Математическая логика

mk.cs.msu.ru → Лекционные курсы → Математическая логика (318, 319/2, 241, 242)

Блок 46

Императивные программы
Формальная верификация программ
Логика Хоара

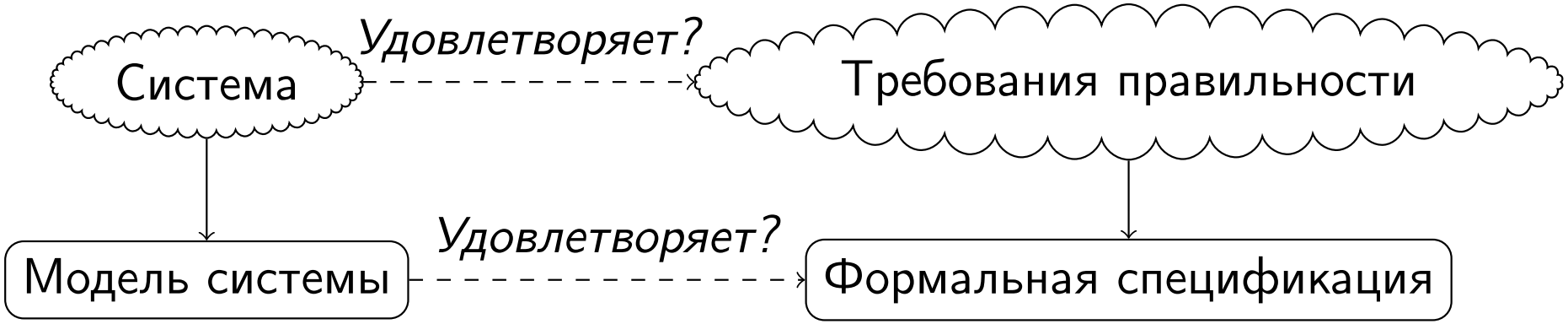
Лектор:
Подымов Владислав Васильевич

E-mail:
valdus@yandex.ru

ВМК МГУ, 2020/2021, весенний 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:

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

Здесь:

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

Императивные программы: синтаксис

В примерах используется арифметическая сигнатура, в которой, в частности, содержатся

- константы 0, 1
- функциональные символы $-^{(2)}$, $.^{(2)}$
- предикатные символы $=^{(2)}$, $>^{(2)}$

Пример: реализация алгоритма Эвклида вычисления наибольшего общего делителя чисел в переменных x, y

while $\neg(\mathrm{x} = \mathrm{y})$ **do**
  if $\mathrm{x} > \mathrm{y}$ **then**
    $\mathrm{x} := \mathrm{x} - \mathrm{y};$
  else
    $\mathrm{y} := \mathrm{y} - \mathrm{x};$
  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 $\widetilde{\Sigma}$:

- $\langle \mathrm{x} := t; \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \emptyset \mid \sigma\{x \leftarrow \overline{t}\} \rangle$
- $\langle \textbf{if } C \textbf{ then } \pi_1 \textbf{ else } \pi_2 \textbf{ fi} \mid \sigma \rangle \xrightarrow{\mathcal{I}} \langle \pi_1 \mid \sigma \rangle$, 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

Иными словами, если $\overset{\mathcal{I}}{\Rightarrow}$ — *транзитивное замыкание* отношения $\xrightarrow{\mathcal{I}}$, то σ' — результат вычисления программы π из состояния данных σ в интерпретации $\mathcal{I}$, если $\langle \pi \mid \sigma \rangle \overset{\mathcal{I}}{\Rightarrow} \langle \emptyset \mid \sigma' \rangle$

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

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

Императивные программы: операционная семантика

Пример

$$\mathrm{Var} = \{\mathrm{x}\} \qquad \sigma = \langle\{0,1\},\{-\},\{>\}\rangle \qquad \mathcal{I} = Ar[0,1;-;>]$$

$$\pi\text{: } \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \qquad \sigma = [\mathrm{x}/1]$$

Вычисление π из σ в $\mathcal{I}$:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$

Пояснение:

$$\mathcal{I} \models (\mathrm{x} > 0)[\mathrm{x}/1]$$

Императивные программы: операционная семантика

Пример

$$\mathrm{Var} = \{\mathrm{x}\} \qquad \sigma = \langle\{0,1\},\{-\},\{>\}\rangle \qquad \mathcal{I} = Ar[0,1;-;>]$$

$$\pi\text{: } \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \qquad \sigma = [\mathrm{x}/1]$$

