Математические методы верификации схем и программ

Лекторы:

Захаров Владимир Анатольевич

Подымов Владислав Васильевич

e-mail рассказчика:

valdus@yandex.ru

Осень 2017

Лекция 2

Общие принципы
дедуктивной верификации программ

Операционная семантика
императивных программ

Формальная постановка
задачи верификации программ

Логика Хоара: теория и практика

Автоматизация
проверки правильности программ

Принципы дедуктивной верификации программ

1. Программа π вычисляет отношение R_π между данными, подаваемыми на вход и получаемыми на выходе
2. Текст программы π — это формальное описание отношения R_π
3. Спецификация программы Φ — это формальное описание отношения R_Φ между данными программы
   - Отношение, описываемое спецификацией, — это требования, которым должно удовлетворять отношение, вычисляемое программой
4. Формальная верификация программы π относительно спецификации Φ — это строгое доказательство того, что программа π удовлетворяет требованиям Φ, то есть доказательство включения $R_\pi \subseteq R_\Phi$

Принципы дедуктивной верификации программ

Чтобы уметь формально верифицировать программы, нужно:

1. Строго описать,
   - какие записи мы считаем программами (синтаксис программ)
   - как программы преобразуют входные данные в выходные (семантику программ)
2. Выбрать формальный язык описания требований к программам
3. Предложить алгоритм проверки того, удовлетворяет ли заданная программа предъявленным к ней требованиям

Напоминание: логика предикатов

Сигнатура, термы

Сигнатура логики предикатов $\langle Const, Func, Pred \rangle$ состоит из

- множества констант Const
- множества функциональных символов Func (символов операций)
- множества предикатных символов Pred (символов отношений)

Var — бесконечное множество переменных

Терм — это выражение, построенное над переменными, константами и функциональными символами

Пример терма в инфиксной записи:

$$\mathrm{x} + \mathbf{3} * \mathrm{y}$$

$$(\mathbf{3} \in Const; \quad +(\cdot,\cdot),\ *(\cdot,\cdot) \in Func)$$

$Term$ — множество всех термов (заданной сигнатуры)

Напоминание: логика предикатов

Формулы

Атом, или атомарная формула, — это выражение $\mathrm{P}(t_1, \ldots, t_k)$, где P — предикатный символ местности k, и t_i — термы

Формула строится из атомов с использованием конъюнкции ($\&$), дизъюнкции ($\vee$), отрицания ($\neg$), импликации ($\rightarrow$), квантора существования ($\exists$) и квантора всеобщности ($\forall$)

Пример формулы:

$$\forall \mathrm{x}\ (\mathrm{x} > \mathbf{2} \rightarrow \exists \mathrm{y}\ \exists \mathrm{z}\ (\mathrm{prime}(\mathrm{y})\ \&\ \mathrm{prime}(\mathrm{z})\ \&\ \mathrm{x} = \mathrm{y} + \mathrm{z}))$$

$$(\mathbf{2} \in \mathit{Const};\quad +(\cdot,\cdot) \in \mathit{Func};\quad >(\cdot,\cdot),\ =(\cdot,\cdot),\ \mathrm{prime}(\cdot) \in \mathit{Pred})$$

Приоритеты логических операций и кванторов (по убыванию):

$\forall$, $\exists$ и $\neg$; затем $\&$; затем $\vee$; затем $\rightarrow$

Императивные программы: синтаксис

Синтаксис императивных программ (сигнатуры σ) задаётся следующей формой Бэкуса-Наура:

$$
\begin{array}{lcl}
\pi & ::= & \emptyset \;\mid\; \mathit{instr} \;\mid\; \mathit{instr};\ \pi \\
\mathit{instr} & ::= & \mathtt{x} := t \;\mid \\
 & & \textbf{if}\ C\ \textbf{then}\ \pi\ \textbf{else}\ \pi\ \textbf{fi} \;\mid \\
 & & \textbf{while}\ C\ \textbf{do}\ \pi\ \textbf{od}
\end{array}
$$

Здесь

- π — программа
- $\emptyset$ — пустая программа
- instr — инструкция
  (сверху вниз в БНФ: присваивание, ветвление, цикл)
- $\mathtt{x}$ — переменная
- t — терм
- C — условие: формула, не содержащая кванторов

Императивные программы: операционная семантика

Два основных способа определения семантики программ:

денотационный и операционный

Денотационный:

- Значение каждой части программы — это математический объект: denotation (денотация, денотат)
- Денотация программы задаётся как композиция денотаций её составных частей
- Денотационной семантикой не определяется способ вычисления денотации на входных данных
- Хорошо подходит для описания значения функциональных программ
- Денотации функциональных программ — это функции

Императивные программы: операционная семантика

Два основных способа определения семантики программ:

денотационный и операционный

Операционный:

- Вычисление программы — это последовательное изменение состояния вычисления
- Программой задаётся отношение переходов, описывающее, какое состояние вычисления будет получено следующим из произвольного текущего состояния
- Значение программы — это функция преобразования входных данных в выходные, определяемая на основе транзитивного замыкания отношения переходов
- Хорошо подходит для описания значения императивных программ

