

# Математическая логика и логическое программирование

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

## Блок 54

Логика Хоара  
Автоматизация проверки правильности программ

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

ВМК МГУ, 2022/2023, осенний семестр

# Логика Хоара

Для проверки проверки частичной корректности модельных императивных программ можно адаптировать **метод семантических таблиц**: ввести понятие **вывода** (дерева, построенного по **правилам вывода**) и свести проверку корректности к построению **успешного** вывода

Правила вывода будут выглядеть так:<sup>1</sup>

$$\frac{\Phi}{\Psi}, \quad \frac{\Phi}{\Psi, \Omega}, \quad \frac{\Phi}{\varphi} \quad \text{или} \quad \frac{\Phi}{\varphi, \Omega, \psi'}$$

где  $\Phi, \Psi, \Omega$  — триплеты Хоара и  $\varphi, \psi$  — формулы логики предикатов

## Содержательное прочтение:

**если** в  $\mathcal{I}$  истинны все триплеты и формулы под чертой  
**то** в  $\mathcal{I}$  истинен триплет  $\Phi$

---

<sup>1</sup> Hoare C.A.R. An axiomatic basis for computer programming. 1969

# Логика Хоара

Вот эти правила:

$$R_{\emptyset} : \frac{\{\varphi\} \emptyset \{\varphi\}}{\top} \quad R_{:=} : \frac{\{\varphi\{x/t\}\} x := t; \{\varphi\}}{\top}$$

(подстановка  $\{x/t\}$  правильна для  $\varphi$ )

$$R_{\text{if}} : \frac{\{\varphi\} \text{if } C \text{ then } \pi_1 \text{ else } \pi_2 \text{ fi } \{\psi\}}{\{\varphi \ \& \ C\} \pi_1 \{\psi\}, \{\varphi \ \& \ \neg C\} \pi_2 \{\psi\}}$$

$$R_{\text{while}} : \frac{\{\varphi\} \text{while } C \text{ do } \pi \text{ od } \{\varphi \ \& \ \neg C\}}{\{\varphi \ \& \ C\} \pi \{\varphi\}}$$

$$R_{\text{seq}} : \frac{\{\varphi\} \pi_1 \pi_2 \{\psi\}}{\{\varphi\} \pi_1 \{\chi\}, \{\chi\} \pi_2 \{\psi\}}$$

$$R_{\text{inf}} : \frac{\{\varphi\} \pi \{\psi\}}{\varphi \rightarrow \varphi', \{\varphi'\} \pi \{\psi'\}, \psi' \rightarrow \psi}$$

# Логика Хоара

## Лемма (о корректности правил вывода Хоара)

Для любой интерпретации  $\mathcal{I}$  и любого из правил

$$\begin{array}{c} R_{\emptyset}, R_{:=}, R_{\text{if}}, R_{\text{while}}, R_{\text{seq}}, R_{\text{while}} \\ \Phi \\ \left( \frac{}{\Psi}, \frac{}{\Psi, \Omega}, \frac{}{\varphi}, \frac{}{\varphi, \Psi, \psi} \right) \end{array}$$

верно следующее: если  $\mathcal{I} \models \Psi$ ,  $\mathcal{I} \models \Omega$ ,  $\mathcal{I} \models \varphi$  и  $\mathcal{I} \models \psi$ , то  $\mathcal{I} \models \Phi$

Доказательство.

Подробно рассмотрим только правило  $R_{:=}: \frac{\{\varphi\{x/t\}\}x := t; \{\varphi\}}{\vdash}$

Пусть  $\sigma$  — произвольное состояние данных, такое что  $\mathcal{I} \models \varphi\{x/t\}\sigma$

Шаг вычисления для программы « $x := t$ ;» устроен так:

$$\langle x := t; \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \emptyset \mid \sigma\{x \leftarrow t\sigma\} \rangle$$

Так как  $\mathcal{I} \models \varphi\{x/t\}\sigma$ , то верно и  $\mathcal{I} \models \varphi(\sigma\{x \leftarrow t\sigma\})$

Следовательно,  $\mathcal{I} \models \{\varphi\{x/t\}\}x := t; \{\varphi\}$

Корректность остальных правил обосновывается по той же схеме



# Логика Хоара

**Вывод триплета**  $\{\varphi\}\pi\{\psi\}$  — это дерево следующего вида:

- ▶ вершины размечены триплетами и формулами
- ▶ корень помечен триплетом  $\{\varphi\}\pi\{\psi\}$
- ▶ дети вершины определяются относительно правил логики Хоара так же, как и в дереве табличного вывода относительно правил табличного вывода
- ▶ все листья помечены формулами

**Успешный вывод** триплета  $\{\varphi\}\pi\{\psi\}$  в интерпретации  $\mathcal{I}$  — это конечный вывод, все листья которого помечены формулами, истинными в  $\mathcal{I}$

# Логика Хоара

## Теорема (о корректности логики Хоара)

Если существует успешный вывод триплета  $\{\varphi\}\pi\{\psi\}$  в интерпретации  $\mathcal{I}$ , то  $\mathcal{I} \models \{\varphi\}\pi\{\psi\}$

Доказательство.