Вычисление π из σ в $\mathcal{I}$:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$

$$\downarrow\mathcal{I}$$

$$\langle \mathrm{x} := \mathrm{x} - 1; \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$

Пояснение:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \pi' \textbf{ od} \mid [\mathrm{x}/1]\rangle \xrightarrow{\mathcal{I}} \langle \pi' \textbf{ while } \mathrm{x} > 0 \textbf{ do } \pi' \textbf{ od} \mid [\mathrm{x}/1]\rangle$$

Императивные программы: операционная семантика

Пример

$$\mathrm{Var} = \{\mathrm{x}\} \qquad \sigma = \langle\{0,1\},\{-\},\{>\}\rangle \qquad \mathcal{I} = Ar[0,1;-;>]$$

$$\pi\text{: } \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \qquad \sigma = [\mathrm{x}/1]$$

Вычисление π из σ в $\mathcal{I}$:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$

$$\downarrow\mathcal{I}$$

$$\langle \mathrm{x} := \mathrm{x} - 1; \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$

Пояснение:

$$\langle \mathrm{x} := \mathrm{x} - 1; \mid [\mathrm{x}/1]\rangle \xrightarrow{\mathcal{I}} \langle \emptyset \mid [\mathrm{x}/1]\{\mathrm{x} \leftarrow 1 - 1\}\rangle = \langle \emptyset \mid [\mathrm{x}/0]\rangle$$

Императивные программы: операционная семантика

Пример

$$\mathrm{Var} = \{\mathrm{x}\} \qquad \sigma = \langle\{0,1\},\{-\},\{>\}\rangle \qquad \mathcal{I} = Ar[0,1;-;>]$$

$$\pi\text{: } \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \qquad \sigma = [\mathrm{x}/1]$$

Вычисление π из σ в $\mathcal{I}$:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$

$$\downarrow\mathcal{I}$$

$$\langle \mathrm{x} := \mathrm{x} - 1; \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$

$$\downarrow\mathcal{I}$$

$$\langle \emptyset \textbf{ while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/0]\rangle$$

Пояснение:

$$\langle \mathrm{x} := \mathrm{x} - 1; \pi \mid [\mathrm{x}/1]\rangle \xrightarrow{\mathcal{I}} \langle \emptyset\ \pi \mid [\mathrm{x}/0]\rangle$$

Императивные программы: операционная семантика

Пример

$$\mathrm{Var} = \{\mathrm{x}\} \qquad \sigma = \langle\{0,1\},\{-\},\{>\}\rangle \qquad \mathcal{I} = Ar[0,1;-;>]$$

$$\pi\text{: } \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \qquad \sigma = [\mathrm{x}/1]$$

Вычисление π из σ в $\mathcal{I}$:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \mathrm{x} := \mathrm{x} - 1; \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \emptyset \textbf{ while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/0]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/0]\rangle$$

Пояснение:

$$\langle \emptyset\ \pi \mid \sigma\rangle \xrightarrow{\mathcal{I}} \langle \pi \mid \sigma\rangle$$

Императивные программы: операционная семантика

Пример

$$\text{Var} = \{\mathrm{x}\} \qquad \sigma = \langle\{0,1\},\{-\},\{>\}\rangle \qquad \mathcal{I} = Ar[0,1;-;>]$$

$$\pi\text{: } \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \qquad \sigma = [\mathrm{x}/1]$$

Вычисление π из σ в $\mathcal{I}$:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \mathrm{x} := \mathrm{x} - 1; \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \emptyset \textbf{ while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/0]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/0]\rangle$$

Пояснение:

$$\mathcal{I} \not\models (\mathrm{x} > 0)[\mathrm{x}/0]$$

Императивные программы: операционная семантика

Пример

$$\mathrm{Var} = \{\mathrm{x}\} \qquad \sigma = \langle\{0,1\},\{-\},\{>\}\rangle \qquad \mathcal{I} = Ar[0,1;-;>]$$

$$\pi\text{: } \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \qquad \sigma = [\mathrm{x}/1]$$