Напоминание: логика предикатов

Подстановки

Подстановка — это отображение $\theta : \mathit{Var} \to \mathit{Term}$, такое что лишь для конечного числа переменных $\mathtt{x}$ верно неравенство $\theta(\mathtt{x}) \neq \mathtt{x}$

Связка — это выражение вида $\mathtt{x}/t$, где $\mathtt{x} \in \mathit{Var}$ и $t \in \mathit{Term}$

Подстановку можно задать конечным множеством связок: $\theta = \{\mathtt{x}_1/t_1, \ldots, \mathtt{x}_\mathtt{n}/t_n\}$ — это подстановка следующего вида:

- $\theta(\mathtt{x}_\mathtt{i}) = t_i$ для $1 \leq i \leq k$
- $\theta(\mathtt{y}) = \mathtt{y}$ для $\mathtt{y} \notin \{\mathtt{x}_1, \ldots, \mathtt{x}_\mathtt{k}\}$

Напоминание: логика предикатов

Свободные и связанные переменные

Связанное вхождение переменной в формулу — это вхождение переменной в область действия квантора, связывающего эту переменную

Свободное вхождение переменной — это вхождение, не являющееся связанным

Свободная переменная формулы — это переменная, имеющая свободное вхождение в формулу

Напоминание: логика предикатов

Применение подстановок, композиция подстановок

$E\theta$ — это результат применения подстановки θ к выражению E, то есть выражение, получающееся из E заменой

- каждого вхождения каждой переменной $\mathtt{x}$ на терм $\theta(\mathtt{x})$, если E — терм
- каждого свободного вхождения каждой свободной переменной $\mathtt{x}$ на терм $\theta(\mathtt{x})$, если E — формула

Например,

$$(\mathtt{x} + \mathtt{x} * \mathtt{y} * \mathtt{z})\{\mathtt{x}/\mathtt{y}, \mathtt{y}/\mathtt{z} + \mathbf{2}\} \equiv \mathtt{y} + \mathtt{y} * (\mathtt{z} + \mathbf{2}) * \mathtt{z}$$

$$(\mathtt{x} = \mathtt{y} \vee \exists \mathtt{x}\ (\mathtt{x} = \mathtt{y}))\{\mathtt{x}/\mathbf{1}, \mathtt{y}/\mathbf{2}\} \equiv \mathbf{1} = \mathbf{2} \vee \exists \mathtt{x}\ (\mathtt{x} = \mathbf{2})$$

Композиция подстановок θ и η — это подстановка $\theta\eta$, такая что для любой переменной $\mathtt{x}$ верно равенство

$$\mathtt{x}(\theta\eta) = (\mathtt{x}\theta)\eta$$

Напоминание: логика предикатов

Интерпретации

Интерпретация (сигнатуры $\langle Const, Func, Pred \rangle$) состоит из

- предметной области D (множества предметов)
- оценки констант
  - оценка константы $\mathbf{c}$ — это предмет $\overline{\mathbf{c}} \in D$
- оценки функциональных символов
  - оценка функционального символа $\mathbf{f}^{(k)}$ — это функция $\overline{\mathbf{f}} : D^k \to D$
- оценки предикатных символов
  - оценка предикатного символа $\mathrm{P}^{(k)}$ — это предикат $\overline{\mathrm{P}} : D^k \to \{\mathbf{true}, \mathbf{false}\}$

Например, предметная область (целочисленной) арифметической интерпретации $\mathcal{I}_{ar}$ — все целые числа, и все символы в этой интерпретации оцениваются естественным образом:

$$\overline{\mathbf{2}}\,\overline{+}\,\overline{\mathbf{2}} \quad \equiv \quad 4$$

Напоминание: логика предикатов

Истинность формул

Формула φ истинна в интерпретации $\mathcal{I}$ ($\mathcal{I} \models \varphi$), если имеет значение **true** при

- оценке всех символов согласно интерпретации $\mathcal{I}$,
- естественной трактовке логических связок и кванторов и
- замены всех свободных вхождений каждой свободной переменной на любой предмет

Например,

$$
\begin{array}{lcl}
\mathcal{I}_{ar} & \models & \mathrm{x} = \mathrm{x} \\
\mathcal{I}_{ar} & \not\models & \mathrm{x} = \mathbf{1} \\
\mathcal{I}_{ar} & \models & \forall \mathrm{x}\, \exists \mathrm{y}\, (\mathrm{y} = \mathrm{x} + \mathbf{1}) \\
\mathcal{I}_{ar} & \models & \exists \mathrm{y}\, (\mathrm{y} = \mathrm{x} + \mathbf{1}) \\
\mathcal{I}_{ar} & \not\models & \exists \mathrm{y}\, (\mathrm{x} = \mathbf{2} * \mathrm{y})
\end{array}
$$

Императивные программы: операционная семантика

Состояние управления — это произвольная программа

Состояние данных (оценка переменных) — это подстановка, отображающая каждую переменную программы в терм, не содержащий переменных

Состояние вычисления — это пара $\langle \pi, \theta \rangle$, где π — состояние управления, а θ — состояние данных

