Хорошо структурированные системы помеченных переходов, их отношение к моделям интуиционистких модальных логик и к суперкомпиляции

(доклад для Совещания по языку Рефал)

Николай Вячеславович Шилов

Университет Иннополис

17 июня 2023

План выступдения

- Сначала формально определим хорошо структурированные помеченные системы переходов,
- затем "вспомним" некоторые результаты 2004 г. (полученные совместно с Е.В. Кузменым и В.А. Соколовым) по вериикации дизъюнктивных формул пропозиционального μ-Исчисления Дехтера Козена в таких системах при условии интуиционисткой интерпретации пропозициональных переменных,
- а в конце рассмотрим связь хорошо структурированные системы переходов с так называемой суперкомпиляцией (в частности, методом обнаружения зацикливания программ, предложенным Валентином Федоровичем Турчиным в конце 1960-ых готов).

Все определения ниже преполагают, что мы выбрали и зафиксировали некоторое множество D.

- Предпорядок (или квазипорядок) — это рефлксивное и транзитивное бинарное отношение.
- Если предаорядок обладает свойством антисимметричности, то это оношение является (частичным) порядком.
- Хороший предпорядок — это такой предпорядок $\preceq$, что в любой бесконечной последовательности $d_0, \ldots d_i, \ldots$ in D найдется упорядоченная пара (такая что $m < n$ и $d_m \preceq d_n$).

- Конечномерные (любой фиксированной размерности) вектора из натуральных чисел (при покомпонентном сравнении).
- Теорема Краскала (Joseph Kruskal, 1960): Множество конечных деревьев над хорошо квазиупорядоченным набором меток является хорошо квазиупорядоченным (при гомеоморфном вложении).

- Идеал (или (верхний) конус) — это замкнутое вверх подмножество, то есть такое множество $I \subseteq D$ что для всех $d', d'' \in D$, если $d' \preceq d''$ и $d' \in I$ то $d'' \in I$.
- С любым элементом $d \in I$ естественным образом связан идеал $(\uparrow d) \equiv \{e \in D \ : \ d \preceq e\}$.
- Для любого множества $S \subseteq D$, его элемент $d \in S$ называется минимальным в S, если для любого $s \in S$ или $d \preceq s$, или d и s несравнимые элементы.
- Для любого множества $S \subseteq D$, множество всех его минимальных элементов обозначается $min(S)$; базис множества S — это любое подмножество $B \subseteq S$ такое, что для любого $s \in S$ существует $b \in B$ для которого $b \preceq s$.

- Хорошо предупорядоченное множество обязательно фундировано (не содержит бесконечных убывающих цепей), но обратное — неверно.
- Более того, в хорошо предупорядоченном множестве любая бесконечная последовательность содержит бесконечную "возрастающую" подпоследовательность.
- Любое множество S хорошо предупорядоченного множества имеет конечный базис, который состоит из всех минимальных элеменов этого множества $min(S)$.

- Любой идеал I хорошо предупорядоченного множества имеет конечный базис $min(I)$ и $I = \bigcup_{d \in min(I)} (\uparrow d)$.
- В хорошо предупордоченном множестве любая неубывающая (по включению) последовательность идеалов $I_0 \subseteq \cdots \subseteq I_i \subseteq \ldots$ стабилизируется: существует такое $k \geq 0$ что $I_m = I_n$ для всех $m, n \geq k$.

Пусть Act — произвольный фиксированный конечный алфавит символов действий.

- Система переходов (или фрейм (Крипке)) — это пара (D, R), в которой носитель (или домен) D — не пустое множество состояний (или миров), аинтерпретация R — это отображение $R : Act \to 2^{D \times D}$ символов действий бинарными отношениями.
- Вычисление (в системе переходов (D, R)) — это максимальная (по включению) последовательность состояний $s_0 \dots s_i s_{i+1} \dots$ такая, что для любой пары последовательных состояний из этой последовательности $(s_i, s_{i+1}) \in R(a)$ для некоторого $a \in Act$.

- Хорошо предупорядоченная система переходов (well-preordered transition system, WPTS) — это тройка $(D, \preceq, R)$ такая, что
  - $(D, \preceq)$ — хорошо перупорядоченное множество,
  - а (D, R) — система переходов.

Существует 4 варианта (сильного) согласования предпорядка $\preceq$ с интерпретацией $R(a)$ каждого из символов действий $a \in Act$ (см. рисунок на слдующем слайде 11).