Рассмотрим произвольный успешный вывод  $D$  для  $\{\varphi\}\pi\{\psi\}$  в  $\mathcal{I}$

Во всех листьях  $D$  записаны формулы, истинные в  $\mathcal{I}$

Применив лемму о корректности правил вывода конечное число раз, получим истинность триплета  $\{\varphi\}\pi\{\psi\}$  в  $\mathcal{I}$  ▼

# Логика Хоара

## Пример: алгоритм Эвклида

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

Докажем, что в результате выполнения  $\pi$  в интерпретации  $Ar_{\mathbb{Z}}$  в  $x$  записывается **наибольший общий делитель** (НОД) положительных чисел, заданных в  $x$  и  $y$  перед выполнением

*Предусловие*  $\varphi$ : числа  $x$  и  $y$  положительны, и  $z$  — переменная, обозначающая НОД  $x$  и  $y$  в начале выполнения программы

$$\begin{aligned} u|v &= \exists w (v = u \cdot w) \\ \text{gcd}(u, v, w) &= (w|u) \& (w|v) \& \forall r ((r|u) \& (r|v) \rightarrow (w \geq r)) \\ \varphi &= x > 0 \& y > 0 \& \text{gcd}(x, y, z) \end{aligned}$$

*Постусловие*  $\psi$ : в  $x$  записан требуемый НОД

$$\psi = (x = z)$$

Для обоснования правильности  $\pi$  достаточно построить успешный табличный вывод для триплета  $\{\varphi\}\pi\{\psi\}$

# Логика Хоара

```
{x > 0 & y > 0 & gcd(x, y, z)}  
while ¬(x = y) do if x > y then x := x - y; else y := y - x; fi od  
{x = z}
```

$\chi_1: x > 0 \& x > 0 \& \text{gcd}(x, y, z) \rightarrow x > 0 \& y > 0 \& \text{gcd}(x, y, z)$

$\chi_2: x > 0 \& y > 0 \& \text{gcd}(x, y, z) \& \neg\neg(x = y) \rightarrow x = z$

```
{x > 0 & y > 0 & gcd(x, y, z)}  
while ¬(x = y) do if x > y then x := x - y; else y := y - x; fi od  
{x > 0 & y > 0 & gcd(x, y, z) & ¬¬(x = y)}
```

$$R_{inf}: \frac{\{\varphi\}\pi\{\psi\}}{\varphi \rightarrow \varphi', \{\varphi'\}\pi\{\psi'\}, \psi' \rightarrow \psi}$$

$$\mathcal{I} \models \chi_1$$

$$\mathcal{I} \models \chi_2$$



# Логика Хоара

```
{x > 0 & y > 0 & gcd(x, y, z)}  
while ¬(x = y) do if x > y then x := x - y; else y := y - x; fi od  
{x = z}
```

$\chi_1: x > 0 \ \& \ x > 0 \ \& \ \text{gcd}(x, y, z) \rightarrow x > 0 \ \& \ y > 0 \ \& \ \text{gcd}(x, y, z)$

$\chi_2: x > 0 \ \& \ y > 0 \ \& \ \text{gcd}(x, y, z) \ \& \ \neg\neg(x = y) \rightarrow x = z$

```
{x > 0 & y > 0 & gcd(x, y, z)}  
while ¬(x = y) do if x > y then x := x - y; else y := y - x; fi od  
{x > 0 & y > 0 & gcd(x, y, z) & ¬¬(x = y)}
```

```
{x > 0 & y > 0 & gcd(x, y, z) & ¬(x = y)}  
if x > y then x := x - y; else y := y - x; fi  
{x > 0 & y > 0 & gcd(x, y, z)}
```

$$R_{\text{while}}: \frac{\{\varphi\} \text{while } C \text{ do } \pi \text{ od } \{\varphi \ \& \ \neg C\}}{\{\varphi \ \& \ C\} \pi \{\varphi\}}$$