Отношение переходов $\to$ на множестве состояний вычисления для программы π в интерпретации $\mathcal{I}$ определяется так:

- $\langle \mathtt{x} := t,\ \theta \rangle \to \langle \emptyset,\ \{\mathtt{x}/t\}\,\theta \rangle$
- если $\mathcal{I} \models C\theta$, то
$$\langle \textbf{if } C \textbf{ then } \pi_1 \textbf{ else } \pi_2 \textbf{ fi},\ \theta \rangle \to \langle \pi_1,\ \theta \rangle$$
- если $\mathcal{I} \not\models C\theta$, то
$$\langle \textbf{if } C \textbf{ then } \pi_1 \textbf{ else } \pi_2 \textbf{ fi},\ \theta \rangle \to \langle \pi_2,\ \theta \rangle$$

Императивные программы: операционная семантика

Состояние управления — это произвольная программа

Состояние данных (оценка переменных) — это подстановка, отображающая каждую переменную программы в терм, не содержащий переменных

Состояние вычисления — это пара $\langle \pi, \theta \rangle$, где π — состояние управления, а θ — состояние данных

Отношение переходов $\to$ на множестве состояний вычисления для программы π в интерпретации $\mathcal{I}$ определяется так:

- если $\mathcal{I} \models C\theta$, то
$$\langle \textbf{while } C \textbf{ do } \pi \textbf{ od},\ \theta \rangle \to \langle \pi;\ \textbf{while } C \textbf{ do } \pi \textbf{ od},\ \theta \rangle$$
- если $\mathcal{I} \not\models C\theta$, то
$$\langle \textbf{while } C \textbf{ do } \pi \textbf{ od},\ \theta \rangle \to \langle \emptyset,\ \theta \rangle$$
- если $\langle \pi_1,\ \theta \rangle \to \langle \pi_1',\ \eta \rangle$, то
$$\langle \pi_1;\ \pi_2,\ \theta \rangle \to \langle \pi_1';\ \pi_2,\ \eta \rangle$$
- $\langle \emptyset;\ \pi,\ \theta \rangle \to \langle \pi,\ \theta \rangle$

Императивные программы: операционная семантика

Трасса программы π на оценке θ в интерпретации $\mathcal{I}$ — это последовательность состояний вычисления вида

$$\langle \pi,\ \theta \rangle \to \langle \pi_1,\ \theta_1 \rangle \to \langle \pi_2,\ \theta_2 \rangle \to \ldots$$

Вычисление программы π на оценке θ в интерпретации $\mathcal{I}$ — это максимальная по длине трасса π на θ в $\mathcal{I}$

Результат конечного вычисления программы π на оценке θ в интерпретации $\mathcal{I}$ — это оценка данных последнего состояния вычисления π на θ в $\mathcal{I}$

Иными словами, если $\langle \pi,\ \theta \rangle \to^* \langle \emptyset,\ \eta \rangle$, то η — результат вычисления π на θ в $\mathcal{I}$

($\to^*$ — *транзитивное замыкание отношения* $\to$)

Императивные программы: пример

π: **while** $\mathtt{x} > 0$ **do** $\mathtt{x} := \mathtt{x} - 1$ **od** $\qquad\qquad \theta = \{\mathtt{x}/1\}$

Вычисление π на θ в $\mathcal{I}_{ar}$:

$$\langle \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$
$$\downarrow$$
$$\langle \mathtt{x} := \mathtt{x} - 1;\ \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$

Пояснение:

$$(\mathtt{x} > 0)\,\{\mathtt{x}/1\} = (1 > 0)$$
$$\mathcal{I} \models (1 > 0)$$

Императивные программы: пример

π: **while** $\mathtt{x} > 0$ **do** $\mathtt{x} := \mathtt{x} - 1$ **od** $\qquad\qquad \theta = \{\mathtt{x}/1\}$

Вычисление π на θ в $\mathcal{I}_{ar}$:

$$\langle \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$
$$\downarrow$$
$$\langle \mathtt{x} := \mathtt{x} - 1;\ \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$
$$\downarrow$$
$$\langle \emptyset;\ \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1 - 1\}\rangle$$

Пояснение:

$$\{\mathtt{x}/\mathtt{x} - 1\}\,\{\mathtt{x}/1\} = \{\mathtt{x}/1 - 1\}$$

Императивные программы: пример

π: **while** $\mathtt{x} > 0$ **do** $\mathtt{x} := \mathtt{x} - 1$ **od** $\qquad\qquad \theta = \{\mathtt{x}/1\}$

Вычисление π на θ в $\mathcal{I}_{ar}$:

$$\langle \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$
$$\downarrow$$
$$\langle \mathtt{x} := \mathtt{x} - 1;\ \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$
$$\downarrow$$
$$\langle \emptyset;\ \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1 - 1\}\rangle$$
$$\downarrow$$
$$\langle \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1 - 1\}\rangle$$

Пояснение:

Императивные программы: пример

π: **while** $\mathtt{x} > 0$ **do** $\mathtt{x} := \mathtt{x} - 1$ **od** $\qquad\qquad \theta = \{\mathtt{x}/1\}$

