Представление инвариантов программ с алгебраическими типами данных в виде синхронных древовидных автоматов

Васенина Анна Игоревна

Санкт-Петербургский государственный университет
Кафедра системного программирования

16 ноября 2023 г.

Условия верификации

Программа

```
int abs (int x) {
   if x ⩾ 0
      return x
   else
      return -x
}
assert(abs(x) ≥ 0)
```

Условия верификации

Программа

```
int abs (int x) {
  if x ⩾ 0
    return x
  else
    return -x
}
assert(abs(x) ≥ 0)
```

Условия верификации

$$x \geqslant 0 \land y = x \to abs(x, y)$$
$$x < 0 \land y = \text{-}x \to abs(x, y)$$
$$abs(x, y) \land y < 0 \to \bot$$

Условия верификации

Функция

$f : X \to Y$

$y = f(x)$

График

$G_f \subset X \times Y$

$(x, y) \in G_f \iff y = f(x)$

Условия верификации

Условия верификации

В общем виде

$$\varphi \wedge R_1(\overline{x}_1, y_1) \wedge R_2(\overline{x}_2, y_2) \wedge \ldots \wedge R_m(\overline{x}_m, y_m) \rightarrow R_0(\overline{x}_0, y_0)$$

R_i — неинтерпретированные предикатные символы, вызовы закодированных предикатными символами функций

x_i — входные значения

y_i — результаты

φ — ограничение в логике первого порядка, оставшееся тело функции

Решение

- Контрпример
- Индуктивный инвариант

Индуктивный инвариант

Программа с АТД

$$
\begin{aligned}
n = Z \land r = S(Z) &\to inc(n, r) \\
n = S(n') \land r = S(r') \land inc(n', r') &\to inc(n, r) \\
inc(n, r) \land r \neq S(n) &\to \bot
\end{aligned}
$$

Теоретико-множественный инвариант

$I = \{\langle Z, S(Z)\rangle, \langle S(Z), S(S(Z))\rangle, \langle S(S(Z)), S(S(S(Z)))\rangle, \ldots\}$

Индуктивный инвариант

Программа с АТД

$$n = Z \wedge r = S(Z) \rightarrow inc(n, r)$$
$$n = S(n') \wedge r = S(r') \wedge inc(n', r') \rightarrow inc(n, r)$$
$$inc(n, r) \wedge r \neq S(n) \rightarrow \bot$$

Теоретико-множественный инвариант

$I = \{\langle Z, S(Z)\rangle, \langle S(Z), S(S(Z))\rangle, \langle S(S(Z)), S(S(S(Z)))\rangle, \dots\}$

Конечное представление

Индуктивный инвариант

Программа с АТД

$$n = Z \land r = S(Z) \to inc(n, r)$$
$$n = S(n') \land r = S(r') \land inc(n', r') \to inc(n, r)$$
$$inc(n, r) \land r \neq S(n) \to \bot$$

Теоретико-множественный инвариант

$I = \{\langle Z, S(Z)\rangle, \langle S(Z), S(S(Z))\rangle, \langle S(S(Z)), S(S(S(Z)))\rangle, \ldots\}$

Конечное представление

- Формула логики первого порядка в теории АТД
- Формула логики первого порядка с ограничением на размер
- Язык автомата
- Язык синхронного автомата

Формулы логики первого порядка

Определение

$$\varphi ::= \ (t = t') \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \forall x.\varphi$$

Формулы логики первого порядка

Определение

$$\varphi ::= \ (t = t') \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \forall x.\varphi$$

Теоретико-множественный инвариант

$$I = \{\langle Z, S(Z)\rangle, \langle S(Z), S(S(Z))\rangle, \langle S(S(Z)), S(S(S(Z)))\rangle, \ldots\}$$

Конечное представление

$$inc(n, r) \iff r = S(n)$$

Формулы логики первого порядка

Пример

$$x = Z \to even(x)$$
$$x = S(S(y)) \land even(y) \to even(x)$$
$$even(x) \land even(y) \land y = S(x) \to \bot$$