Вычисление π из σ в $\mathcal{I}$:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \mathrm{x} := \mathrm{x} - 1; \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \emptyset \textbf{ while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/0]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/0]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \emptyset \mid [\mathrm{x}/0]\rangle$$

Пояснение:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \pi' \textbf{ od} \mid [\mathrm{x}/0]\rangle \xrightarrow{\mathcal{I}} \langle \emptyset \mid [\mathrm{x}/0]\rangle$$

Императивные программы: операционная семантика

Пример

$$\mathrm{Var} = \{\mathrm{x}\} \qquad \sigma = \langle\{0,1\},\{-\},\{>\}\rangle \qquad \mathcal{I} = Ar[0,1;-;>]$$

$$\pi\text{: } \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \qquad \sigma = [\mathrm{x}/1]$$

Вычисление π из σ в $\mathcal{I}$:

$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \mathrm{x} := \mathrm{x} - 1; \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/1]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \emptyset \textbf{ while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/0]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \textbf{while } \mathrm{x} > 0 \textbf{ do } \mathrm{x} := \mathrm{x} - 1; \textbf{ od} \mid [\mathrm{x}/0]\rangle$$
$$\downarrow\mathcal{I}$$
$$\langle \emptyset \mid [\mathrm{x}/0]\rangle$$

Пояснение:

$[\mathrm{x}/0]$ — результат вычисления

Задача верификации программ

Требования правильности выполнения программы могут быть записаны в виде двух 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 $\{\varphi\}\,\pi\,\{\psi\}$ выполним 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 $\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$

Логика Хоара

Для проверки проверки частичной корректности программ можно адаптировать *метод семантических таблиц*: определить понятие *вывода* (дерева, построенного согласно *правилам вывода*), и свести проверку частичной корректности программы к построению особого (*успешного*) вывода

Правила вывода будут выглядеть так:[1]

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

где Φ, Ψ, Ω — триплеты Хоара и φ, ψ — формулы логики предикатов

Содержательное прочтение:

если в $\mathcal{I}$ выполнимы все триплеты под чертой
и истинны все формулы под чертой,
то в $\mathcal{I}$ выполним триплет Φ

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

Логика Хоара

Вот эти правила:

$$R_{\emptyset}: \frac{\{\varphi\}\,\emptyset\,\{\varphi\}}{\mathrm{tt}} \qquad R_{:=}: \frac{\{\varphi\,\{\mathtt{x}/t\}\}\,\mathtt{x} := t;\{\varphi\}}{\mathrm{tt}}$$

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

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

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

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

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

Логика Хоара

Лемма(*о корректности правил вывода Хоара*). Для любой интерпретации $\mathcal{I}$ и любого правила вывода логики Хоара

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

верно следующее:
если $\mathcal{I} \models \Psi$, $\mathcal{I} \models \Omega$ и формулы φ, ψ истинны в $\mathcal{I}$, то $\mathcal{I} \models \Phi$

Доказательство.

Подробно рассмотрим только правило $R_{:=}$: $\dfrac{\{\varphi\{\mathtt{x}/t\}\}\,\mathtt{x} := t;\{\varphi\}}{\mathtt{t}}$

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

Шаг вычисления для программы $\mathtt{x} := t;$ устроен так:

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

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

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

Корректность остальных правил можете попробовать обосновать самостоятельно

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 так же, как и в дереве табличного вывода для логики предикатов
- 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 программа π частично корректна в $\mathcal{I}$ относительно предусловия φ и постусловия ψ

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 выполнимость триплета $\{\varphi\}\,\pi\,\{\psi\}$ в $\mathcal{I}$

Значит, по *определению частичной корректности*, программа π частично корректна в $\mathcal{I}$ относительно φ и ψ ▼

Логика Хоара

Пример: алгоритм Эвклида.