- Прилагательное future означает, что нас интересуют состояния после выполнения действия $R(a)$.
- Прилагательное past означает, что нас интересуют состояния до выполнения действия $R(a)$.
- Прилагательное upward означает, что нас интересуют состояния "в порядке возрастания" $\preceq$.
- Прилагательное future означает, что нас интересуют состояния "в порядке убывания" $\preceq$.

Варианты согласования предпорядка и переходов

Future Upward – FU

- Logic: $\forall s_1', s_1'', s_2' \exists s_2''$: $(s_1', s_1'') \in R(a) \,\&\, s_1' \preccurlyeq s_2' \Longrightarrow$
  $\Longrightarrow (s_2', s_2'') \in R(a) \,\&\, s_1'' \preccurlyeq s_2''$
- Algebraic: $(\preccurlyeq)^- \circ R(a) \subseteq R(a) \circ (\preccurlyeq)^-$

$$\begin{tikzcd} s_1'' \arrow[r, phantom, "\boxed{\preccurlyeq}"] & s_2'' \\ s_1' \arrow[u, "a"] \arrow[r, phantom, "\preccurlyeq"] & s_2' \arrow[u, dashed, "a"'] \end{tikzcd}$$

Future Downward – FD

- Logic: $\forall s_1', s_2', s_2'' \exists s_1''$: $(s_2', s_2'') \in R(a) \,\&\, s_1' \preccurlyeq s_2' \Longrightarrow$
  $\Longrightarrow (s_1', s_1'') \in R(a) \,\&\, s_1'' \preccurlyeq s_2''$
- Algebraic: $(\preccurlyeq) \circ R(a) \subseteq R(a) \circ (\preccurlyeq)$

$$\begin{tikzcd} s_1'' \arrow[r, phantom, "\boxed{\preccurlyeq}"] & s_2'' \\ s_1' \arrow[u, dashed, "a"] \arrow[r, phantom, "\preccurlyeq"] & s_2' \arrow[u, "a"'] \end{tikzcd}$$

Past Upward – PU

- Logic: $\forall s_1', s_1'', s_2'' \exists s_2'$: $(s_1', s_1'') \in R(a) \,\&\, s_1'' \preccurlyeq s_2'' \Longrightarrow$
  $\Longrightarrow (s_2', s_2'') \in R(a) \,\&\, s_1' \preccurlyeq s_2'$
- Algebraic: $R(a) \circ (\preccurlyeq) \subseteq (\preccurlyeq) \circ R(a)$

$$\begin{tikzcd} s_1'' \arrow[r, phantom, "\preccurlyeq"] & s_2'' \\ s_1' \arrow[u, "a"] \arrow[r, phantom, "\boxed{\preccurlyeq}"] & s_2' \arrow[u, dashed, "a"'] \end{tikzcd}$$

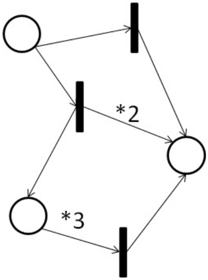
Past Downward – PD

- Logic: $\forall s_1'', s_2', s_2'' \exists s_1'$: $(s_2', s_2'') \in R(a) \,\&\, s_1'' \preccurlyeq s_2'' \Longrightarrow$
  $\Longrightarrow (s_1', s_1'') \in R(a) \,\&\, s_1' \preccurlyeq s_2'$
- Algebraic: $R(a) \circ (\preccurlyeq)^- \subseteq (\preccurlyeq)^- \circ R(a)$

$$\begin{tikzcd} s_1'' \arrow[r, phantom, "\preccurlyeq"] & s_2'' \\ s_1' \arrow[u, dashed, "a"] \arrow[r, phantom, "\boxed{\preccurlyeq}"] & s_2' \arrow[u, "a"'] \end{tikzcd}$$

- Хорошо структурированная система переходов (Well-Structured Transition System, WSTS) — это FU-согласованная хорошо предупорядоченная система переходов.
- Отношения подобия (simulation) и бисимуляции (bisimulation) могут быть определены в терминах согласованности предпорядка и переходов следующим образом:
  - Отношение подобия — это предпорядок FU-согласованный с системой переходов.
  - Отношение бисимуляции — это отношение эквивалентности FU-согласованное с системой переходов.

- Сеть Петри представляет собой двудольный ориентированный (мульти)граф, в котором узлы в одной части называются местами (изображаются кружками), а узлы в другой части называются переходами (изображаются в виде черты)...



"Классические" вопросы для WSTS

- Зацикливание: существует ли бесконечное вычисление, начинающееся с заданного состояния?
- Ограниченность: является ли множество состояний, достижимых из заданного состояния?
- Покрытие: для заданных состояний s и t, существует ли состояние, достижимое из s, которое "мажорирует"t?