Оценка выразительной силы

- Процедура устранения кванторов[1]

[1] Oppen D. C. Reasoning about recursively defined data structures. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. – 1978

[2] Kostyukov Y., Mordvinov D., Fedyukovich G. Beyond the elementary representations of program invariants over algebraic data types, ACM SIGPLAN International Conference on Programming Language Design and Implementation. – 2021

Пример

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

Оценка выразительной силы

- Процедура устранения кванторов[1]
- После устранения кванторов: $x = S^n(Z)$, $x = Z$
- Объединение конечных и коконечных множеств $\{x \mid x \neq 1 \wedge x \neq 2\} \cup \{x \mid x = 5 \vee x = 7\}$

[1]Oppen D. C. Reasoning about recursively defined data structures. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. – 1978

[2]Kostyukov Y., Mordvinov D., Fedyukovich G. Beyond the elementary representations of program invariants over algebraic data types, ACM SIGPLAN International Conference on Programming Language Design and Implementation. – 2021

Формулы логики первого порядка

Пример

$$x = Z \to even(x)$$
$$x = S(S(y)) \wedge even(y) \to even(x)$$
$$even(x) \wedge even(y) \wedge y = S(x) \to \bot$$

Оценка выразительной силы

- Процедура устранения кванторов[1]
- После устранения кванторов: $x = S^n(Z),\ x = Z$
- Объединение конечных и коконечных множеств $\{x \mid x \neq 1 \wedge x \neq 2\} \cup \{x \mid x = 5 \vee x = 7\}$
- Лемма о накачке[2]

[1] Oppen D. C. Reasoning about recursively defined data structures. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. – 1978

[2] Kostyukov Y., Mordvinov D., Fedyukovich G. Beyond the elementary representations of program invariants over algebraic data types, ACM SIGPLAN International Conference on Programming Language Design and Implementation. – 2021

Ограничения на размер

Определение

- $size(t)$ — количество конструкторов в t
- Пресбургеровские формулы над $size$

Ограничения на размер

Определение

- $size(t)$ — количество конструкторов в t
- Пресбургеровские формулы над *size*

Пример

$$x = Z \to even(x)$$
$$x = S(S(y)) \land even(y) \to even(x)$$
$$even(x) \land even(y) \land y = S(x) \to \bot$$

Конечное представление

$even(x) \iff (size(x) + 1) \mid 2$

Ограничения на размер

Пример

$$
\begin{aligned}
x = Leaf &\rightarrow evenLeft(x) \\
x = Node(Node(x', y), z) \wedge evenLeft(x') &\rightarrow evenLeft(x) \\
evenLeft(x) \wedge evenLeft(y) \wedge y = Node(x, z) &\rightarrow \bot
\end{aligned}
$$

Оценка выразительной силы

- Лемма о накачке[3]

[3] Kostyukov Y., Mordvinov D., Fedyukovich G. Beyond the elementary representations of program invariants over algebraic data types, ACM SIGPLAN International Conference on Programming Language Design and Implementation. – 2021

Регулярные инварианты

Пример

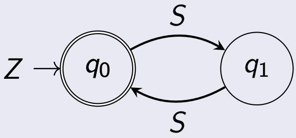
$$x = Z \to even(x)$$
$$x = S(S(y)) \wedge even(y) \to even(x)$$
$$even(x) \wedge even(y) \wedge y = S(x) \to \bot$$

Теоретико-множественный инвариант

$I = \{Z,\ S(S(Z)),\ S(S(S(Z))),\ \ldots\}$

Конечное представление

$even(x) \iff x \in L$



Регулярные инварианты

Общий случай АТД

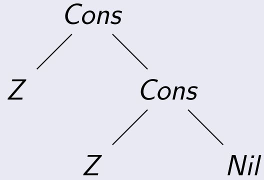
$T ::= C_0 \mid C_1(T) \mid C_2(T, T) \mid \ldots \mid C_n(T, \ldots, T)$