# Логика Хоара

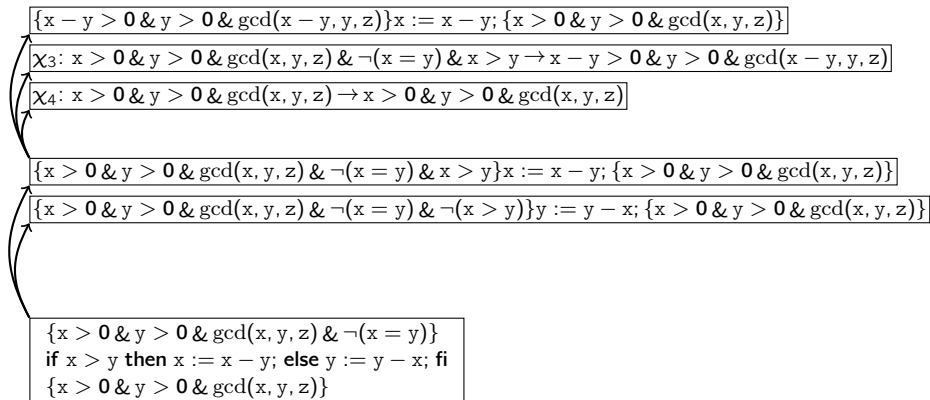
$\{x > 0 \ \& \ y > 0 \ \& \ \text{gcd}(x, y, z) \ \& \ \neg(x = y) \ \& \ x > y\} x := x - y; \{x > 0 \ \& \ y > 0 \ \& \ \text{gcd}(x, y, z)\}$

$\{x > 0 \ \& \ y > 0 \ \& \ \text{gcd}(x, y, z) \ \& \ \neg(x = y) \ \& \ \neg(x > y)\} y := y - x; \{x > 0 \ \& \ y > 0 \ \& \ \text{gcd}(x, y, z)\}$

$\{x > 0 \ \& \ y > 0 \ \& \ \text{gcd}(x, y, z) \ \& \ \neg(x = y)\}$   
**if**  $x > y$  **then**  $x := x - y$ ; **else**  $y := y - x$ ; **fi**  
 $\{x > 0 \ \& \ y > 0 \ \& \ \text{gcd}(x, y, z)\}$

$$R_{\text{if}}: \frac{\{\varphi\} \text{if } C \text{ then } \pi_1 \text{ else } \pi_2 \text{ fi} \{\psi\}}{\{\varphi \ \& \ C\} \pi_1 \{\psi\}, \{\varphi \ \& \ \neg C\} \pi_2 \{\psi\}}$$

# Логика Хоара

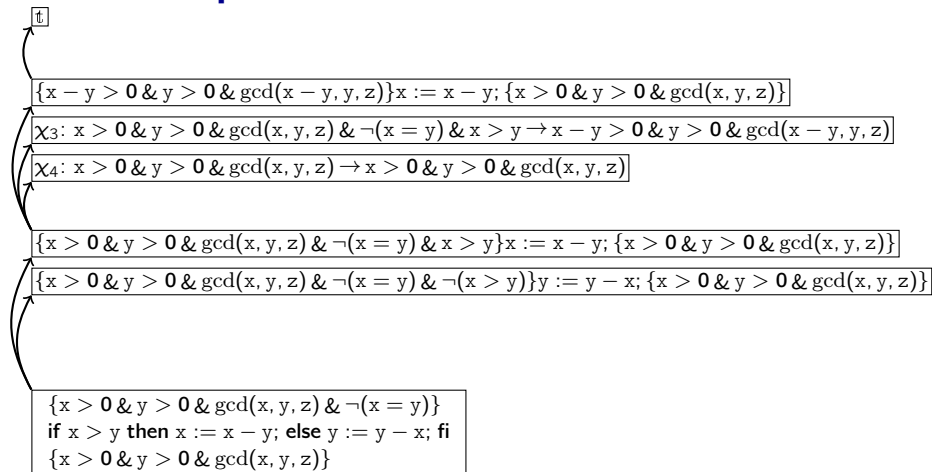


$$R_{inf}: \frac{\{\varphi\} \pi \{\psi\}}{\varphi \rightarrow \varphi', \{\varphi'\} \pi \{\psi'\}, \psi' \rightarrow \psi}$$