Вычисление π на θ в $\mathcal{I}_{ar}$:

$$\langle \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$
$$\downarrow$$
$$\langle \mathtt{x} := \mathtt{x} - 1;\ \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$
$$\downarrow$$
$$\langle \emptyset;\ \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1-1\}\rangle$$
$$\downarrow$$
$$\langle \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1-1\}\rangle$$
$$\downarrow$$
$$\langle \emptyset,\ \{\mathtt{x}/1-1\}\rangle$$

Пояснение:

$$(\mathtt{x} > 0)\,\{\mathtt{x}/1-1\} = (1-1>0)$$
$$\mathcal{I} \not\models (1-1>0)$$

Императивные программы: пример

π: **while** $\mathtt{x} > 0$ **do** $\mathtt{x} := \mathtt{x} - 1$ **od** $\qquad\qquad \theta = \{\mathtt{x}/1\}$

Вычисление π на θ в $\mathcal{I}_{ar}$:

$$\langle \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$
$$\downarrow$$
$$\langle \mathtt{x} := \mathtt{x} - 1;\ \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1\}\rangle$$
$$\downarrow$$
$$\langle \emptyset;\ \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1-1\}\rangle$$
$$\downarrow$$
$$\langle \textbf{while } \mathtt{x} > 0 \textbf{ do } \mathtt{x} := \mathtt{x} - 1 \textbf{ od},\ \{\mathtt{x}/1-1\}\rangle$$
$$\downarrow$$
$$\langle \emptyset,\ \{\mathtt{x}/1-1\}\rangle$$

Пояснение:

$\{\mathtt{x}/1-1\}$ — результат вычисления

Задача верификации программ

Неформальная постановка

Программа частично корректна относительно требований, выражаемых предусловием φ и постусловием ψ, если для любых начальных данных, удовлетворяющих φ, результат любого конечного вычисления программы удовлетворяет ψ

Программа тотально корректна, если

- она частично корректна и
- любое её вычисление на начальных данных, удовлетворяющих предусловию φ, конечно

Задача верификации императивных программ состоит в проверке тотальной или частичной корректности заданной программы относительно заданных предусловия и постусловия

Задача верификации программ

Формальная постановка

Предусловие φ и постусловие ψ — это формулы логики предикатов, а π — императивная программа

Требование корректности π относительно φ, ψ записывается в виде триплета Хоара (или тройки Хоара):

$$\{\varphi\}\,\pi\,\{\psi\}$$

Триплет Хоара $\{\varphi\}\,\pi\,\{\psi\}$ истинен в интерпретации $\mathcal{I}$ ($\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$), если для любых оценок θ, η верно:

$$\text{если } \mathcal{I} \models \varphi\theta \text{ и } \langle \pi,\ \theta \rangle \to^* \langle \emptyset,\ \eta \rangle, \text{ то } \mathcal{I} \models \psi\eta$$

Программа π частично корректна в интерпретации $\mathcal{I}$ относительно предусловия φ и постусловия ψ, если

$$\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$$

Логика Хоара: теория

Для доказательства частичной корректности программ будем использовать систему правил[1] вида:

$$\frac{\Phi}{\Psi_1}, \quad \frac{\Phi}{\varphi}, \quad \frac{\Phi}{\Psi_1, \Psi_2}, \quad \frac{\Phi}{\varphi, \Psi_1, \psi}$$

(φ, ψ — формулы логики предикатов;
Φ, Ψ_1, Ψ_2 — триплеты Хоара)

Содержательно каждое из правил прочитывается так:
если истинны все триплеты и формулы, записанные под чертой,
то триплет Φ истинен

Правила можно прочитать и немного по-другому:
если **доказана** истинность
всех триплетов и формул под чертой,
то **доказана** и истинность триплета Φ

[1] Hoare C.A.R. An axiomatic basis for computer programming. 1969.

Логика Хоара: теория

Вот эти правила:

SKIP: $\dfrac{\{\varphi\}\,\emptyset\,\{\varphi\}}{\textbf{true}}$

AS: $\dfrac{\{\varphi\,\{\mathtt{x}/t\}\}\,\mathtt{x} := t\,\{\varphi\}}{\textbf{true}}$
(*переменная* $\mathtt{x}$ *свободна для терма* t *в формуле* φ)

INF: $\dfrac{\{\varphi\}\,\pi\,\{\psi\}}{\varphi \to \varphi', \{\varphi'\}\,\pi\,\{\psi'\}, \psi' \to \psi}$

COMP: $\dfrac{\{\varphi\}\,\pi_1;\ \pi_2\,\{\psi\}}{\{\varphi\}\,\pi_1\,\{\chi\}, \{\chi\}\,\pi_2\,\{\psi\}}$

IF: $\dfrac{\{\varphi\}\,\textbf{if}\ C\ \textbf{then}\ \pi_1\ \textbf{else}\ \pi_2\ \textbf{fi}\,\{\psi\}}{\{\varphi\,\&\,C\}\,\pi_1\,\{\psi\}, \{\varphi\,\&\,\neg C\}\,\pi_2\,\{\psi\}}$