Регулярные инварианты

Общий случай АТД

$T ::= C_0 \mid C_1(T) \mid C_2(T, T) \mid \ldots \mid C_n(T, \ldots, T)$

Древесное представление



Регулярные инварианты

Определение

Автоматом над деревьями называют кортеж $\mathcal{A} = \langle Q, Q_F, \Delta \rangle$, где Q — конечное множество состояний, $Q_F \subseteq Q^n$ — множество финальных состояний, Δ – отношение перехода с правилами следующего вида:

$$F(s_1, \ldots, s_m) \to s,$$

Определение

Кортеж термов $\langle t_1, \ldots, t_n \rangle$ принимается автоматом $\mathcal{A}$ тогда и только тогда, когда, $\langle \mathcal{A}[t_1], \ldots, \mathcal{A}[t_n] \rangle \in Q_F$, где

$$\mathcal{A}\big[F(t_1, \ldots, t_m)\big] = \begin{cases} s, & \text{если } \big(F(A[t_1], \ldots, A[t_m]) \to s\big) \in \Delta, \\ \bot, & \text{иначе.} \end{cases}$$

Пример

$$x = Z \wedge y = S(y') \to It(x, y)$$
$$x = S(x') \wedge y = S(y') \wedge It(x', y') \to It(x, y)$$
$$It(x, y) \wedge It(y, x) \to \bot$$

Пример

$$x = y \to Eq(x, y)$$

[4] Comon H. et al. Tree automata techniques and applications. – 2008.

Регулярные инварианты

Пример

$$
\begin{aligned}
x = Z \land y = S(y') &\rightarrow It(x, y) \\
x = S(x') \land y = S(y') \land It(x', y') &\rightarrow It(x, y) \\
It(x, y) \land It(y, x) &\rightarrow \bot
\end{aligned}
$$

Пример

$$x = y \rightarrow Eq(x, y)$$

Оценка выразительной силы

- Представление как декартового произведения унарных языков

[4] Comon H. et al. Tree automata techniques and applications. – 2008.

Регулярные инварианты

Пример

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

Пример

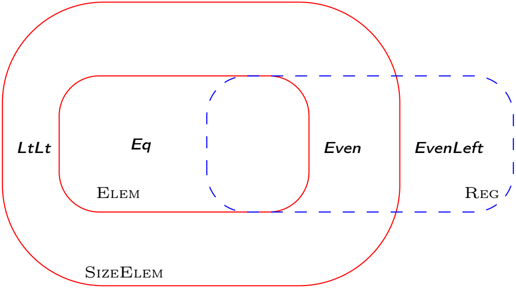
$$x = y \rightarrow Eq(x, y)$$

Оценка выразительной силы

- Представление как декартового произведения унарных языков
- Лемма о накачке[4]

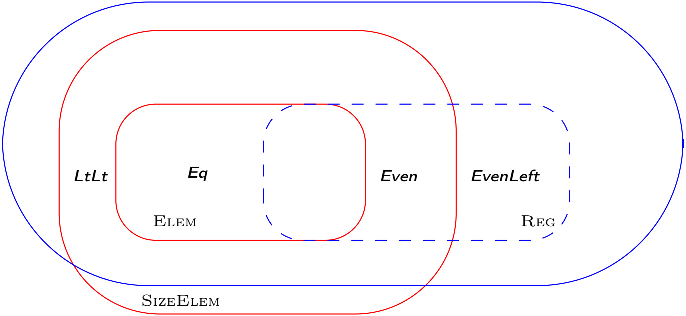
[4] Comon H. et al. Tree automata techniques and applications. – 2008.