$$\mathcal{I} \models \chi_3$$

$$\mathcal{I} \models \chi_4$$

# Логика Хоара



$$R ::= \frac{\{\varphi\{x/t\}\} x := t; \{\varphi\}}{\vdash}$$



# Логика Хоара

 $\top$  $\{x - y > 0 \& y > 0 \& \text{gcd}(x - y, y, z)\} x := x - y; \{x > 0 \& y > 0 \& \text{gcd}(x, y, z)\}$  $\chi_3: x > 0 \& y > 0 \& \text{gcd}(x, y, z) \& \neg(x = y) \& x > y \rightarrow x - y > 0 \& y > 0 \& \text{gcd}(x - y, y, z)$  $\chi_4: x > 0 \& y > 0 \& \text{gcd}(x, y, z) \rightarrow x > 0 \& y > 0 \& \text{gcd}(x, y, z)$  $\{x > 0 \& y > 0 \& \text{gcd}(x, y, z) \& \neg(x = y) \& x > y\} x := x - y; \{x > 0 \& y > 0 \& \text{gcd}(x, y, z)\}$  $\{x > 0 \& y > 0 \& \text{gcd}(x, y, z) \& \neg(x = y) \& \neg(x > y)\} y := y - x; \{x > 0 \& y > 0 \& \text{gcd}(x, y, z)\}$  $\chi_5: x > 0 \& y > 0 \& \text{gcd}(x, y, z) \& \neg(x = y) \& \neg(x > y) \rightarrow$   
 $x > 0 \& y - x > 0 \& \text{gcd}(x, y - x, z)$  $\chi_6: x > 0 \& y > 0 \& \text{gcd}(x, y, z) \rightarrow x > 0 \& y > 0 \& \text{gcd}(x, y, z)$  $\{x > 0 \& y - x > 0 \& \text{gcd}(x, y - x, z)\} y := y - x; \{x > 0 \& y > 0 \& \text{gcd}(x, y, z)\}$  $\top$ 
$$R ::= \frac{\{\varphi\{x/t\}\} x := t; \{\varphi\}}{\top}$$

# Логика Хоара

$\top$

$\{x - y > 0 \& y > 0 \& \text{gcd}(x - y, y, z)\} x := x - y; \{x > 0 \& y > 0 \& \text{gcd}(x, y, z)\}$

$\chi_3: x > 0 \& y > 0 \& \text{gcd}(x, y, z) \& \neg(x = y) \& x > y \rightarrow x - y > 0 \& y > 0 \& \text{gcd}(x - y, y, z)$

$\chi_4: x > 0 \& y > 0 \& \text{gcd}(x, y, z) \rightarrow x > 0 \& y > 0 \& \text{gcd}(x, y, z)$

$\{x > 0 \& y > 0 \& \text{gcd}(x, y, z) \& \neg(x = y) \& x > y\} x := x - y; \{x > 0 \& y > 0 \& \text{gcd}(x, y, z)\}$

$\{x > 0 \& y > 0 \& \text{gcd}(x, y, z) \& \neg(x = y) \& \neg(x > y)\} y := y - x; \{x > 0 \& y > 0 \& \text{gcd}(x, y, z)\}$

$\chi_5: x > 0 \& y > 0 \& \text{gcd}(x, y, z) \& \neg(x = y) \& \neg(x > y) \rightarrow$   
 $x > 0 \& y - x > 0 \& \text{gcd}(x, y - x, z)$

$\chi_6: x > 0 \& y > 0 \& \text{gcd}(x, y, z) \rightarrow x > 0 \& y > 0 \& \text{gcd}(x, y, z)$

$\{x > 0 \& y - x > 0 \& \text{gcd}(x, y - x, z)\} y := y - x; \{x > 0 \& y > 0 \& \text{gcd}(x, y, z)\}$

$\top$

Частичная корректность программы доказана

# Полнота логики Хоара

Согласно теореме о корректности логики Хоара, правильность программы можно обосновать, построив успешный табличный вывод подходящего триплета Хоара

А правда ли, что корректность **любой** правильной программы может быть обоснована построением успешного табличного вывода подходящего триплета?

На самом деле это не один вопрос, а два принципиально разных:

1. Все ли свойства правильности программ могут быть записаны в виде формул логики предикатов?
2. Для любого ли истинного триплета существует успешный вывод?



