Вычисления высших порядков и верификация моделей программ

Пеленицын А. М.
ulysses4ever@gmail.com

Кафедра алгебры и дискретной математики
Факультет математики, механики и компьютерных наук
Южный федеральный университет

19 октября 2009 г.

Содержание

Общая информация о школе

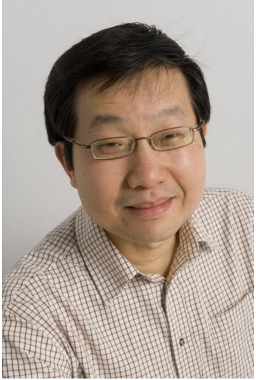
- проводится с 1970 года,
- официальное название:
  «Logics and Languages for Reliability and Security»,
- посвящена «формальным методам»

  > *a particular kind of mathematically-based techniques for the specification, development and verification of software and hardware systems*[1]

- подробности — на it.mmcs.sfedu.ru

[1] Wikipedia:Formal methods

Dr. Luke Ong



Основа сегодняшнего доклада — курс лекций Dr. Luke Ong[2]:

[2] Oxford University Computing Laboratory

История

- Подход к верификации, появившийся в начале 80-х [CE81].
- Изначально предназначался для систем с конечным числом состояний (аппаратное обеспечение, коммуникационные протоколы).

Кларк, Эмерсон и Сифакис награждены премией Тьюринга за 2007 год

«за их роль в превращении model checking в высоко эффективную технологию верификации, широко применяемую в индустрии аппаратного и программного обеспечения».

Разработки последних десяти лет направлены на распространение техники на верификацию ПО.

Основная трудность: комбинаторный взрыв в пространстве состояний.

Подход Model checking

Подход

Задача: пусть задана система *Sys* (например, ОС) и некоторое желаемое свойство *Prop* (например, отсутствие блокировок), удовлетворяет ли *Sys Spec*?

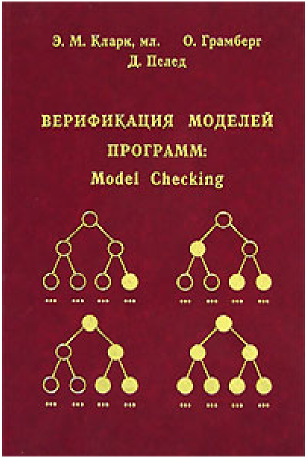
Решение:

1. Построить модель M для *Sys*.
2. Описать свойство *Prop* как формулу φ подходящего логического языка.
3. Направить имеющиеся вычислительные ресурсы на проверку выполнимости φ на M.

Результаты

- Далеко продвинулась верификация императивных программ (1-го порядка).
- Множество инструментов: SLAM, Blast, Terminator, SatAbs, и т. д.
- Основные используемые техники:
  1. абстракция (CEGAR: Counter-Example Guided Abstraction Refinement)
  2. повышение производительности (SAT- и SMT-solvers)

Продолжить знакомство. . .



Э. М. Кларк, О. Грамберг, Д. Пелед
Верификация моделей программ.
Model Checking / МЦНМО: 2002.
(MIT Press: 2001.)

Верификация вычислений высших порядков

- молодое направление
- некоторые теоретические результаты, мало инструментов
- сложность:
  1. бесконечное число состояний
  2. семантика моделей часто слишком абстрактна для алгоритмического анализа
     - типичный пример: денотационная семантика
     - важный контрпример: игровая семантика

Актуальность вычислений высших порядков

- обилие реализаций: Haskell, OCaml, F#, Lisp/Scheme, Ptalon, etc.,
- традиционные приложения: автоматическое доказательство теорем, вычислительная лингвистика, разбор языков программирования,
- новые приложения: базы данных и поиск информации (Google's MapReduce), сетевые взаимодействия[3].
- теоретические проблемы (анализ завершения программ, анализ достижимости).

[3] См. также новый журнал: «Практика функционального программирования», fprog.ru

Определение стекового автомата порядка n [Маслов 74]

Стековые автоматы порядка 2

1-стек это обычный стек, содержащий символы некоторого алфавита. 2-стек (соотв., n-стек) это стек 1-стеков (соотв., $n-1$-стеков).

Операции с 2-стеками: пусть s_k представляет 1-стек; вершина стека справа.

$$
\begin{array}{rcll}
\mathrm{push}_2: & [s_1 \dots s_{i-1} \overbrace{[a_1 \dots a_k]}^{s_i}] & \mapsto & [s_1 \dots s_{i-1} s_i s_i] \\
\mathrm{pop}_2: & [s_1 \dots s_{i-1} [a_1 \dots a_k]] & \mapsto & [s_1 \dots s_{i-1}] \\
\mathrm{push}_1\ a: & [s_1 \dots s_{i-1} [a_1 \dots a_k]] & \mapsto & [s_1 \dots s_{i-1} [a_1 \dots a_k\, a]] \\
\mathrm{pop}_1: & [s_1 \dots s_{i-1} [a_1 \dots a_k]] & \mapsto & [s_1 \dots [a_1 \dots a_{k-1}]]
\end{array}
$$

Аналогично определяются n-стеки. Стековый автомат порядка n имеет n-стек и push_i, pop_i для всех $1 \leqslant i \leqslant n$.