Выразительная сила представлений



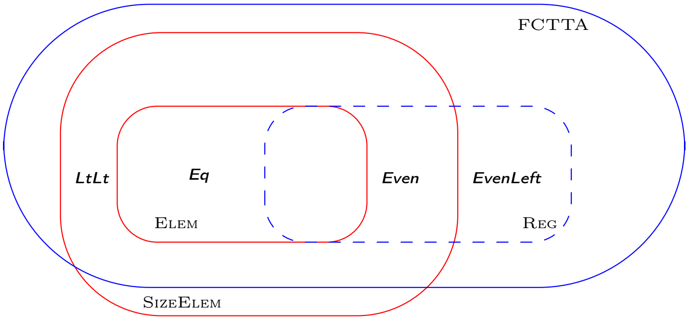
Сравнение выразительной силы различных абстрактных доменов
`Источник: Ю. О. Костюков, Автоматический вывод регулярных инвариантов программ с алгебраическими типами данных`

Выразительная сила представлений



Сравнение выразительной силы различных абстрактных доменов

Выразительная сила представлений



Сравнение выразительной силы различных абстрактных доменов

Синхронизированные древовидные автоматы

Определение

Синхронизированным древовидным автоматом называют автомат над новым алфавитом $(\Sigma_F \cup \{\perp\})^n$

Стратегия синхронизации

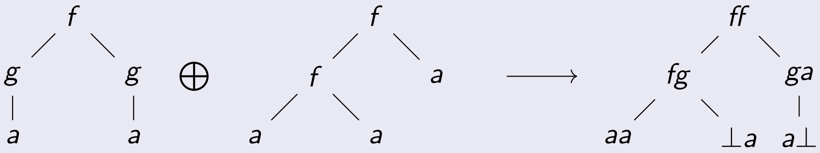
Стратегия синхронизации — это инъективная функция из множества кортежей слов в исходном алфавите Σ_F в слова в алфавите $(\Sigma_F \cup \{\perp\})^n$

Синхронизированные древесные автоматы

Стандартная стратегия

$$f(s_1, \dots, s_n) \oplus g(t_1, \dots, t_m) = fg(s_1 \oplus t_1, \dots, s_N \oplus t_N)$$

Пример



Пример

$$
\begin{array}{c} S \\ | \\ S \\ | \\ Z \end{array}
\quad \oplus \quad
\begin{array}{ccccc}
 & & cons & & \\
 & / & & \backslash & \\
Z & & & & cons \\
 & & & / & & \backslash \\
 & & Z & & & & nil
\end{array}
\quad \longrightarrow \quad
\begin{array}{ccccccc}
 & & & Scons & & & \\
 & & / & & \backslash & & \\
 & SZ & & & & \perp cons & \\
 / & & & & / & & \backslash \\
Z\perp & & & \perp nil & & & \perp Z
\end{array}
$$

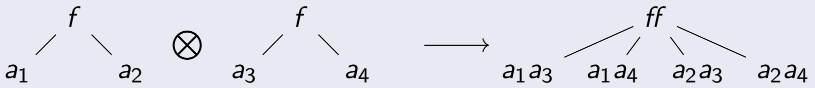
$$
\begin{array}{c} S \\ | \\ S \\ | \\ Z \end{array}
\quad \oplus \quad
\begin{array}{ccccc}
 & & & snoc & \\
 & & / & & \backslash \\
 & snoc & & & Z \\
 / & & \backslash & & \\
nil & & & Z &
\end{array}
\quad \longrightarrow \quad
\begin{array}{ccccc}
 & & & Ssnoc & \\
 & & / & & \backslash \\
 & Ssnoc & & & \perp Z \\
 / & & \backslash & & \\
\perp Z & & & Znill &
\end{array}
$$

Синхронизированные древесные автоматы

Полная стратегия

$$f(s_1, \ldots, s_n) \otimes g(t_1, \ldots, t_m) = fg(s_1 \otimes t_1, \ldots, s_1 \otimes t_m, \ldots, s_n \otimes t_m)$$