WHILE: $\dfrac{\{\varphi\}\,\textbf{while}\ C\ \textbf{do}\ \pi\ \textbf{od}\,\{\varphi\,\&\,\neg C\}}{\{\varphi\,\&\,C\}\,\pi\,\{\varphi\}}$

Переменная $\mathtt{x}$ свободна для терма t в формуле φ, если ни одно свободное вхождение $\mathtt{x}$ не входит в область действия квантора, связывающего какую-либо переменную терма t

Формула φ в правиле **WHILE** называется инвариантом цикла

Логика Хоара: теория

Теорема корректности правил вывода Хоара

Для любой интерпретации $\mathcal{I}$ и любого правила вывода логики Хоара (SKIP, AS, INF, COMP, IF, WHILE)

$$\frac{\Phi}{\Psi_1}, \quad \frac{\Phi}{\varphi}, \quad \frac{\Phi}{\Psi_1, \Psi_2}, \quad \frac{\Phi}{\varphi, \Psi_1, \psi},$$

если $\mathcal{I} \models \Psi_1$, $\mathcal{I} \models \Psi_2$, $\mathcal{I} \models \varphi$ **и** $\mathcal{I} \models \psi$, **то** $\mathcal{I} \models \Phi$

Доказательство.

Подробно рассмотрим только правило **AS**: $\dfrac{\{\varphi\,\{\mathtt{x}/t\}\}\,\mathtt{x} := t\,\{\varphi\}}{\textbf{true}}$

Рассмотрим произвольную оценку θ, такую что $\mathcal{I} \models \varphi\,\{\mathtt{x}/t\}\,\theta$, и соотношение $\langle \mathtt{x} := t,\ \theta\rangle \to_{\mathcal{I}} \langle \pi,\ \eta\rangle$

Операционная семантика программ: $\pi = \emptyset$ и $\eta = \{\mathtt{x}/t\}\,\theta$

Значит, $\mathcal{I} \models \varphi\eta$ и $\mathcal{I} \models \{\varphi\,\{\mathtt{x}/t\}\}\,\mathtt{x} := t\,\{\varphi\}$

Корректность остальных правил доказывается аналогично ▼

Логика Хоара: теория

Зачем нужна теорема корректности[1]

Истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$ доказана, если построен успешный вывод этого триплета, то есть конечный вывод вида

$$\dfrac{\{\varphi\}\,\pi\,\{\psi\}}{\begin{array}{ccccc} \cdots & \dfrac{\cdots}{\chi} & \cdots & \dfrac{\cdots}{\chi' \quad \dfrac{\{\varphi'\}\,\pi'\,\{\psi'\}}{\cdots} \quad \chi''} & \cdots \\ \cdots & & \cdots & & \cdots \end{array}},$$

где

- под каждым триплетом Хоара в выводе стоит черта
- под каждой чертой записаны формулы и триплеты Хоара согласно правилам
- каждая формула, располагающаяся в выводе вне триплетов (как χ, χ', χ'' на рисунке), истинна в интерпретации $\mathcal{I}$

---

[1] Подробнее о выводе в логике Хоара рассказывается в курсе “Математическая логика и логическое программирование”

Логика Хоара: практика

Доказательство корректности программ, основанное на правилах вывода Хоара, можно сделать более наглядным, если "встроить" информацию о применении правил вывода непосредственно в текст программы

Аннотация — это запись вида $\{\varphi\}$, где φ — произвольная формула

Аннотированная программа — это программа, в которой до и после каждой инструкции могут располагаться последовательности аннотаций

Аннотация может расцениваться как

- предусловие инструкции, следующей за аннотацией
- постусловие инструкции, предшествующей аннотации
- составная часть триплета, располагающегося в выводе (некоторого исходного триплета)

Логика Хоара: практика

Аннотированная программа $\widetilde{\pi}$ обосновывает истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, если выполняются следующие условия:

1. при удалении всех аннотаций из $\widetilde{\pi}$ получается программа π
2. $\widetilde{\pi}$ начинается с аннотации $\{\varphi\}$
   и оканчивается аннотацией $\{\psi\}$
3. если $\pi = \emptyset$, то верно $\widetilde{\pi} = \{\varphi\}\,\emptyset\,\{\psi\}$ и $\mathcal{I} \models \varphi \to \psi$
4. перед каждой инструкцией и после каждой инструкции в $\widetilde{\pi}$ располагается хотя бы одна аннотация

Логика Хоара: практика

Аннотированная программа $\widetilde{\pi}$ обосновывает истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, если выполняются следующие условия:

5. аннотации перед каждым присваиванием и после него соотносятся так же, как и в правиле **AS**:
$$\{\chi\,\{\mathtt{x}/t\}\}\,\mathtt{x} := t\,\{\chi\},$$
где переменная $\mathtt{x}$ свободна для терма t в формуле χ