Рассмотрим такую программу π:

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**

Докажем, что в результате выполнения π в "естественной" арифметической интерпретации с целыми числами в $\mathtt{x}$ записывается **наибольший общий делитель** (НОД) положительных чисел, заданных в $\mathtt{x}$ и $\mathtt{y}$ перед выполнением

Предусловие φ: числа $\mathtt{x}$ и $\mathtt{y}$ положительны, и $\mathtt{z}$ — переменная, обозначающая НОД $\mathtt{x}$ и $\mathtt{y}$ в начале выполнения программы

$$\begin{array}{ll} \mathtt{u}|\mathtt{v}: & \exists \mathtt{w}\,(\mathtt{v} = \mathtt{u}\cdot\mathtt{w}) \\ \gcd(\mathtt{u},\mathtt{v},\mathtt{w}): & (\mathtt{w}|\mathtt{u})\,\&\,(\mathtt{w}|\mathtt{v})\,\&\,\forall \mathtt{r}\,((\mathtt{r}|\mathtt{u})\,\&\,(\mathtt{r}|\mathtt{v})\to(\mathtt{r}\le\mathtt{w})) \\ \varphi: & \mathtt{x} > 0\,\&\,\mathtt{y} > 0\,\&\,\gcd(\mathtt{x},\mathtt{y},\mathtt{z}) \end{array}$$

Постусловие ψ: в $\mathtt{x}$ записан требуемый НОД

$$\psi\colon \mathtt{x} = \mathtt{z}$$

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

Логика Хоара

$\{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z})\}$
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**
$\{\mathtt{x} = \mathtt{z}\}$

χ_1: $\mathtt{x} > \mathbf{0} \,\&\, \mathtt{x} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z}) \to \mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z})$

χ_2: $\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z}) \,\&\, \neg\neg(\mathtt{x} = \mathtt{y}) \to \mathtt{x} = \mathtt{z}$

$\{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z})\}$
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**
$\{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z}) \,\&\, \neg\neg(\mathtt{x} = \mathtt{y})\}$

$$R_{inf}\colon \frac{\{\varphi\}\,\pi\,\{\psi\}}{\varphi \to \varphi',\ \{\varphi'\}\,\pi\,\{\psi'\},\ \psi' \to \psi}$$

χ_1 истинна в $\mathcal{I}$
χ_2 истинна в $\mathcal{I}$

Логика Хоара

$\{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z})\}$
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**
$\{\mathtt{x} = \mathtt{z}\}$

χ_1: $\mathtt{x} > \mathbf{0} \,\&\, \mathtt{x} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z}) \to \mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z})$

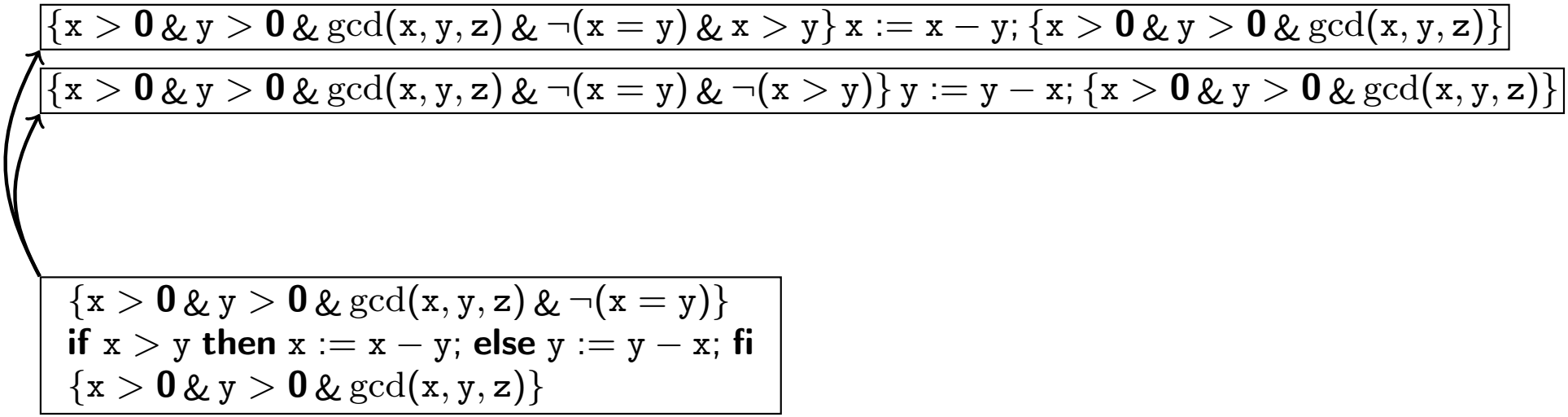
χ_2: $\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z}) \,\&\, \neg\neg(\mathtt{x} = \mathtt{y}) \to \mathtt{x} = \mathtt{z}$