- Разрешимость зацикливания и ограниченности:
    - Alain Finkel. Reduction and covering of infinite reachability trees. Information and Computation, 89(2):144-179, December 1990.
    - A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! Theoretical Computer Science, 256(1):63-92, 2001.
- Разрешимость покрытия:
    - Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. Algorithmic Analysis of Programs with Well Quasi-ordered Domains. Information and Computation, 160(1):109-127, 2000.

Теорема

The model checking problem is decidable for disjunctive formulas of the Propositional μ-Calculus in intuitionistic models over well-structured transition systems with decidable well-preorder tractable pasts.

Теорема доказана в работе

- E. V. Kouzmin, N. V. Shilov, V. A. Sokolov. Model checking mu-calculus in well-structured transition systems. In: Proc. of 11th International Symposium on Temporal Representation and Reasoning, 2004 (TIME 2004), 152-155, 2004. (Extended version — in Bulletine of Novosibirsk Computing Center. Computer Science. 20, 49-59, 2004.)

- Задача верификации моделей (Model Checking) — это алгоритмическая проблема вычисления семантики заданной формулы (заданной логики) в заданной модели (этой логики).
- Требование decidability for the well-pre-order означает что предпорядок $\preceq$ имеет рекурсивный график (эффективно проверяется для любой пары состояний).
- Требование tractable past означает что для любого символа действия $a \in Act$ функция $\lambda s \in D \ : \ \min\{t \in D \ : \ (t, s) \in R(a)\}$ эффективно вычислима.

Пусть Prp — фиксированный конечный алфавит пропозициональных переменных.

- Помеченная система переходов (или модель Крипке) с фреймом $F = (D, R)$ (или хорошо предупорядоченной, или даже хорошо структурированной системой переходов $(D, \preceq, R)$) — это фрейм F дополненный интерпретацией (или оценкой) $I : Prp \to 2^D$ пропозициональных переменных множествами состояний.
- Будем называть интерпретацию пропозициональной переменной $p \in Prp$ интуиционисткой, если фрейм явлется хорошей предупорядоченной системой переходов и $I(p)$ является идеалом.
- Будем называть модель интуиционисткой, если она имеет интуиционисткую интерпретацию всех пропозициональных переменных.

- В работе Model checking mu-calculus in well-structured transition systems (2004) условия совместимости называются условиями Фишер Серви из-за интуиционистской модальной логики, предложенной Г. Фишер Серви (G. Fisher Servi):
  - F. Wolter, M. Zakharyaschev. Intuitionistic Modal Logic. In: Logic and Foundations of Mathematics. Synthese Library, vol 280. Springer, Dordrecht, 1999.
- Семантика логики Фишер Серви определяется в частично упорядоченных системах переходов $(D, \preceq, R)$, где $\preceq$ — бисимуляция для всех $R(a)$, $a \in Act$.

- Однако, из доклада С.П. Одинцева "Модальная логика Мойсила и родственные системы" на семинаре "Нестандартные логики" (27 марта 2023 г.) следует, что похожие условия совместимости были введены и изучены не только Г. Фишер Серви, но и другими исследователями примерно в тож время 1982-1986 гг.
- В работах по WSTS вариации условий согласованнсти называются условиями монотонности, например:
    - Bollig, B., Finkel, A., Suresh, A. (2022). Branch-Well-Structured Transition Systems and Extensions. In: FORTE 2022. Lecture Notes in Computer Science, vol 13273, 50-66, 2022.

Синтаксис пропозиционального μ-Исчисления (μC) Д. Козена (1983) состоит из формул:

$$\phi \ ::= \ \underline{p} \mid (\neg\phi) \mid (\phi \wedge \phi) \mid \underline{(\phi \vee \phi)} \mid ([a]\phi) \mid \underline{(\langle a\rangle\phi)} \mid (\nu p.\ \phi) \mid \underline{(\mu p.\ \phi)}$$

В этом определении

- метапеременные p и a "пробегают" алфавиты Prp и Act пропозициональных переменных и символов действий,
- а метапеременная ϕ "определяет" множество формул.
- Имеются контестные ограничения: все вхождения связанных μ или ν переменных являются "позитивными" (находятся в области четного числа отрицаний).
- В определении дизъюнктивных формул используются разрешены только консрукции, которые выделены подчеркиванием.

В модели $M = (D, R, I)$ семантика $M(\phi)$ любой формулы ϕ — это множество состояний (домена D).

