инвариантов



Проблема выразимости класса элементарных инвариантов

$$x = Z \rightarrow \text{even}(x)$$

$$\text{even}(y) \wedge x = S(S(y)) \rightarrow \text{even}(x)$$

$$\text{even}(x) \wedge \text{even}(S(x)) \rightarrow \perp$$

Вывод инвариантов в языке АТД расходится!

например, Z3/SPACER расходится

Выполнима

Невыполнима

Инвариант выразим
элементарно

$$\mathcal{I}(\text{even}) \equiv \varphi$$

Инвариант невыразим
элементарно

???

Проблема выразимости класса элементарных инвариантов

$$x = Z \rightarrow \text{even}(x)$$

$$\text{even}(y) \wedge x = S(S(y)) \rightarrow \text{even}(x)$$

$$\text{even}(x) \wedge \text{even}(S(x)) \rightarrow \perp$$

Проблема: класс элементарных инвариантов невыразителен

Выполнима

Невыполнима

Инвариант выразим
элементарно

$$\mathcal{I}(\text{even}) \equiv \varphi$$

Инвариант невыразим
элементарно

???

Постановка задачи

Цель работы — предложение новых классов индуктивных инвариантов для программ с АТД и создание для них методов автоматического вывода. **Задачи:**

1. Предложить методы вывода инвариантов в существующих классах
2. Предложить новый класс индуктивных инвариантов программ с АТД
3. Предложить метод автоматического вывода инвариантов в новом классе
4. Выполнить пилотную программную реализацию предложенных методов
5. Провести экспериментальное сопоставление реализованного инструмента с существующими на представительном тестовом наборе

Результаты

1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность
2. Предложен метод вывода синхронных регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность
3. Предложен новый класс инвариантов, основанный на булевой комбинации элементарных и регулярных инвариантов
4. Предложен метод совместного вывода инвариантов в этом классе посредством вывода инвариантов в подклассах и доказана его корректность
5. Проведено теоретическое сравнение рассмотренных классов инвариантов, доказаны леммы о накачке для языков первого порядка
6. Выполнена пилотная реализация предложенных методов на языке F# в рамках инструмента RINGEN
Разработанный инструмент решил из бенчмарка «Tons of Inductive Problems» в 3.74 раза больше задач, чем наилучший из существующих инструментов

Результат 1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 1. Устранить АД ограничения при помощи унификации и введения новых дизъюнктов

Система дизъюнктов

$$\top \rightarrow \text{even}(Z)$$

$$\text{even}(y) \rightarrow \text{even}(S(S(y)))$$

$$\text{even}(x) \wedge \text{even}(S(x)) \rightarrow \perp$$

АД ограничения устранены

Результат 1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 2. Трансформировать систему в формулу ЛПП введением логических связок

Система дизъюнктов как формула ЛПП

$$\begin{aligned} & \top \rightarrow \text{even}(Z)) \wedge \\ & \forall y.(\text{even}(y) \rightarrow \text{even}(S(S(y)))) \wedge \\ & \forall x.(\text{even}(x) \wedge \text{even}(S(x)) \rightarrow \perp) \end{aligned}$$

Результат 1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 3. Передать формулу в сторонний инструмент поиска конечных моделей

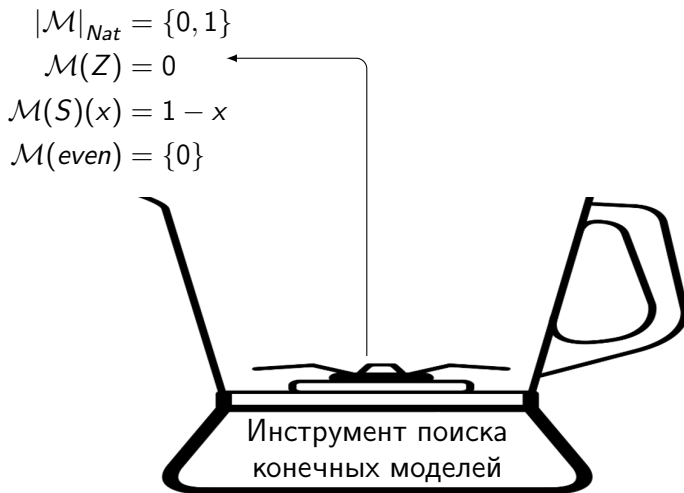
Система дизъюнктов как формула ЛПП

$$\begin{aligned} & \top \rightarrow \text{even}(Z)) \wedge \\ & \forall y.(\text{even}(y) \rightarrow \text{even}(S(S(y)))) \wedge \\ & \forall x.(\text{even}(x) \wedge \text{even}(S(x)) \rightarrow \perp) \end{aligned}$$