Пример



Синхронизированные древесные автоматы

Оценка выразительной силы

- Регулярные отношения

Синхронизированные древесные автоматы

Оценка выразительной силы

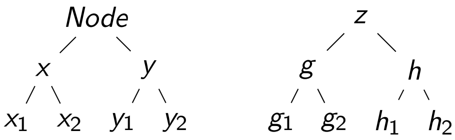
- Регулярные отношения
- Отношение размера
    - Длина списка

Оценка выразительной силы

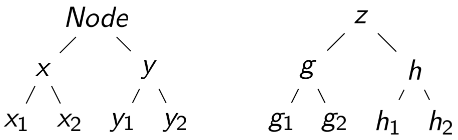
- Регулярные отношения
- Отношение размера
    - Длина списка
- Отношение равенства

Синхронный автомат для равенства

$z = Node(x, y)$



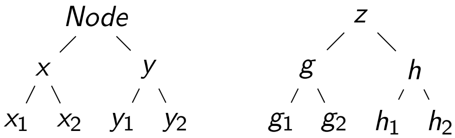
$z = Node(x, y)$



$$\mathcal{A}'[\langle x, y, z\rangle] \qquad \begin{array}{lll} \langle z, & \mathcal{A}[\langle x, g\rangle] & \mathcal{A}[\langle x, h\rangle] \\ & \mathcal{A}[\langle y, g\rangle] & \mathcal{A}[\langle y, h\rangle]\rangle \end{array}$$

Синхронный автомат для равенства

$z = Node(x, y)$



$$\mathcal{A}'[\langle x, y, z\rangle] \qquad \begin{aligned} &\langle z, \\ &\qquad delta_{\mathcal{A}}(x, g, x_1 g_1, x_1 g_2, x_2 g_1, x_2 g_2), \\ &\qquad delta_{\mathcal{A}}(x, h, x_1 h_1, x_1 h_2, x_2 h_1, x_2 h_2), \\ &\qquad delta_{\mathcal{A}}(y, g, y_1 g_1, y_1 g_2, y_2 g_1, y_2 g_2), \\ &\qquad delta_{\mathcal{A}}(y, h, y_1 h_1, y_1 h_2, y_2 h_1, y_2 h_2)\rangle. \end{aligned}$$

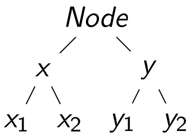
Синхронный автомат для равенства

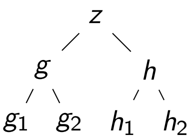
$z = Node(x, y)$

$$
\begin{array}{ccccccc}
 & & Node & & \\
 & / & & \backslash & \\
 & x & & & y \\
/ & & \backslash & / & & \backslash \\
x_1 & & x_2 & y_1 & & y_2
\end{array}
\qquad\qquad
\begin{array}{ccccccc}
 & & z & & \\
 & / & & \backslash & \\
 & g & & & h \\
/ & & \backslash & / & & \backslash \\
g_1 & & g_2 & h_1 & & h_2
\end{array}
$$

$$
\begin{aligned}
delta_{\mathcal{A}'}(x, y, z, \qquad & \langle z, \\
x_1y_1z_1, x_1y_1z_2, \qquad & delta_{\mathcal{A}}(x, g, x_1g_1, x_1g_2, x_2g_1, x_2g_2), \\
x_1y_2z_1, x_1y_2z_2, \qquad & delta_{\mathcal{A}}(x, h, x_1h_1, x_1h_2, x_2h_1, x_2h_2), \\
x_2y_1z_1, x_2y_1z_2, \qquad & delta_{\mathcal{A}}(y, g, y_1g_1, y_1g_2, y_2g_1, y_2g_2), \\
x_2y_2z_1, x_2y_2z_2). \qquad & delta_{\mathcal{A}}(y, h, y_1h_1, y_1h_2, y_2h_1, y_2h_2)\rangle.
\end{aligned}
$$