6. каждое ветвление в $\widetilde{\pi}$ аннотировано следующим образом:
$$\begin{array}{l}\{\chi_1\}\\ \textbf{if } C \textbf{ then}\\ \quad\{\chi_1 \,\&\, C\}\,\widetilde{\pi}_1\,\{\chi_2\}\\ \textbf{else}\\ \quad\{\chi_1 \,\&\, \neg C\}\,\widetilde{\pi}_2\,\{\chi_2\}\\ \textbf{fi}\\ \{\chi_2\}\end{array}$$

Логика Хоара: практика

Аннотированная программа $\widetilde{\pi}$ обосновывает истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, если выполняются следующие условия:

7. каждый цикл в $\widetilde{\pi}$ аннотирован следующим образом:

$$\begin{array}{l} \{\chi\} \\ \textbf{while } C \textbf{ do} \\ \quad \{\chi \,\&\, C\}\,\widetilde{\pi}'\,\{\chi\} \\ \textbf{od} \\ \{\chi \,\&\, \neg C\} \end{array}$$

8. аннотации, располагающиеся в $\widetilde{\pi}$ перед знаком “;” и после него, равны:

$$\widetilde{\pi}_1\,\{\chi\};\ \{\chi\}\,\widetilde{\pi}_2$$

9. для любых двух подряд идущих аннотаций $\{\chi_1\}\,\{\chi_2\}$ в $\widetilde{\pi}$ верно $\mathcal{I} \models \chi_1 \to \chi_2$

Логика Хоара: пример (алгоритм Эвклида)

Рассмотрим такую программу π:

$$\textbf{while } \mathtt{x} \neq \mathtt{y} \textbf{ do if } \mathtt{x} > \mathtt{y} \textbf{ then } \mathtt{x} := \mathtt{x} - \mathtt{y} \textbf{ else } \mathtt{y} := \mathtt{y} - \mathtt{x} \textbf{ fi od}$$

Докажем, что программа π корректно реализует вычисление наибольшего общего делителя натуральных чисел, хранящихся в $\mathtt{x}$ и $\mathtt{y}$, с записью результата в $\mathtt{x}$

Для этого достаточно доказать истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}_{ar}$, где:

$$\begin{array}{rl}
\varphi(\mathtt{x},\mathtt{y},\mathtt{z}): & \mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y},\mathtt{z}) \\
\psi(\mathtt{x},\mathtt{y},\mathtt{z}): & \mathtt{x} = \mathtt{z} \\
\gcd(\mathtt{x},\mathtt{y},\mathtt{z}): & \exists \mathtt{u}\,(\mathtt{z} \times \mathtt{u} = \mathtt{x}) \,\&\, \exists \mathtt{u}\,(\mathtt{z} \times \mathtt{u} = \mathtt{y}) \,\& \\
& \forall \mathtt{w}\,(\exists \mathtt{u}\,(\mathtt{w} \times \mathtt{u} = \mathtt{x}) \,\&\, \exists \mathtt{u}\,(\mathtt{w} \times \mathtt{u} = \mathtt{y}) \to (\mathtt{w} \leq \mathtt{x}))
\end{array}$$

Логика Хоара: пример (алгоритм Эвклида)

Аннотированная программа, обосновывающая истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$, может выглядеть так:

$$
\begin{array}{l}
\{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y},\mathtt{z})\} \\
\textbf{while } \mathtt{x} \neq \mathtt{y} \textbf{ do} \\
\quad \{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y},\mathtt{z}) \,\&\, \mathtt{x} \neq \mathtt{y}\} \\
\quad \textbf{if } \mathtt{x} > \mathtt{y} \textbf{ then} \\
\qquad \{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y},\mathtt{z}) \,\&\, \mathtt{x} \neq \mathtt{y} \,\&\, \mathtt{x} > \mathtt{y}\} \\
\qquad \{\mathtt{x} - \mathtt{y} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x} - \mathtt{y},\mathtt{y},\mathtt{z})\} \\
\qquad \mathtt{x} := \mathtt{x} - \mathtt{y} \\
\qquad \{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y},\mathtt{z})\} \\
\quad \textbf{else} \\
\qquad \{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y},\mathtt{z}) \,\&\, \mathtt{x} \neq \mathtt{y} \,\&\, \neg(\mathtt{x} > \mathtt{y})\} \\
\qquad \{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} - \mathtt{x} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y} - \mathtt{x},\mathtt{z})\} \\
\qquad \mathtt{y} := \mathtt{y} - \mathtt{x} \\
\qquad \{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y},\mathtt{z})\} \\
\quad \textbf{fi} \\
\quad \{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y},\mathtt{z})\} \\
\textbf{od} \\
\{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x},\mathtt{y},\mathtt{z}) \,\&\, \neg(\mathtt{x} \neq \mathtt{y})\} \\
\{\mathtt{x} = \mathtt{z}\}
\end{array}
$$

Логика Хоара: связь теории и практики

Теорема. Если существует аннотированная программа $\widetilde{\pi}$, обосновывающая истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, то $\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$

Доказательство.