Результат 1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 3. Передать формулу в сторонний инструмент поиска конечных моделей



Результат 1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 4. По конечной модели построить автомат над деревьями

$$|\mathcal{M}|_{Nat} = \{0, 1\}$$

$$\mathcal{M}(Z) = 0$$

$$\mathcal{M}(S)(x) = 1 - x$$

$$\mathcal{M}(even) = \{0\}$$



Результат 1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 4. По конечной модели построить автомат над деревьями

$$|\mathcal{M}|_{Nat} = \{0, 1\}$$

$$\mathcal{M}(Z) = 0$$

$$\mathcal{M}(S)(x) = 1 - x$$

$$\mathcal{M}(even) = \{0\}$$



Результат 1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

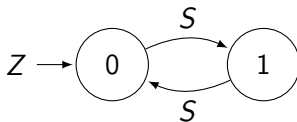
Этап 4. По конечной модели построить автомат над деревьями

$$|\mathcal{M}|_{Nat} = \{0, 1\}$$

$$\mathcal{M}(Z) = 0$$

$$\mathcal{M}(S)(x) = 1 - x$$

$$\mathcal{M}(even) = \{0\}$$



Результат 1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

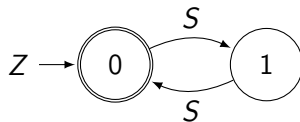
Этап 4. По конечной модели построить автомат над деревьями

$$|\mathcal{M}|_{Nat} = \{0, 1\}$$

$$\mathcal{M}(Z) = 0$$

$$\mathcal{M}(S)(x) = 1 - x$$

$$\mathcal{M}(\text{even}) = \{0\}$$



Результат 1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

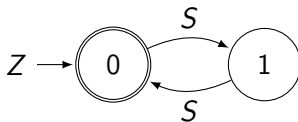
Этап 4. По конечной модели построить автомат над деревьями

$$|\mathcal{M}|_{Nat} = \{0, 1\}$$

$$\mathcal{M}(Z) = 0$$

$$\mathcal{M}(S)(x) = 1 - x$$

$$\mathcal{M}(even) = \{0\}$$



Язык построенного автомата является регулярным инвариантом исходной системы

$$\mathcal{I}(even) = \mathcal{L}(\mathcal{A}) = \{S^{2n}(Z) \mid n \geq 0\}$$

Результат 2. Предложен метод вывода синхронных регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Регулярные языки не позволяют
представлять синхронные отношения

$$\top \rightarrow lt(Z, S(x))$$

$$lt(x, y) \rightarrow lt(S(x), S(y))$$

$$lt(x, y) \wedge lt(y, x) \rightarrow \perp$$

Результат 2. Предложен метод вывода синхронных регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 1. Устранить АТД ограничения при помощи унификации и введения новых дизъюнктов

Регулярные языки не позволяют
представлять синхронные отношения

$$\top \rightarrow lt(Z, S(x))$$

$$lt(x, y) \rightarrow lt(S(x), S(y))$$

$$lt(x, y) \wedge lt(y, x) \rightarrow \perp$$

Результат 2. Предложен метод вывода синхронных регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 2. Построить декларативное описание синхронного автомата, выражающего инвариант системы

Регулярные языки не позволяют
представлять синхронные отношения

$$\begin{aligned}\top &\rightarrow It(Z, S(x)) \\ It(x, y) &\rightarrow It(S(x), S(y)) \\ It(x, y) \wedge It(y, x) &\rightarrow \perp\end{aligned}$$

Декларативное описание
синхронного автомата

$$\begin{aligned}R(q) &\rightarrow R(p(d(f, g, q), d(f, g, q))) \\ R(p(q_1, q_2)) &\rightarrow (F(q_1) \rightarrow F(d(S, S, q_2))) \\ &\dots\end{aligned}$$