САВП как распознаватели

САВП могут использоваться для распознавания:

1. языков из (конечных) слов [Maslov74] (а также, и ω-слов),
2. возможно бесконечных деревьев [KNU01] и «языков из деревьев»,
3. возможно бесконечных графов [Courcelle95], [Cachat03].

Языки САВП (Маслов 74, 76)

1. САВП определяют бесконечную иерархию языков.
2. Малые порядки хорошо изучены: языки порядков 0, 1 и 2 это в точности регулярные, контекстно-свободные и индексированные [Aho68] языки. Языки высших порядков остаются плохо изученными.
3. Для каждого $n \geqslant 0$ языки порядка n образуют «абстрактное семейство языков» (замкнутое относительно $\cdot$, $+$, *, пересечения с регулярными языками и гомоморфизмов).
4. Для каждого $n \geqslant 0$ проблема непустоты языка порядка n разрешима.

Пример: $L = \{a^n b^n c^n \mid n \geqslant 0\}$ распознаётся 2-САВП

L не контекстно-свободный (см. лемму «о накачке»).

Идея: использовать верхний 1-стек для разбора $a^n b^n$ и высоту 2-стека для хранения n.

$$
\begin{tikzcd}
q_1[[]] \arrow[r, "a"] & q_1[[][z]] \arrow[r, "a"] & q_1[[][z][zz]] \arrow[d, "b"'] \\
 & & q_2[[][z][z]] \arrow[d, "b"'] \\
q_3[[]] & q_3[[][z]] \arrow[l, "c"'] & q_2[[][z][]] \arrow[l, "c"']
\end{tikzcd}
$$

Схемы рекурсии: пример

Пусть $\Sigma = \{f : 2, g : 1, a : 0\}$ — «взвешенный алфавит». Схема рекурсии G это «система уравнений»:

$$G : \left\{ \begin{array}{rcl} S & = & F\,a, \\ F\,x & = & f\,x\,(F\,(g\,x)). \end{array} \right.$$

Развёртка, начиная со стартового символа S:

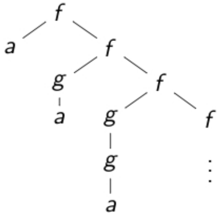
$$\begin{array}{rcl} S & \rightarrow & F\,a \\ & \rightarrow & f\,a\,(F\,(g\,a)) \\ & \rightarrow & f\,a\,(f\,(g\,a)\,(F\,(g\,(g\,a)))) \\ & \rightarrow & \ldots \end{array}$$

Дерево термов, генерируемое G:

$$[\![G]\!] = f\,a\,(f\,(g\,a)\,(f\,(g\,(g\,a))(\ldots))).$$

Представление $[\![G]\!]$

Дерево термов $[\![G]\!] = f\ a\,(f\ (g\ a)\,(f\ (g\ (g\ a))(\ldots)))$ в более привычном виде:



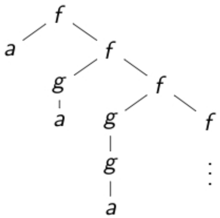
Попробуем дать формальные определения встреченных понятий.

Дерево термов

Взвешенный алфавит Σ это пара $(D(\Sigma), \mathrm{weight}\colon D(\Sigma) \to \mathbb{N})$, где $D(\Sigma)$ некоторое множество.
(Часто отождествляют Σ и $D(\Sigma)$).

Дерево термов t взвешенного алфавита Σ. Пусть

- $Dir\colon f \in \Sigma \mapsto \{1, \ldots, \mathrm{weight}(f)\}$;
- $Dir^* := (\bigcup_{f\in\Sigma} Dir(f))^*$
- $T \subset Dir^*$ — префикс-замкнутое подмножество Dir^*



Тогда $t\colon T \to \Sigma$ — дерево термов.

Простые типы

Типы $A ::= \circ \mid (A \to A)$

Каждый тип единственным образом записывается в виде

$$A_1 \to (A_2 \to \dots (A_n \to \circ) \dots), \quad n \geqslant 0, \tag{1}$$

который будем обозначать $A_1 \to A_2 \to \dots A_n \to \circ$.
n — арность данного типа.

Порядок типа измеряет вложенность левых частей $\to$:

$$\begin{aligned} \mathrm{order}(\circ) &:= 0, \\ \mathrm{order}(A \to B) &:= \max\left(\mathrm{order}(A) + 1, \mathrm{order}(B)\right). \end{aligned}$$

Примеры. $\mathbb{N} \to \mathbb{N}$ и $\mathbb{N} \to (\mathbb{N} \to \mathbb{N})$ имеют порядок 1.
$(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ имеет порядок 2.

Обозначение. «$e\colon A$» значит «e имеет тип A».

Определение схемы рекурсии $G = (\mathcal{N}, \Sigma, \mathcal{R}, S)$ порядка n

Пусть имеется неограниченный запас типизированных переменных $(\varphi,\ x,\ y,\ \ldots)$ Var.