$\{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z})\}$
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**
$\{\mathtt{x} > \mathbf{0} \,\&\, \mathtt{y} > \mathbf{0} \,\&\, \gcd(\mathtt{x}, \mathtt{y}, \mathtt{z}) \,\&\, \neg\neg(\mathtt{x} = \mathtt{y})\}$

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

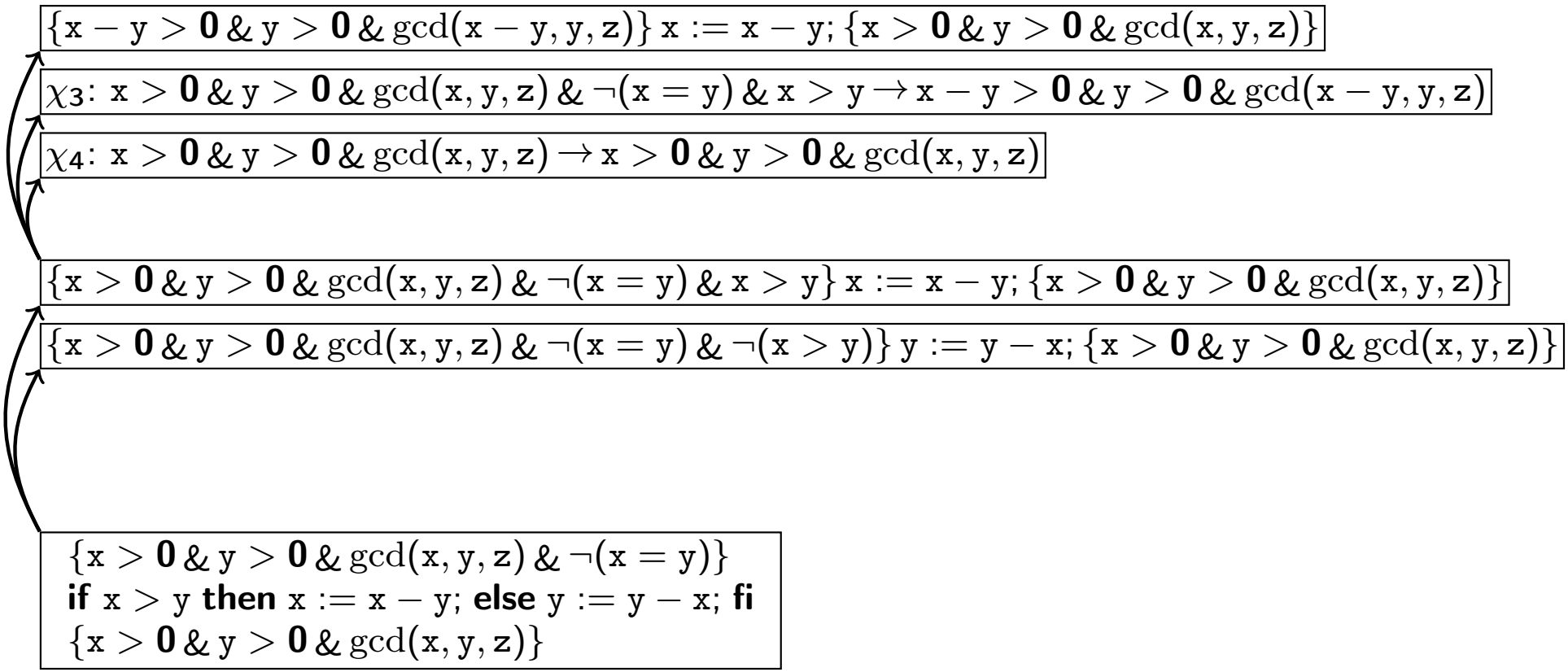
$$R_{\textbf{while}}\colon \frac{\{\varphi\}\ \textbf{while}\ C\ \textbf{do}\ \pi\ \textbf{od}\ \{\varphi \,\&\, \neg C\}}{\{\varphi \,\&\, C\}\ \pi\ \{\varphi\}}$$