$z = Node(x, y)$





$$
\begin{aligned}
delta_{\mathcal{A}'}(x, y, z, \quad & \quad \langle z, \\
x_1y_1z_1, x_1y_1z_2, \quad & \quad delta_{\mathcal{A}}(x, g, x_1g_1, x_1g_2, x_2g_1, x_2g_2), \\
x_1y_2z_1, x_1y_2z_2, \quad & \quad delta_{\mathcal{A}}(x, h, x_1h_1, x_1h_2, x_2h_1, x_2h_2), \\
x_2y_1z_1, x_2y_1z_2, \quad & \quad delta_{\mathcal{A}}(y, g, y_1g_1, y_1g_2, y_2g_1, y_2g_2), \\
x_2y_2z_1, x_2y_2z_2). \quad & \quad delta_{\mathcal{A}}(y, h, y_1h_1, y_1h_2, y_2h_1, y_2h_2)\rangle.
\end{aligned}
$$

$x_1y_1z_1 \rightarrow x_1y_1g \rightarrow \langle g, x_1g_1, x_1g_2, y_1g_1, y_1g_2\rangle$

Автоматический вывод

Условия верификации

В общем виде

$$\varphi \wedge R_1(\overline{x}_1, y_1) \wedge R_2(\overline{x}_2, y_2) \wedge \ldots \wedge R_m(\overline{x}_m, y_m) \to R_0(\overline{x}_0, y_0)$$

R_i — неинтерпретированные предикатные символы, вызовы закодированных предикатными символами функций

x_i — входные значения

y_i — результаты

φ — ограничение в логике первого порядка, оставшееся тело функции

Ограничение

$\varphi ::= \ (t = t') \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \forall x.\varphi$

Языковая семантика

Определение

Языковую семантику бескванторной формулы определим индуктивно

- $L(p) \subseteq T(\Sigma)^k$
- $L[\![\neg\varphi]\!] = T(\Sigma)^n \setminus L[\![\varphi]\!]$
- $L[\![\varphi_1 \wedge \varphi_2]\!] = L[\![\varphi_1]\!] \cap L[\![\varphi_2]\!]$
- $L[\![\varphi_1 \vee \varphi_2]\!] = L[\![\varphi_1]\!] \cup L[\![\varphi_2]\!]$
- $L[\![p(\overline{t}(x_1, \dots, x_n))]\!] = \{u \in T(\Sigma)^n \mid t[u] \in L[\![p]\!]\}$

Определение

$L \models \varphi \iff L[\![\neg\varphi]\!] = \varnothing$

Теорема

Бескванторная формула φ выполнима в языковой семантике тогда и только тогда, когда она выполнима в семантике Тарского

Синхронные древовидные автоматы

Определение

Языковую семантику бескванторной формулы определим индуктивно

- $L(p) \subseteq T(\Sigma)^k$
- $L[\![\varphi_1 \wedge \varphi_2]\!] = L[\![\varphi_1]\!] \cap L[\![\varphi_2]\!]$
- $L[\![\neg\varphi]\!] = T(\Sigma)^n \setminus L[\![\varphi]\!]$
- $L[\![\varphi_1 \vee \varphi_2]\!] = L[\![\varphi_1]\!] \cup L[\![\varphi_2]\!]$
- $L[\![p(\overline{t}(x_1, \ldots, x_n))]\!] = \{u \in T(\Sigma)^n \mid t[u] \in L[\![p]\!]\}$

Определение

Языковую семантику бескванторной формулы определим индуктивно

- $L(p) \subseteq T(\Sigma)^k$
- $L[\![\varphi_1 \wedge \varphi_2]\!] = L[\![\varphi_1]\!] \cap L[\![\varphi_2]\!]$
- $L[\![\neg\varphi]\!] = T(\Sigma)^n \setminus L[\![\varphi]\!]$
- $L[\![\varphi_1 \vee \varphi_2]\!] = L[\![\varphi_1]\!] \cup L[\![\varphi_2]\!]$
- $L[\![p(\overline{t}(x_1, \dots, x_n))]\!] = \{u \in T(\Sigma)^n \mid t[u] \in L[\![p]\!]\}$