# Полнота логики Хоара

Ответ на оба вопроса существенно зависит от **сигнатуры**, в которой записываются предусловия и постусловия

Если эта сигнатура слишком «скудная», то:

- ▶ Её может быть недостаточно для записи свойств правильности программы
- ▶ При применении правил

$$R_{seq} : \frac{\{\varphi\}\pi_1 \pi_2\{\psi\}}{\{\varphi\}\pi_1\{\chi\}, \{\chi\}\pi_2\{\psi\}} \quad R_{inf} : \frac{\{\varphi\}\pi\{\psi\}}{\varphi \rightarrow \varphi', \{\varphi'\}\pi\{\psi'\}, \psi' \rightarrow \psi}$$

может и не найтись подходящих формул  $\varphi'$ ,  $\psi'$ ,  $\chi$ , позволяющих достроить вывод до успешного

При этом чем «богаче» сигнатура, тем труднее анализировать истинность формул и подбирать «подходящие» формулы при построении вывода

# Автоматизация проверки правильности программ

А если сигнатура достаточно «богата», то можно ли реализовать программу, автоматически доказывающую корректность программ?

Как и в вопросе про полноту, это на самом деле не один вопрос, а два:

1. Можно ли автоматизировать запись триплетов Хоара, отвечающих свойствам правильности программ?
2. Можно ли автоматизировать построение успешного вывода для заданного триплета Хоара?

Ответ на первый вопрос однозначен — **нет**: один из главных недостатков формальной верификации состоит в том, что формальная спецификация программы, как правило, создаётся вручную специально обученным экспертом

# Автоматизация проверки правильности программ

С автоматизацией построения успешного вывода для заданного триплета Хоара дела обстоят чуть лучше

**Слабейшим предусловием** для программы  $\pi$  и постусловия  $\psi$  в интерпретации  $\mathcal{I}$  называется формула  $wpr(\pi, \psi, \mathcal{I})$ , такая что

- ▶  $\mathcal{I} \models \{wpr(\pi, \psi, \mathcal{I})\}\pi\{\psi\}$  и
- ▶ для любой формулы  $\varphi$ , такой что  $\mathcal{I} \models \{\varphi\}\pi\{\psi\}$ , верно соотношение  $\mathcal{I} \models \varphi \rightarrow wpr(\pi, \psi, \mathcal{I})$

Слабейших предусловий на самом деле может быть много, но ими реализуются одинаковые отношения, так что для простоты иногда будем считать, что оно единственно

**Теорема.**  $\mathcal{I} \models \{\varphi\}\pi\{\psi\} \Leftrightarrow \mathcal{I} \models \{wpr(\pi, \psi, \mathcal{I})\}\pi\{\psi\}$  и  $\mathcal{I} \models \varphi \rightarrow wpr(\pi, \psi, \mathcal{I})$

**Доказательство.** Следует из определения слабейшего предусловия



# Автоматизация проверки правильности программ

## Теорема (о слабейшем предусловии)

- ▶  $wpr(\emptyset, \psi, \mathcal{I}) = \psi$
- ▶  $wpr(x := t; , \psi, \mathcal{I}) = \psi\{x/t\}$ ,  
если подстановка  $\{x/t\}$  правильна для  $\psi$
- ▶  $wpr(\text{if } C \text{ then } \pi_1 \text{ else } \pi_2 \text{ fi}, \psi, \mathcal{I}) =$   
 $C \ \& \ wpr(\pi_1, \psi, \mathcal{I}) \vee \neg C \ \& \ wpr(\pi_2, \psi, \mathcal{I})$
- ▶  $wpr(\pi_1 \ \pi_2, \psi, \mathcal{I}) = wpr(\pi_1, wpr(\pi_2, \psi, \mathcal{I}), \mathcal{I})$

Доказательство. Можете попробовать сами при желании, не будем тратить на это время

Таким образом,

- ▶ проверка корректности программы сводится к вычислению слабейшего предусловия и проверки истинности заданной формулы
- ▶ для программ **без циклов** слабейшее предусловие не зависит от выбора интерпретации и вычисляется очень просто

# Автоматизация проверки правильности программ

А как вычислить слабейшее предусловие для цикла?

Устройство слабейшего предусловия тесно связано с устройством правил доказательства корректности программ: вычисление этого предусловия — настолько же (не)простая задача, насколько и применение правила для построения успешного вывода

Чтобы применить правило

$$R_{\text{while}} : \frac{\{\varphi\} \mathbf{while} C \mathbf{do} \pi \mathbf{od} \{\varphi \ \& \ \neg C\}}{\{\varphi \ \& \ C\} \pi \{\varphi\}},$$

требуется предварительно найти формулу  $\varphi$ , реализующую свойство состояний данных, сохраняющееся при выполнении каждого витка цикла

Эта формула  $\varphi$  называется **инвариантом цикла** и явно или неявно используется практически всегда при анализе циклов

Автоматическая генерация инвариантов циклов — это **ключевая проблема** автоматизации проверки правильности программ