- $M(p) = I(p)$, $M(\neg\phi) = D \setminus M(\phi)$,
  $M(\psi \wedge \theta) = M(\psi) \cap M(\theta)$, а $M(\psi \vee \theta) = M(\psi) \cup M(\theta)$
- $M([a]\psi) = \{s$: для каждого $t \in D$
  если$(s, t) \in R(a)$ то $t \in M(\psi)\}$
- $M(\langle a\rangle\psi) = \{s$: для некоторого $t \in D$
  имеет место $(s, t) \in R(a)$ и $t \in M(\psi)\}$

- Пусть модель $M_{S/p}$ отличается модели M только интерпретацией пропозициональной переменной p: $M_{S/p}(p) = I_{S/p}(p) = S$.
  - $M(\nu p.\ psi)$ = наибольшая (по включению) неподвижная точка $\lambda S \subseteq D\ .\ \left(M_{S/p}(\psi)\right)$
  - $M(\mu p.\ psi)$ = наименьшая (по включению) неподвижная точка$\lambda S \subseteq D\ .\ \left(M_{S/p}(\psi)\right)$

- Обозначим предикат зацикливания (недетерминированной структурированной) программы α в состоянии $\mathit{loop}(\alpha)$.
- Имеем:
  - $\mathit{loop}(a) \equiv \mathit{loop}(\phi?) \equiv \mathit{false}$ для любого символа действия и формулы
  - $\mathit{loop}(\beta;\gamma) \equiv (\mathit{loop}(\beta) \vee \langle\beta\rangle \mathit{loop}(\gamma))$
  - $\mathit{loop}(\beta \cup \gamma) \equiv (\mathit{loop}(\beta) \vee \mathit{loop}(\gamma))$
  - $\mathit{loop}(\beta^*) \equiv (\langle\beta^*\rangle \mathit{loop}(\beta) \vee \Delta(\beta))$
  - $\Delta(\beta) \equiv \nu p.\ (\langle\beta\rangle p)$

- Суперкомпиляция — это метод обнаружения зацикливания программ, предложенный В.Ф. Турчиным в конце 1960-ых годов.
- В его основе лежит или бисимуляция, или обобщение вычислений (см. далее).
- V. F. Turchin. The concept of a supercompiler. ACM Trans. Program. Lang. Syst. 8, 3, 292-325, 1986.
- А.В. Климов. О работах Валентина Федоровича Турчина по кибернетике и информатике. Материалы международной конференции SORUCOM 2011 (12–16 сентября 2011 г.). Доступна на https://www.computer-museum.ru/histussr/turchin_sorucom_2011.htm.

- Начиная с этого места мы фиксируем модель, в которой происходят вычисления (D, R, I).
- Поэтому мы будем использовать
  - $s \xrightarrow{a} t$ вместо $(s, t) \in R(a)$,
  - и $p(s)$ — вместо $s \in I(p)$.

Каждая программа α — эток конечное множество помеченных операторов:

- присваиваний вида $l: \ a \ goto \ J$, где $l \in \mathbb{N}$ — метка (натуральное число), $a \in Act$ — действие (тело оператора), а $J \subseteq \mathbb{N}$ — конечное множество меток (возможно пустое)
- условного переходв (выбора) вида $l: \ if \ p \ then \ J \ else \ K$, где $l \in \mathbb{N}$ — метка (натуральное число), $p \in Prp$ — условие (выраженное логической формулой), а $J, K \subseteq \mathbb{N}$ — конечные множества меток (каждое, возможно, пустое).

- Стартовая метка (программы α) — это наименьшая метка, которая метит хотябы один оператор программы.
- Выходная метка ("выход" программы α) — это произвольная метка, которая встречается в программе, но н метит ниодного оператора в ней.

- Конфигурация (программы α) — это произвольная пара (l, s), в которой l — это метка, а s — состояние.
- Срабатывание (программы α) — это проивольное срабатывание какого-либо из ее операторов (см. следующий слайд).

- Срабатывание присваивания $(I : a \ goto \ J) \in \alpha$ — это произвольная пара конфигураци $(I, s)(j, t)$ такая, что $s \xrightarrow{a} t$ и $j \in J$.
- Позитивное срабатывание условного перехода $(I : \ if \ p \ then \ J \ else \ K) \in \alpha$ — это произвольная пара конфигураций $(I, s)(j, s)$ такая, что $p(s)$ и $j \in J$.
- Негативное срабатывание условного перехода $(I : \ if \ p \ then \ J \ else \ K) \in \alpha$ — это произвольная пара конфигураций $(I, s)(k, s)$ такая, что $p(s)$ ложно и $k \in K$.
- Срабатывание условного перехода $(I : \ if \ p \ then \ J \ else \ K) \in \alpha$ — это любое его позитивное или негативное срабатывание.