Результат 2. Предложен метод вывода синхронных регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 3. Передать формулу в сторонний инструмент поиска конечных моделей

Регулярные языки не позволяют
представлять синхронные отношения

$$\begin{aligned}\top &\rightarrow It(Z, S(x)) \\ It(x, y) &\rightarrow It(S(x), S(y)) \\ It(x, y) \wedge It(y, x) &\rightarrow \perp\end{aligned}$$

Декларативное описание
синхронного автомата

$$\begin{aligned}R(q) &\rightarrow R(p(d(f, g, q), d(f, g, q))) \\ R(p(q_1, q_2)) &\rightarrow (F(q_1) \rightarrow F(d(S, S, q_2))) \\ &\dots\end{aligned}$$

Результат 2. Предложен метод вывода синхронных регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность

Этап 4. По конечной модели построить автомат над деревьями

Регулярные языки не позволяют
представлять синхронные отношения

Декларативное описание
синхронного автомата

$$\begin{aligned}\top &\rightarrow It(Z, S(x)) \\ It(x, y) &\rightarrow It(S(x), S(y)) \\ It(x, y) \wedge It(y, x) &\rightarrow \perp\end{aligned}$$

$$\begin{aligned}R(q) &\rightarrow R(p(d(f, g, q), d(f, g, q))) \\ R(p(q_1, q_2)) &\rightarrow (F(q_1) \rightarrow F(d(S, S, q_2))) \\ &\dots\end{aligned}$$

Из модели можно извлечь определение синхронного автомата

$$A = \langle \{0, 1\}, \Sigma_F^{\leq 2}, \{1\}, \Delta \rangle$$

$$\begin{array}{lll}\langle Z, Z \rangle \mapsto 0 & Z \mapsto 0 & S(q) \mapsto 0 \\ \langle Z, S \rangle(q) \mapsto 1 & \langle S, Z \rangle(q) \mapsto 0 & \langle S, S \rangle(q) \mapsto q\end{array}$$

$$\mathcal{L}(A) = \{ \langle S^n(Z), S^m(Z) \rangle \mid n < m \}$$

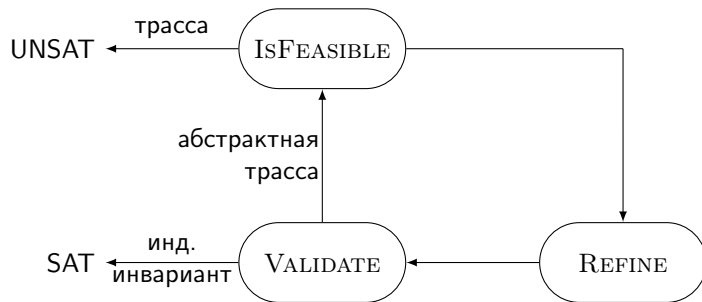
Результат 3. Предложен новый класс инвариантов, основанный на булевой комбинации элементарных и регулярных инвариантов

Новый класс *комбинированных инвариантов* представляется формулами вида:

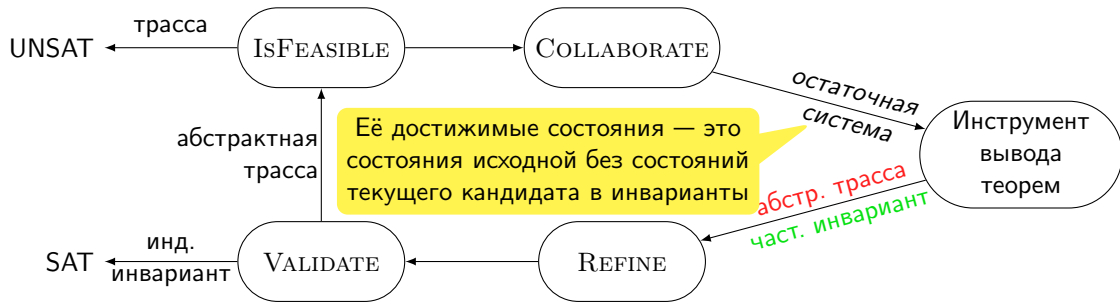
$$\varphi ::= \bar{t} \in \mathcal{L}(A) \mid t = t' \mid \neg\psi \mid \psi \wedge \psi' \mid \psi \vee \psi'$$