Логика Хоара



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

Логика Хоара

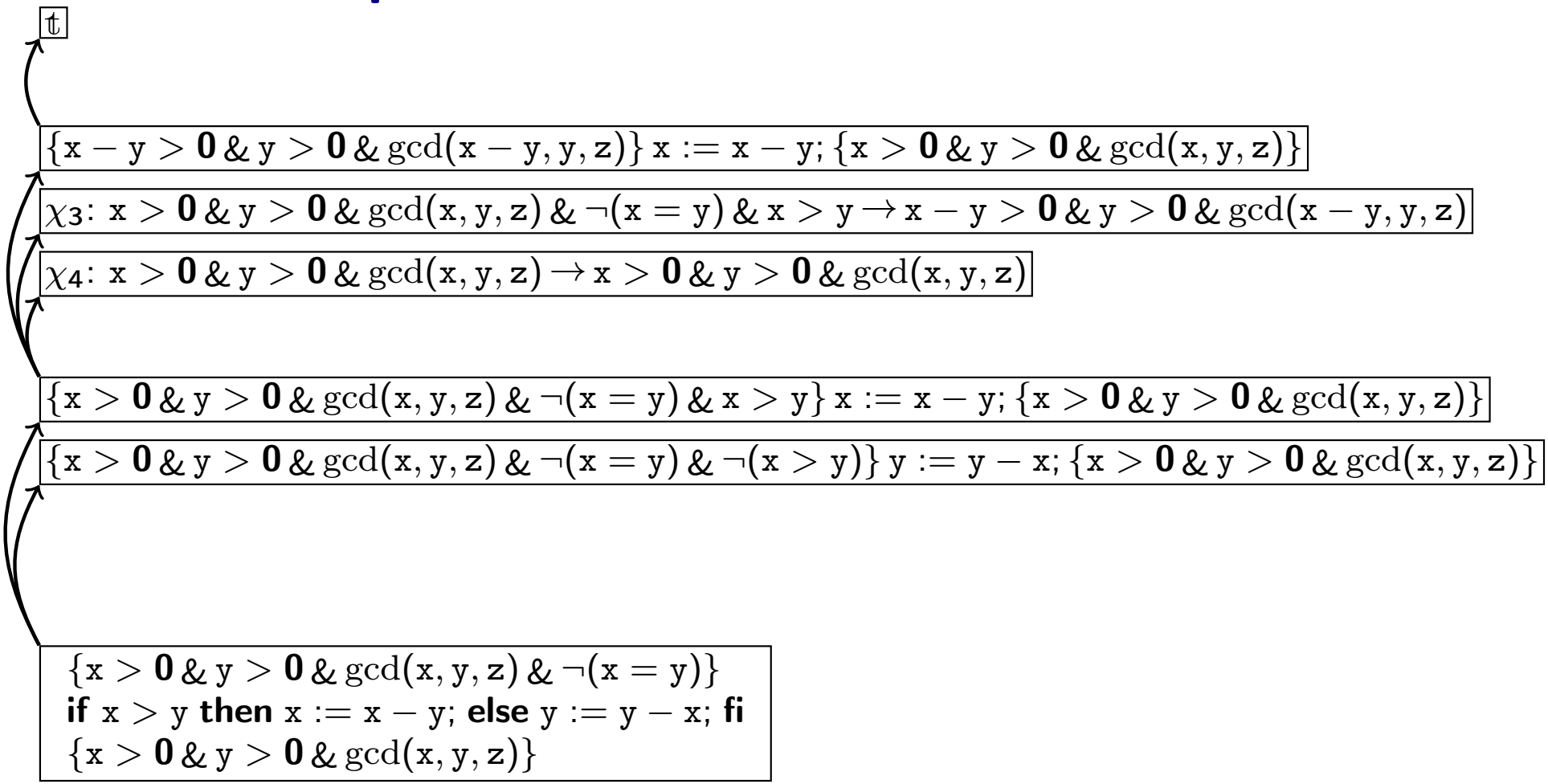


$$R_{inf}\colon \frac{\{\varphi\}\,\pi\,\{\psi\}}{\varphi \to \varphi',\ \{\varphi'\}\,\pi\,\{\psi'\},\ \psi' \to \psi}$$