Вычисление (программы α) — это произвольная (конечная или бесконечная) последовательность конфигураций $(l_0, s_0) \ldots (l_n, s_n)(l_{n+1}, s_{n+1}) \ldots$ такая, что любая пар последовательных конфигураций $(l_n, s_n)(l_{n+1}, s_{n+1}$ в этой последовательности — срабатывание α.

- Начальное вычисление — это вычисление, начинающееся с начальной метки.
- Завершенное вычисление — это вычисление, заканчивающееся выходной меткой.
- Зацикливание — это любое бесконечное вычисление.
- Полное вычисление — это начальное завершенное вычисление или зацикливание.

Распространим естественным образом бисимуляцию с фреймов на модели: для любых состояний таких что $s \simeq t$ дополнительно потребуем $p(s) \Leftrightarrow p(t)$ (для каждой пропозициональной переменной).

Лемма

Для любых программы α, метки l, сщстояний s, t и бисимуляции в модели $\simeq$,
если $s \simeq t$ и α зацикливается из (l, s),
то α зацикливается и из (l, t).

- Обобщение — это произвольное бинарное отношение $\overset{gen}{\Longrightarrow}$ на домене D.
- Если $\overset{gen}{\Longrightarrow}$ — обобщение, то для любых меток l, k и состояний s, t будем писать $(l, s) \overset{gen}{\Longrightarrow} (k, t)$, если $l = k$ и $s \overset{gen}{\Longrightarrow} t$.
- Говорят, что обобщение $\overset{gen}{\Longrightarrow}$ корректно (для нашей программы α), если для любых состояний $s, t, u \in D$ и меок l, k, возможность $s \overset{gen}{\Longrightarrow} t$ и (обобщенного или спекулятивного) срабатывания $(l, t)(k, u)$ влечет существование состояния $v \in D$ такого, что $v \overset{gen}{\Longrightarrow} u$, а $(l, s)(k, v)$ — тоже срабатывание (реальное срабатывание).
- Очевидно, что корректность обобщения — это FD-согласованность, определенная на слайде 11.

Лемма
Для любой программы α, любого корректного обобщения $\stackrel{gen}{\Longrightarrow}$, любых метки l и состояния $s \in D$, для любого "спекулятивного" состояния $t \in D$ обобщающего s (то есть такого, что $s \stackrel{gen}{\Longrightarrow} t$),
если у α существует "спекулятивное" вычисление ρ начинающееся и заканчивающееся (l, t),

тогда α имеет зацикливание начиная с (l, s),
которое обобщается до вычисления ρ^ω
(до бесконечного повторения ρ).

Суперкомпиляция как метод обнаружения зацикливаний программы основан на предыдущей лемме (слайд 34).

Предусловие: α — программа, (k, r) — ее конфигурация, а Gen — множество корректных обобщений.

Инвариант: Для любой конфигурации (l, s) достижимой из (k, r), если эта конигурация (k, s) помечена *loop*, то тогда программа α имеет бесконечное вычисление ("зацикливается"), начинающееся в (l, s).

Начиная с конфигурации (k, r) выполняем обход в ширину дерева вычислений программы α.

- Как ктолько на каком-либо вычислении из (k, r) встретились две реальные конфигурации (l, s) и (l, v),
    - которые обобщаются (каким-либо одним отношением $\overset{gen}{\Longrightarrow}$ из Gen) до одной и той же спекулятивной конфигурации (l, t), $(l, s) \overset{gen}{\Longrightarrow} (l, t)$ и $(l, v) \overset{gen}{\Longrightarrow} (l, t)$,
    - для которой существует непустое спекулятивное вычисление, начинающееся и заканчивающееся в (l, t),

тогда обе реальные конфигурации (l, s) и (l, v) помечаются $loop$.

- В феврале-марте 2023 г. на семинаре "Нестандартные логики" (Институт математики СО РАН, Новосибирск) выступил Е.В. Борисов с циклом докладов "Негибридные логики для кросс-мировой предикации".
- В терминах этого доклада, по-видимому, можно сказать, что при суперкомпиляции мы рассуждаем в стиле кросс-мировой предикации (de dicto?): мы пытаемся выполнить спекулятивные вычисления чтобы сделать заключение о реальных вычислениях.

Спасибо за внимание!
Есть вопросы?