- $\bar{t} \in \mathcal{L}(A)$ — принадлежность кортежа термов регулярном языке автомата A

Результат 4. Предложен метод совместного вывода инвариантов в этом классе посредством вывода инвариантов в подклассах и доказана его корректность



Результат 4. Предложен метод совместного вывода инвариантов в этом классе посредством вывода инвариантов в подклассах и доказана его корректность



Результат 5. Проведено теоретическое сравнение рассмотренных классов инвариантов, доказаны леммы о накачке

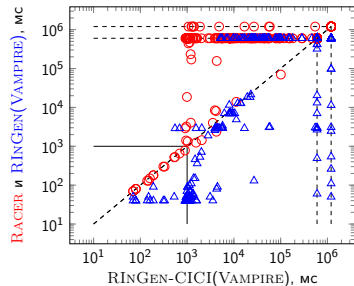
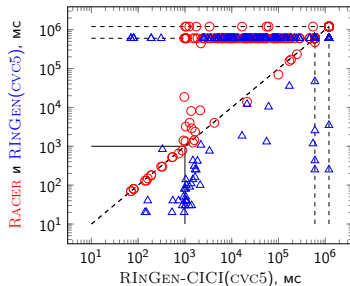
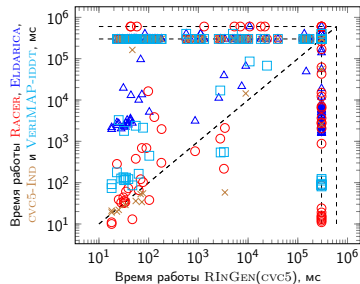
Результаты, доказанные в диссертации; результаты с тривиальным доказательством

Класс \ Свойство	ELEM	SIZEELEM	REG	REG ₊	REG _×	ELEMREG
Замкнут по \cap	Да	Да	Да	Да	Да	Да
Замкнут по \cup	Да	Да	Да	Да	Да	Да
Замкнут по \setminus	Да	Да	Да	Да	Да	Да
Разрешимо $\bar{t} \in I$	Да	Да	Да	Да	Да	Да
Разрешимо $I = \emptyset$	Да	Да	Да	Да	Да	Да
Выразимы рекурсивные отношения	Нет	Частично	Да	Да	Да	Да
Выразимы синхронные отношения	Да	Да	Нет	Частично	Да	Да

Класс	ELEM	SIZEELEM	REG	REG ₊	REG _×	ELEMREG
ELEM	\emptyset	\emptyset	<i>lr</i>	<i>lr</i>	<i>lr</i>	\emptyset
SIZEELEM	∞	\emptyset	<i>lr</i>	<i>lr</i>	<i>lr</i>	<i>lt</i>
REG	<i>even</i>	<i>even</i>	\emptyset	\emptyset	\emptyset	\emptyset
REG ₊	<i>even</i>	<i>even</i>	∞	\emptyset	\emptyset	<i>lt</i>
REG _×	<i>even</i>	<i>even</i>	∞	∞	\emptyset	<i>lt</i>
ELEMREG	∞	<i>even</i>	∞	<i>lr</i>	<i>lr</i>	\emptyset

Результат 6. Выполнена реализация и проведены эксперименты

Инструмент	SAT	UNSAT
RACER	26	22
ELDARICA	46	12
VERICAT	16	10
CVC5-IND	0	13
RInGen(CVC5)	25	21
RInGen(VAMPIRE)	135	46
RInGen-SYNC	43	21
RInGen-CICI(CVC5)	117	19
RInGen-CICI(VAMPIRE)	189	28



Результаты

1. Предложен метод вывода регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность
2. Предложен метод вывода синхронных регулярных инвариантов при помощи поиска конечных моделей и доказана его корректность
3. Предложен новый класс инвариантов, основанный на булевой комбинации элементарных и регулярных инвариантов
4. Предложен метод совместного вывода инвариантов в этом классе посредством вывода инвариантов в подклассах и доказана его корректность
5. Проведено теоретическое сравнение рассмотренных классов инвариантов, доказаны леммы о накачке для языков первого порядка
6. Выполнена пилотная реализация предложенных методов на языке F# в рамках инструмента RINGEN
Разработанный инструмент решил из бенчмарка «Tons of Inductive Problems» в 3.74 раза больше задач, чем наилучший из существующих инструментов

Соответствие результатов паспорту специальности 2.3.5

Результаты соответствуют направлению исследования № 1

- ▶ Модели, **методы и алгоритмы** проектирования, анализа, трансформации, **верификации** и тестирования **программ** и программных систем

из паспорта специальности.