χ_3 истинна в $\mathcal{I}$

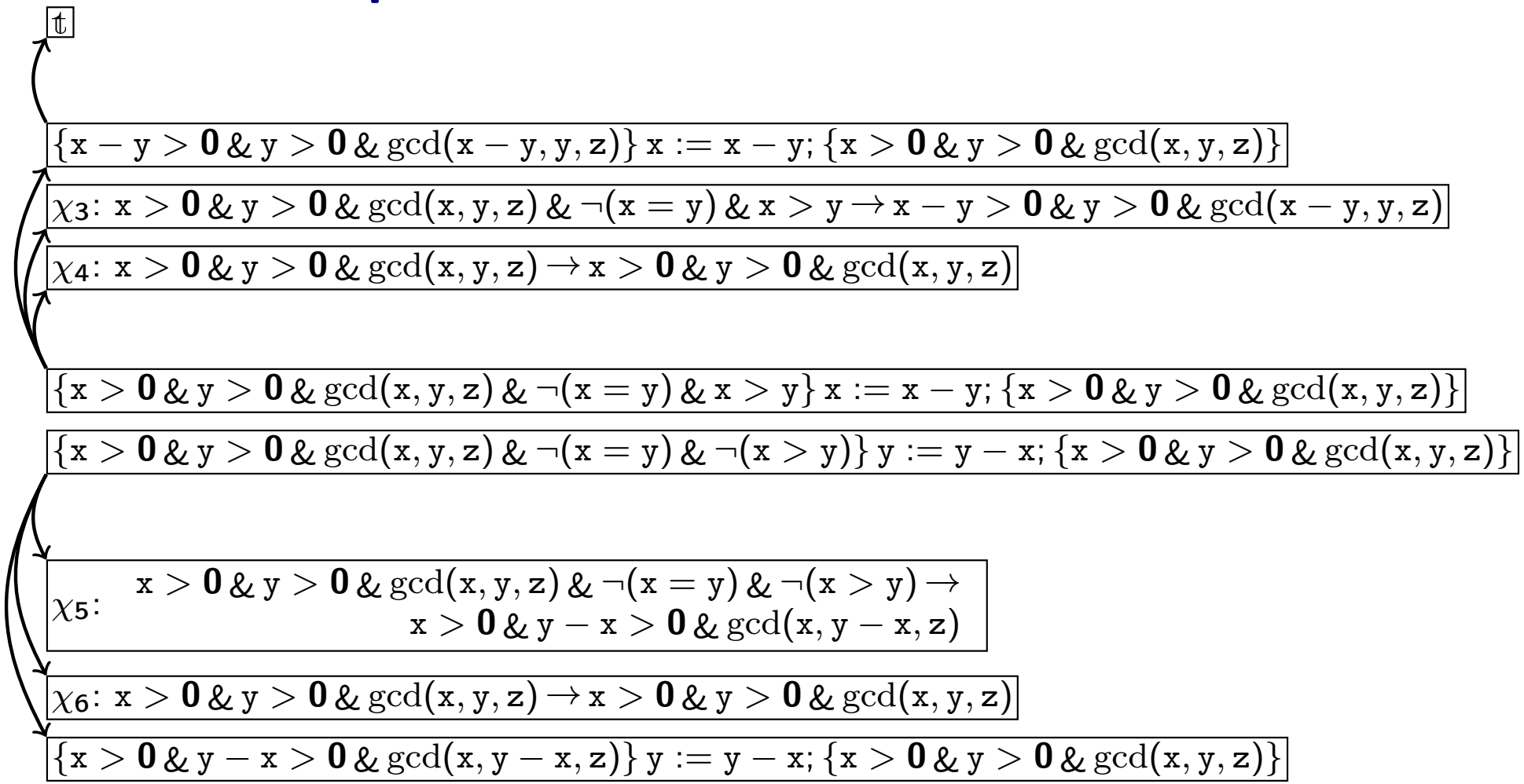
χ_4 истинна в $\mathcal{I}$

Логика Хоара



$$R_{:=}:\ \frac{\{\varphi\,\{\mathtt{x}/t\}\}\,\mathtt{x}:=t;\{\varphi\}}{\mathtt{t}}$$

