Математическая логика и логическое программирование

mk.cs.msu.ru → Лекционные курсы
→ Математическая логика и логическое программирование (3-й поток)

Блок 53

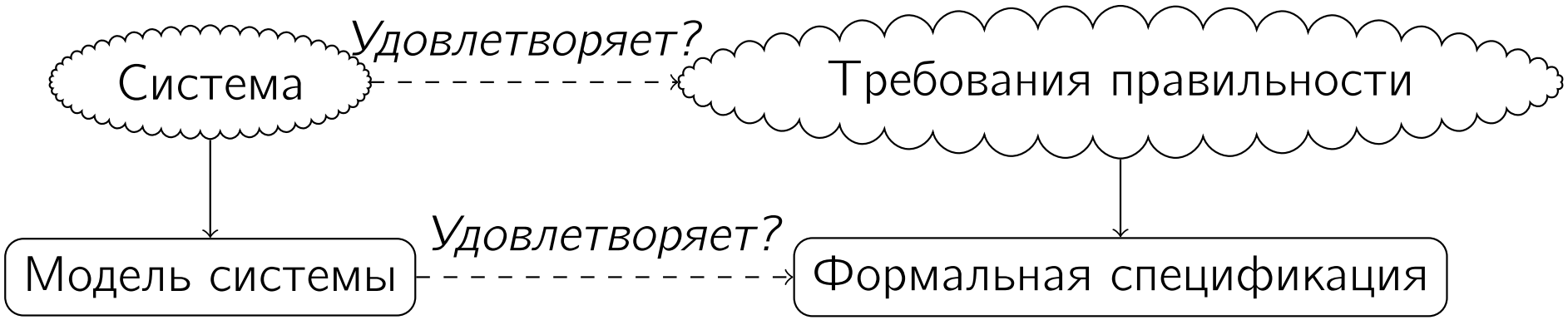
Модельные императивные программы
Постановка задачи верификации программ

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2023/2024, осенний семестр

Вступление

Формальная верификация:



Обсудим то, как можно использовать **логику предикатов** для формальной верификации **императивных программ**

- Как может быть устроена математическая модель программы?
- Как можно записывать требования к программе с использованием логики предикатов?
- Как проверить, удовлетворяет ли модель программы записанным требованиям?

Императивные программы: синтаксис

Далее считаются заданными сигнатура σ логики предикатов и множество предметных переменных Var

Синтаксис императивных программ зададим следующей БНФ:

$$
\begin{array}{lll}
\pi & ::= & \mathit{stmt} \mid \mathit{stmt}\ \pi \\
\mathit{stmt} & ::= & \emptyset \mid \qquad\qquad\qquad\qquad\qquad\qquad \text{(пустая команда)} \\
 & & \mathtt{x} := t; \mid \qquad\qquad\qquad\qquad\qquad\qquad \text{(присваивание)} \\
 & & \textbf{if } C \textbf{ then } \pi \textbf{ else } \pi \textbf{ fi} \mid \qquad \text{(ветвление)} \\
 & & \textbf{while } C \textbf{ do } \pi \textbf{ od} \qquad\qquad\qquad\qquad \text{(цикл)}
\end{array}
$$

Здесь:

- π — программа
- stmt — команда программы (или, по-другому, инструкция)
- $\mathtt{x} \in \mathrm{Var}$
- t — выражение: произвольный терм, такой что $\mathrm{Var}_t \subseteq \mathrm{Var}$
- C — условие: произвольная формула без кванторов, такая что $\mathrm{Var}_C \subseteq \mathrm{Var}$

Императивные программы: синтаксис

В примерах будут использоваться

- сигнатура, содержащая общеизвестные арифметические символы, включая
  - константы 0, 1,
  - функциональные символы $+^{(2)}$, $-^{(2)}$, $.^{(2)}$,
  - предикатные символы $=^{(2)}$, $>^{(2)}$, $\geq^{(2)}$

- арифметическая интерпретация Ar_X логики предикатов над множеством чисел X, в которой эти символы оцениваются «естественно» как соответствующие чи́сла, операции и отношения, в том числе 0, 1, $+$, $-$, $\cdot$, $=$, $>$, $\geq$ — соответственно как
  - чи́сла 0 и 1,
  - операции сложения, вычитания и умножения чисел и
  - отношения равенства чисел и их строгого и нестрогого неравенства в бо́льшую сторону

Императивные программы: синтаксис

Пример: реализация алгоритма Эвклида
вычисления наибольшего общего делителя чисел в переменных $\mathtt{x}$, $\mathtt{y}$

while $\neg(\mathtt{x} = \mathtt{y})$ **do**
  if $\mathtt{x} > \mathtt{y}$ **then**
    $\mathtt{x} := \mathtt{x} - \mathtt{y};$
  else
    $\mathtt{y} := \mathtt{y} - \mathtt{x};$
  fi
od