Научная новизна

1. Впервые предложены алгоритмы вывода регулярных и синхронных регулярных инвариантов для программ с АТД, **отличающиеся тем, что** с целью автоматизации перебора автоматов над деревьями используют поиск конечных моделей
2. Впервые предложен класс комбинированных инвариантов, **отличающийся тем, что** с целью объединения выразительных возможностей классов элементарных и регулярных инвариантов предложенный класс построен как их булева комбинация
3. Предложен новый алгоритм вывода инвариантов в классе комбинированных инвариантов, **отличающийся тем, что** с целью переиспользования эффективных алгоритмов для вывода инвариантов в комбинируемых классах применяет их на промежуточных артефактах друг друга
4. Впервые введены и доказаны леммы о «накачке» для языков первого порядка **отличающиеся тем, что** с целью отделения классов элементарных инвариантов от других классов используют неспособность языков первого порядка различать заранее не ограниченные по высоте термы

Публикации по теме диссертации

- [1] Kostyukov Yurii, Mordvinov Dmitry, Fedyukovich Grigory. Beyond the Elementary Representations of Program Invariants over Algebraic Data Types // Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation. — PLDI 2021. — New York, NY, USA : Association for Computing Machinery, 2021. — P. 451–465.
- [2] Kostyukov Yurii, Mordvinov Dmitry, Fedyukovich Grigory. Collaborative Inference of Combined Invariants // Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning / Ed. by Ruzica Piskac, Andrei Voronkov. — Vol. 94 of EPIc Series in Computing. — EasyChair, 2023. — P. 288–305.
- [3] Автоматическое доказательство корректности программ с динамической памятью / Юрий Олегович Костюков, Константин Аланович Батоев, Дмитрий Александрович Мордвинов и др. // Труды Института системного программирования РАН. — 2019. — Т. 31, № 5. — С. 37–62.
- [4] Генерация слабейших предусловий программ с динамической памятью в символьном исполнении / Александр Владимирович Мисонижник, Юрий Олегович Костюков, Михаил Павлович Костицын и др. // Научно-технический вестник информационных технологий, механики и оптики. — 2022. — Т. 22, № 5. — С. 982–991.

Выступления по теме диссертации

- ▶ Международный семинар HCVS 2021 (28 марта 2021, Люксембург)
- ▶ Семинар компании Huawei (18-19 ноября 2021, Санкт-Петербург)
- ▶ Ежегодный внутренний семинар JetBrains Research (18 декабря 2021, Санкт-Петербург)
- ▶ Конференция PLDI 2021 (23-25 июня 2021, Канада)
- ▶ Внутренний семинар Венского технического университета (3 июня 2022, Австрия)
- ▶ Конференция LPAR 2023 (4-9 июня 2023, Колумбия)

Разработанный инструмент в 2021 и 2022 годах занял, соответственно, 2 и 1 место на АТД секции международных соревнований CHC-COMP.

Сравнение Хорн-решателей с поддержкой АД

Инструмент	Класс инвариантов	Метод	Возвращает инвариант	Полностью автоматический
SPACER	ELEM	IC3/PDR	Да	Да
RACER	CATELEM	IC3/PDR	Нет	Нет
ELДАРICA	SIZEELEM	CEGAR	Да	Да
VERICAT	–	Трансф.	Нет	Да
HoICE	ELEM	ICE	Да	Да
RCHC	REG ₊	ICE	Да	Да
RINGEN(CVC5)	REG	Трансф. + FMF	Да	Да
RINGEN(VAMPIRE)	–	Трансф. + Насыщение	Нет	Да
RINGEN-SYNC	REG _x	Трансф. + FMF	Да	Да
RINGEN-CICI(CVC5)	ELEMREG	CEGAR(\emptyset)	Да	Да
RINGEN-CICI(VAMPIRE)	–	CEGAR(\emptyset)	Нет	Да