- $\mathcal{N}$: множество типизированных «нетерминалов» порядка максимум n, включая выделенный стартовый символ $S : \circ$.
- Σ: типизированный алфавит «терминалов»
- $\mathcal{R}$: множество уравнений для каждого нетерминала $X : A_1 \to \ldots \to A_m \to \circ$ вида:

$$X\,\varphi_1\,\ldots\,\varphi_m = e,$$

  где $e : \circ$ это терм, сконструированный из
  - терминалов $f, g, \ldots \in \Sigma$,
  - переменных $\varphi_1, \varphi_2, \ldots, \varphi_m$,
  - нетерминалов $Y, Z, \ldots \in \mathcal{N}$

  по правилу аппликации: если $s : A \to B$ и $t : A$, определён терм $(s\,t) : B$.

Дерево терма

Для терма $t \in (\Sigma \cup \mathcal{N} \cup Var)^*$ определим «дерево $t^\perp$»:

$$t^\perp := \begin{cases} f & \text{если } t \text{ — терминал } f, \\ t_1^\perp t_2^\perp & \text{если } t = t_1 t_2 \text{ и } t_1^\perp \neq \perp, \\ \perp & \text{в остальных случаях.} \end{cases}$$

Это определение совпадает с определением дерева термов взвешенного алфавита Σ, данным ранее. (За функцию веса принимается функция арности символа.)

Дерево, порождённое G

Введём частичный порядок на $\Sigma \cup \{\perp\}$: $\forall a \in \Sigma\colon \perp \leqslant a$. Он, очевидно, распространяется на деревья. Например:

$$\perp \leqslant f \perp \perp \leqslant f \perp b \leqslant f\, b\, b.$$

Доказывается, что для любого множества деревьев T существует наименьшая верхняя граница $\bigsqcup T$, и корректно определено:

$$[\![G]\!] := \bigsqcup \{t^{\perp} \mid S \to^* t\}.$$

Связь САВП и схем рекурсии

Теорема

Для каждого $n \geqslant 0$ три формализма

1. *САВП порядка n,*
2. *сохранные схемы рекурсии порядка n [Damm82],*
3. *обобщённые индексированные грамматики порядка n [Маслов76]*

порождают один и тот же класс языков.

Почему МВП-теории?

- МВП-теории очень выразительны (более выразительны, чем темпоральные логики и μ-исчисление).
- МВП-теории сложно расширить каким-либо разумным способом.

Логическое представление деревьев термов

$t\colon T \to \Sigma$ можно представить так:

$$\langle T, \quad \langle d_i : 1 \leqslant i \leqslant m \rangle, \quad \langle p_f : f \in \Sigma \rangle \rangle,$$

где $d_i(x, y)$ это отношение «x является i-м ребёнком y»,
$p_f(x)$ это отношение «x имеет метку f».

Определение МВП-теории (для Σ-отмеченного дерева)

Язык теории состоит из:

- переменных первого порядка: $x, y, \ldots$ (пробегающих вершины деревьев);
- переменных второго порядка: $X, Y, \ldots$ (пробегающих множества вершин деревьев);
- атомарных формул вида: $d_i(x, y)$, $p_f(x)$, $x \in X$;
- формул, построенных из атомарных при помощи булевых связок, квантификации по переменным первого порядка, квантификации по переменным второго порядка.

Обзор результатов разрешимости МВП-теорий

- Рабин, 1969: регулярные деревья.
- Muller and Schupp 1985: конфигурационные графы МП-автоматов.
- Knapik, Niwińsky and Urzyczyn 2001:
  $\mathbf{PushdownTree}_n\Sigma$ — деревья, порождённые САВП порядка n;
  $\mathbf{SafeRecSchTree}_n\Sigma$ — деревья, порождённые сохранными схемами рекурсии порядка n;

[Aho68] *A. Aho.* Indexed grammars — an extension of context-free grammars. // J. ACM, 15:647–671, 1968.

[CE81] *E. M. Clarke, E. A. Emerson.* Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. // Logic of Programs — 1981 — Pp. 52-71.

[Cachat03] *T. Cachat.* Higher order pushdown automata, the Caucal hierarchy of graphs and parity games. // In Proc. ICALP'03, LNCS 2719, pp. 556-569, 2003.

[Courcelle95] *B. Courcelle.* The monadic second-order logic of graphs IX: machines and their behaviours. // TCS 151:125-162, 1995.

[Damm82] *W. Damm.* The IO- and OI-hierarchy. // TCS 20:95-207, 1982.

[KNU01] *T. Knapik, D. Niwiński, P. Urzyczyn.* Deciding monadic theories of hyperalgebraic trees. // In Proc. TLCA'01, LNCS 2044 — 2001 — Pp. 253-267.

Библиография II

[Maslov74] *A. N. Maslov.* The hierarchy of indexed languages of an arbitrary level. // Soviet Math. Dokl., 15. — 1974 — C. 1170–1174.

[Маслов76] *А. Н. Маслов.* Многоуровневые магазинные автоматы. // Пробл. передачи информ., 12:1 — 1976 — С. 55–62.