Императивные программы: операционная семантика

Значение программы — это **вычисляемая ей функция** преобразования входных данных в выходные данные

Для задания этой функции определим следующие понятия:

- Состояние данных: совокупность значений переменных, преобразуемая при выполнении программы
- Состояние управления: описание того, как текущее состояние данных будет изменяться программой в дальнейшем выполнении
- Состояние вычисления: состояние данных + состояние управления, то есть описание значений данных сейчас и в оставшейся части выполнения программы

Императивные программы: операционная семантика

Состояние данных над переменными Var в интерпретации с предметной областью D — это отображение $\sigma : \mathrm{Var} \to \mathrm{D}$
Обозначение: $[\mathtt{x}_1/\sigma(x_1), \ldots, \mathtt{x}_\mathtt{n}/\sigma(x_n)]$, если $\mathrm{Var} = \{\mathtt{x}_1, \ldots, \mathtt{x}_\mathtt{n}\}$

Состояние управления — это произвольная программа

Состояние вычисления — это пара $\langle \pi \mid \sigma \rangle$,
где π — состояние управления и σ — состояние данных

Σ — множество всех состояний данных

$\widetilde{\Sigma}$ — множество всех состояний вычисления

$\sigma\{x \leftarrow d\}$ — состояние данных, получающееся из состояния данных σ в результате присваивания переменной $\mathtt{x}$ значения d:

$$\sigma\{x \leftarrow d\}(\mathtt{x}) = d$$
$$\sigma\{x \leftarrow d\}(\mathtt{y}) = \sigma(\mathtt{y}), \text{ если } \mathtt{y} \neq \mathtt{x}$$

Императивные программы: операционная семантика

Шаг выполнения программы в интерпретации $\mathcal{I}$ описывается двуместным отношением переходов $\xrightarrow{\mathcal{I}}$ на множестве $\widetilde{\Sigma}$, состоящим из всех пар следующего вида:

- $\langle \mathrm{x} := t; \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \emptyset \mid \sigma\{x \leftarrow t\sigma\} \rangle$
- $\langle \textbf{if } C \textbf{ then } \pi_1 \textbf{ else } \pi_2 \textbf{ fi} \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \pi_1 \mid \sigma \rangle$, если $\mathcal{I} \models C\sigma$
- $\langle \textbf{if } C \textbf{ then } \pi_1 \textbf{ else } \pi_2 \textbf{ fi} \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \pi_2 \mid \sigma \rangle$, если $\mathcal{I} \not\models C\sigma$
- $\langle \textbf{while } C \textbf{ do } \pi \textbf{ od} \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \emptyset \mid \sigma \rangle$, если $\mathcal{I} \not\models C\sigma$
- $\langle \textbf{while } C \textbf{ do } \pi \textbf{ od} \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \pi \textbf{ while } C \textbf{ do } \pi \textbf{ od} \mid \sigma \rangle$, если $\mathcal{I} \models C\sigma$
- $\langle \pi_1\ \pi_2 \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \pi_1'\ \pi_2 \mid \sigma' \rangle$, если $\langle \pi_1 \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \pi_1' \mid \sigma' \rangle$
- $\langle \emptyset\ \pi \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \pi \mid \sigma \rangle$

Императивные программы: операционная семантика

Трасса программы π из состояния данных σ в интерпретации $\mathcal{I}$ — это последовательность состояний вычисления вида

$$\langle \pi \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \pi_1 \mid \sigma_1 \rangle \xrightarrow{\mathcal{I}} \langle \pi_2 \mid \sigma_2 \rangle \xrightarrow{\mathcal{I}} \dots$$

Вычислениями программы называются бесконечные трассы и трассы, оканчивающиеся состоянием управления $\emptyset$

Последнее состояние данных конечной трассы называется результатом этой трассы

Запись $\langle \pi \mid \sigma \rangle \xrightarrow{\mathcal{I}}_* \widetilde{\sigma}$ будет означать, что существует конечная трасса программы π из состояния данных σ в интерпретации $\mathcal{I}$, оканчивающаяся состоянием вычисления $\widetilde{\sigma}$

Программой π в интерпретации $\mathcal{I}$ вычисляется частичная функция $\mathcal{I}[\pi] : \Sigma \to \Sigma$ следующего вида:

$$\mathcal{I}[\pi](\sigma) = \sigma' \quad \Leftrightarrow \quad \langle \pi \mid \sigma \rangle \xrightarrow{\mathcal{I}}_* \langle \emptyset \mid \sigma' \rangle$$