По аннотированной программе $\widetilde{\pi}$ можно легко построить успешный вывод триплета $\{\varphi\}\,\pi\,\{\psi\}$:

Фрагмент программы $\widetilde{\pi}$ вида

$$\{\chi\}\,\emptyset\,\{\chi\}$$

соответствует фрагменту вывода

$$\textbf{SKIP}:\ \frac{\{\chi\}\,\emptyset\,\{\chi\}}{\textbf{true}}$$

Логика Хоара: связь теории и практики

Теорема. Если существует аннотированная программа $\widetilde{\pi}$, обосновывающая истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, то $\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$

Доказательство.

По аннотированной программе $\widetilde{\pi}$ можно легко построить успешный вывод триплета $\{\varphi\}\,\pi\,\{\psi\}$:

Фрагмент программы $\widetilde{\pi}$ вида

$$\{\chi\,\{\mathtt{x}/t\}\}\,\mathtt{x} := t\,\{\chi\}$$

соответствует фрагменту вывода

$$\textbf{AS}\colon\ \frac{\{\chi\,\{\mathtt{x}/t\}\}\,\mathtt{x} := t\,\{\chi\}}{\textbf{true}}$$

Логика Хоара: связь теории и практики

Теорема. **Если существует аннотированная программа $\widetilde{\pi}$, обосновывающая истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, то $\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$**

Доказательство.

По аннотированной программе $\widetilde{\pi}$ можно легко построить успешный вывод триплета $\{\varphi\}\,\pi\,\{\psi\}$:

Фрагмент программы $\widetilde{\pi}$ вида

$$\{\chi\}\,\{\chi'\}\,\widetilde{\pi}'\,\{\chi''\}$$

соответствует фрагменту вывода

$$\textbf{INF}\colon\ \frac{\{\chi\}\,\pi'\,\{\chi''\}}{\chi \to \chi', \{\chi'\}\,\pi'\,\{\chi''\}, \chi'' \to \chi''}$$

Логика Хоара: связь теории и практики

Теорема. Если существует аннотированная программа $\widetilde{\pi}$, обосновывающая истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, то $\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$

Доказательство.

По аннотированной программе $\widetilde{\pi}$ можно легко построить успешный вывод триплета $\{\varphi\}\,\pi\,\{\psi\}$:

Фрагмент программы $\widetilde{\pi}$ вида

$$\{\chi\}\,\widetilde{\pi}'\,\{\chi'\}\,\{\chi''\}$$

соответствует фрагменту вывода

$$\textbf{INF}\colon\ \frac{\{\chi\}\,\pi'\,\{\chi''\}}{\chi \to \chi, \{\chi\}\,\pi'\,\{\chi'\}\,, \chi' \to \chi''}$$

Логика Хоара: связь теории и практики

Теорема. Если существует аннотированная программа $\widetilde{\pi}$, обосновывающая истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, то $\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$

Доказательство.

По аннотированной программе $\widetilde{\pi}$ можно легко построить успешный вывод триплета $\{\varphi\}\,\pi\,\{\psi\}$:

Фрагмент программы $\widetilde{\pi}$ вида

$$\{\chi\}\,\widetilde{\pi}_1\,\{\chi'\};\ \{\chi'\}\,\widetilde{\pi}_2\,\{\chi''\}$$

соответствует фрагменту вывода

$$\textbf{COMP}:\ \frac{\{\chi\}\,\pi_1;\ \pi_2\,\{\chi''\}}{\{\chi\}\,\pi_1\,\{\chi'\}\,,\{\chi'\}\,\pi_2\,\{\chi''\}}$$

Логика Хоара: связь теории и практики

Теорема. **Если существует аннотированная программа $\widetilde{\pi}$, обосновывающая истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, то $\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$**

Доказательство.

По аннотированной программе $\widetilde{\pi}$ можно легко построить успешный вывод триплета $\{\varphi\}\,\pi\,\{\psi\}$:

Фрагмент программы $\widetilde{\pi}$ вида

$$\{\chi\}\,\textbf{if}\ C\ \textbf{then}\ \{\chi\,\&\,C\}\,\widetilde{\pi}_1\,\{\chi'\}\ \textbf{else}\ \{\chi\,\&\,\neg C\}\,\widetilde{\pi}_2\,\{\chi'\}\ \textbf{fi}\,\{\chi'\}$$

соответствует фрагменту вывода

$$\textbf{IF}\colon\ \frac{\{\chi\}\,\textbf{if}\ C\ \textbf{then}\ \pi_1\ \textbf{else}\ \pi_2\ \textbf{fi}\,\{\chi'\}}{\{\chi\,\&\,C\}\,\pi_1\,\{\chi'\}\,,\,\{\chi\,\&\,\neg C\}\,\pi_2\,\{\chi'\}}$$

Логика Хоара: связь теории и практики

Теорема. **Если существует аннотированная программа $\widetilde{\pi}$, обосновывающая истинность триплета $\{\varphi\}\,\pi\,\{\psi\}$ в интерпретации $\mathcal{I}$, то $\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$**

Доказательство.