Нижний остаток (downward quotient)

$$t_1(x_{11}, \dots, x_{1N_1}), \dots, t_K(x_{K1}, \dots x_{KN_K}) \in L[\![p]\!] \iff$$
$$u_{11}, \dots, u_{1N_1}, \dots, u_{K1}, \dots u_{KN_K} \in L[\![p(\overline{t}(x_{11}, \dots, x_{KN_K}))]\!]$$

Синхронные древовидные автоматы

Определение

Языковую семантику бескванторной формулы определим индуктивно

- $L(p) \subseteq T(\Sigma)^k$
- $L[\![\varphi_1 \wedge \varphi_2]\!] = L[\![\varphi_1]\!] \cap L[\![\varphi_2]\!]$
- $L[\![\neg\varphi]\!] = T(\Sigma)^n \setminus L[\![\varphi]\!]$
- $L[\![\varphi_1 \vee \varphi_2]\!] = L[\![\varphi_1]\!] \cup L[\![\varphi_2]\!]$
- $L[\![p(\overline{t}(x_1, \ldots, x_n))]\!] = \{u \in T(\Sigma)^n \mid t[u] \in L[\![p]\!]\}$

Теорема

Класс языков, представимых синхронными древовидными автоматами *с полной сверткой* замкнут относительно операции взятия нижнего остатка

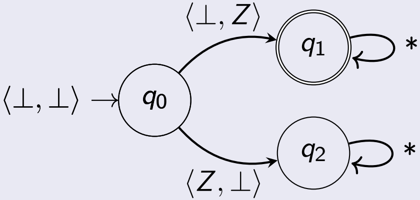
Синхронные древовидные автоматы

Логическая программа

$$x = Z \land y = S(y') \to lt(x, y)$$
$$x = S(x') \land y = S(y') \land lt(x', y') \to lt(x, y)$$
$$lt(x, y) \land lt(y, x) \to \bot$$

Инвариант

$$\mathcal{A} = \{Q, Init, Q_F, \delta\}$$
$$Q = \{q_0, q_1, q_2\}$$
$$Init = q_0$$
$$Q_F = \{q_1\}$$
$$\mathcal{L}(\mathcal{A}) = \{\langle Z, S(Z)\rangle, \langle Z, S(S(Z))\rangle, \ldots\}$$



Вывод инвариантов

RInGen

Условия верификации над АТД

Условия верификации над EUF

Логическое представление автоматов

$$
\begin{aligned}
S(x) &= y \\
Q_{\mathcal{B}} &= \{Z, S\} \times Q \\
F_{\mathcal{B}} &= \{\langle S, q\rangle \mid q \in F_{=}\} \\
\delta_{\mathcal{B}}(f, g, \langle g_1, q\rangle) &= \langle g, \delta_{=}(f, g_1, q)\rangle
\end{aligned}
$$

Конечная модель

Инвариант исходной программы

Эксперименты

SAT — number of derived invariants
UNSAT — number of derived counterexamples
∃! — number of unique results

Data set	#	Result	Spacer	Eldarica	RInGen	RInGen-TTA
TIP	454	SAT	26	46	25	43
		∃! SAT	7	14	0	4
		UNSAT	22	12	21	21
		∃! UNSAT	7	0	0	0

Результаты экспериментов на весну 2022 г.

Коллаборативным инвариантом[5] называется формула вида

$$\varphi(\overline{x}) \vee \overline{x} \in L,$$

где φ — формула первого порядка, а L некоторый формальный язык

Data set	#	Result	Spacer	RInGen	Collaborative
TIP	454	SAT	20	135	189
		UNSAT	15	46	28

Результаты экспериментов на лето 2023 г.

[5] Kostyukov Y., Mordvinov D., Fedyukovich G. Collaborative Inference of Combined Invariants //Proceedings of 24th International Conference on Logic. – 2023. – Т. 94. – С. 288-305.