Императивные программы: операционная семантика (пример)

$$\mathrm{Var} = \{\mathrm{x}, \mathrm{y}\}$$

$$\pi = \textbf{while } \neg(\mathrm{x} = \mathrm{y}) \textbf{ do if } \mathrm{x} > \mathrm{y} \textbf{ then } \mathrm{x} := \mathrm{x} - \mathrm{y}; \textbf{ else } \mathrm{y} := \mathrm{y} - \mathrm{x}; \textbf{ fi od}$$

Вычисление π из $[\mathrm{x}/2, \mathrm{y}/4]$ в $Ar_{\mathbb{Z}}$, где $\mathbb{Z}$ — множество всех целых чисел:

$$\langle \pi \mid [\mathrm{x}/2, \mathrm{y}/4] \rangle$$

$$Ar_{\mathbb{Z}} \downarrow \quad \text{т.к. } Ar_{\mathbb{Z}} \models \neg(\mathrm{x} = \mathrm{y})[\mathrm{x}/2, \mathrm{y}/4]$$

$$\langle \textbf{if } \mathrm{x} > \mathrm{y} \textbf{ then } \mathrm{x} := \mathrm{x} - \mathrm{y}; \textbf{ else } \mathrm{y} := \mathrm{y} - \mathrm{x}; \textbf{ fi } \pi \mid [\mathrm{x}/2, \mathrm{y}/4] \rangle$$

$$Ar_{\mathbb{Z}} \downarrow \quad \text{т.к. } Ar_{\mathbb{Z}} \not\models (\mathrm{x} > \mathrm{y})[\mathrm{x}/2, \mathrm{y}/4]$$

$$\langle \mathrm{y} := \mathrm{y} - \mathrm{x}; \pi \mid [\mathrm{x}/2, \mathrm{y}/4] \rangle$$

$$Ar_{\mathbb{Z}} \downarrow \quad \text{т.к. } [\mathrm{x}/2, \mathrm{y}/4]\{\mathrm{y} \leftarrow (\mathrm{y} - \mathrm{x})[\mathrm{x}/2, \mathrm{y}/4]\} = [\mathrm{x}/2, \mathrm{y}/2]$$

$$\langle \emptyset\ \pi \mid [\mathrm{x}/2, \mathrm{y}/2] \rangle$$

$$Ar_{\mathbb{Z}} \downarrow$$

$$\langle \pi \mid [\mathrm{x}/2, \mathrm{y}/2] \rangle$$

$$Ar_{\mathbb{Z}} \downarrow \quad \text{т.к. } Ar_{\mathbb{Z}} \not\models \neg(\mathrm{x} = \mathrm{y})[\mathrm{x}/2, \mathrm{y}/2]$$

$$\langle \emptyset \mid [\mathrm{x}/2, \mathrm{y}/2] \rangle$$

Результат этого вычисления: $[\mathrm{x}/2, \mathrm{y}/2]$

Следовательно, $Ar_{\mathbb{Z}}[\pi]([\mathrm{x}/2, \mathrm{y}/4]) = [\mathrm{x}/2, \mathrm{y}/2]$

Задача верификации программ

Требования правильности выполнения программы могут быть записаны как два отношений на состояниях данных:

- предусловие φ,
  задающее общий вид допустимых входных данных
- постусловие ψ,
  описывающее устройство правильных выходных данных

Принято рассматривать два вида правильности выполнения программы относительно заданных предусловия и постусловия:

- частичная корректность: результат любого **конечного** вычисления программы на допустимых входных данных правилен
- полная корректность: любое вычисление программы на допустимых входных данных **конечно**, и результат этого вычисления правилен

Остановимся подробнее на **частичной** корректности программ

Задача верификации программ

Тройка Хоара (по-другому — триплет Хоара) — это запись вида $\{\varphi\}\pi\{\psi\}$, где

- φ — формула логики предикатов, называемая предусловием
- π — программа
- ψ — формула логики предикатов, называемая постусловием

Триплет $\{\varphi\}\pi\{\psi\}$ истинен в интерпретации $\mathcal{I}$ ($\mathcal{I} \models \{\varphi\}\pi\{\psi\}$), если для любых состояний данных σ, σ' верно следующее:

если $\mathcal{I} \models \varphi\sigma$ и значение $\sigma' = \mathcal{I}[\pi](\sigma)$ определено, то $\mathcal{I} \models \psi\sigma'$

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

Задача верификации императивных программ:
для заданных программы π, предусловия φ, постусловия ψ и интерпретации $\mathcal{I}$ проверить справедливость соотношения $\mathcal{I} \models \{\varphi\}\pi\{\psi\}$