По аннотированной программе $\widetilde{\pi}$ можно легко построить успешный вывод триплета $\{\varphi\}\,\pi\,\{\psi\}$:

Фрагмент программы $\widetilde{\pi}$ вида

$$\{\chi\}\,\textbf{while}\ C\ \textbf{do}\ \{\chi\,\&\,C\}\,\widetilde{\pi}'\,\{\chi\}\ \textbf{od}\,\{\chi\,\&\,\neg C\}$$

соответствует фрагменту вывода

$$\textbf{WHILE}\colon\ \frac{\{\chi\}\,\textbf{while}\ C\ \textbf{do}\ \pi'\ \textbf{od}\,\{\chi\,\&\,\neg C\}}{\{\chi\,\&\,C\}\,\pi'\,\{\chi\}}$$

▼

Автоматическая проверка правильности программ

Можно ли автоматизировать проверку корректности программ?

Иллюстрация, показывающая, насколько это непростая задача:

доказательство истинности триплета

$$\{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \mathtt{z} > \mathbf{0} \,\&\, \mathtt{n} > \mathbf{2}\}$$
$$\textbf{if } \mathtt{x}^{\mathtt{n}} + \mathtt{y}^{\mathtt{n}} = \mathtt{z}^{\mathtt{n}} \textbf{ then } \mathtt{u} := \mathbf{0} \textbf{ else } \mathtt{u} := 1 \textbf{ fi}$$
$$\{\mathtt{u} > \mathbf{0}\}$$

в интерпретации $\mathcal{I}_{ar}$ равносильно доказательству Великой теоремы Ферма

Автоматическая проверка правильности программ

Можно ли автоматизировать проверку корректности программ?

Более того, проблема корректности программ в общем случае **неразрешима**:

- программы с операцией $+$ и отношением $<$ в интерпретации $\mathcal{I}_{ar}$ полны по Тьюрингу
- любое нетривиальное семантическое свойство частично-рекурсивных функций, а значит, и машин Тьюринга, неразрешимо
  - (*Rice. Classes of recursively enumerable sets and their decision problems. 1953*)

Автоматическая проверка правильности программ

Слабейшее предусловие[1] для программы π и постусловия ψ в интерпретации $\mathcal{I}$ — это формула $wpr(\pi, \psi, \mathcal{I})$, для которой выполнены следующие условия:

1. $\mathcal{I} \models \{wpr(\pi, \psi, \mathcal{I})\}\, \pi\, \{\psi\}$
2. Для любой формулы χ, такой что $\mathcal{I} \models \{\chi\}\, \pi\, \{\psi\}$, верно $\mathcal{I} \models \chi \to wpr(\pi, \psi, \mathcal{I})$

Будем писать $wpr(\pi, \psi)$ вместо $wpr(\pi, \psi, \mathcal{I})$, если слабейшее предусловие не зависит существенно от интерпретации $\mathcal{I}$.

Теорема. $\mathcal{I} \models \{\varphi\}\, \pi\, \{\psi\} \quad \Leftrightarrow \quad \mathcal{I} \models \varphi \to wpr(\pi, \psi, \mathcal{I})$

Доказательство. Очевидно?

---

[1] Weakest precondition

Автоматическая проверка правильности программ

Таким образом, для полной автоматизации проверки корректности программ достаточно иметь два алгоритма:

1. Алгоритм проверки истинности формул логики предикатов
2. Алгоритм вычисления слабейшего предусловия для произвольных программ и постусловий

Первый алгоритм зависит от выбора интерпретации программ, и в общем случае может не существовать

Автоматическая проверка правильности программ

Таким образом, для полной автоматизации проверки корректности программ достаточно иметь два алгоритма:

1. Алгоритм проверки истинности формул логики предикатов
2. Алгоритм вычисления слабейшего предусловия для произвольных программ и постусловий

Со вторым алгоритмом всё немного лучше:

Теорема.

- $wpr(\mathtt{x} := t, \psi) = \psi\,\{\mathtt{x}/t\}$
- $wpr(\pi_1;\ \pi_2, \psi) = wpr(\pi_1, wpr(\pi_2, \psi))$
- $wpr(\textbf{if } C \textbf{ then } \pi_1 \textbf{ else } \pi_2 \textbf{ fi}, \psi) =$
$$C \,\&\, wpr(\pi_1, \psi) \vee \neg C \,\&\, wpr(\pi_2, \psi)$$

Доказательство. Это очень просто

Автоматическая проверка правильности программ

$$\textbf{WHILE}: \frac{\{\varphi\}\,\textbf{while}\ C\ \textbf{do}\ \pi\ \textbf{od}\,\{\varphi\,\&\,\neg C\}}{\{\varphi\,\&\,C\}\,\pi\,\{\varphi\}}$$

Для применения правила **WHILE** (а также для аннотации цикла, и для вычисления слабейшего предусловия цикла) необходимо предварительно найти подходящую формулу φ —

инвариант цикла

Автоматическая генерация инвариантов циклов, как правило, является **основной проблемой** автоматизации проверки правильности программ

Эта проблема осложняется тем, что

1. никто не гарантирует существование свойства данных, выражаемого инвариантом цикла
2. если такое свойство существует, то никто не гарантирует, что это свойство можно записать в виде формулы в используемой сигнатуре

Конец лекции 2