Логика Хоара

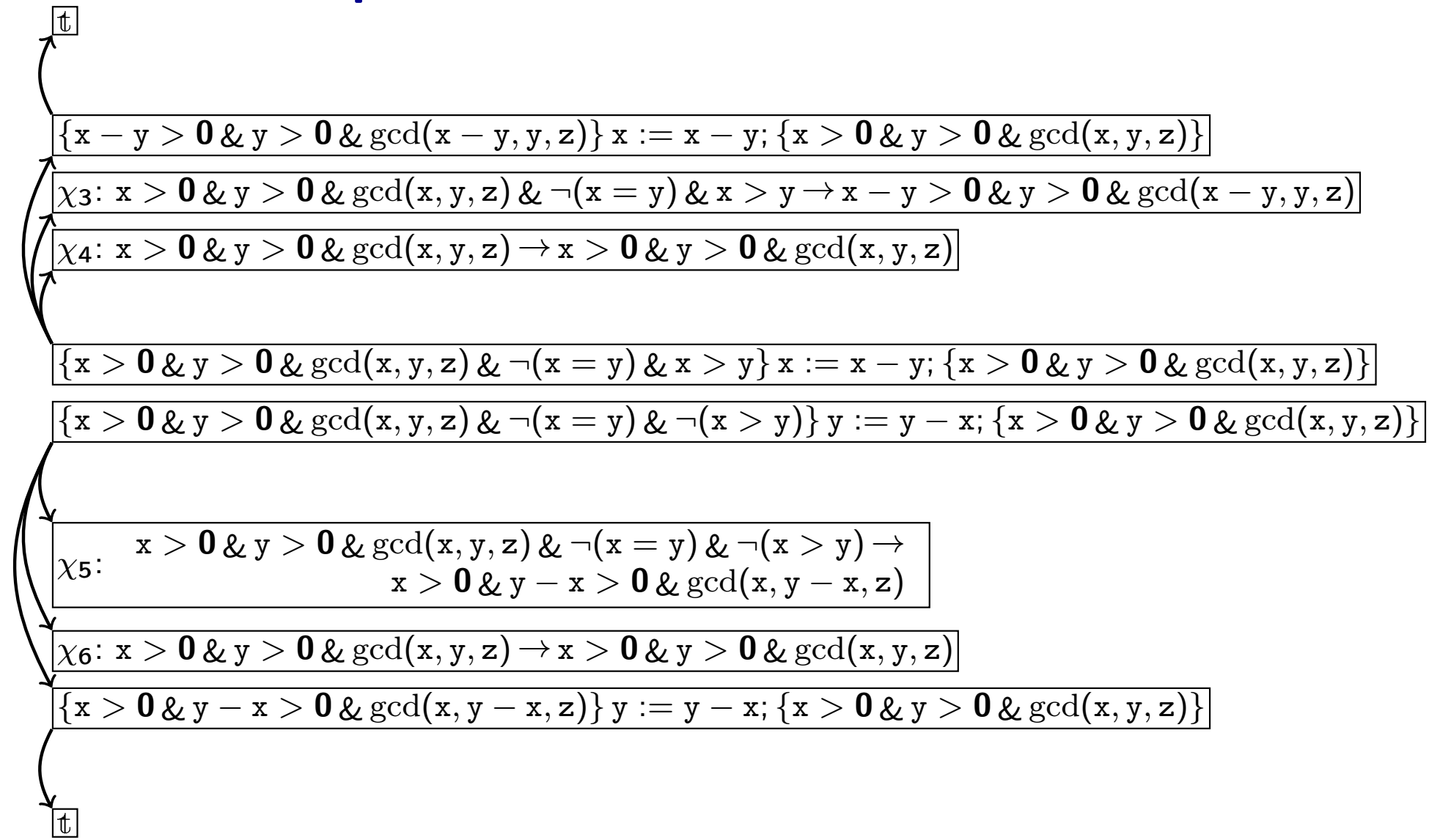


R_{inf}: $\dfrac{\{\varphi\}\,\pi\,\{\psi\}}{\varphi\to\varphi',\{\varphi'\}\,\pi\,\{\psi'\},\psi'\to\psi}$

χ_5 истинна в $\mathcal{I}$

χ_6 истинна в $\mathcal{I}$

Логика Хоара



$$R_{:=}\colon \frac{\{\varphi\{x/t\}\}\, x := t; \{\varphi\}}{\top}$$

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 выполнимого 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
  Например, как записать свойство "$x = y \cdot z$", если в сигнатуру включена только операция сложения чисел?
- при применении правил
  $$R_{seq} : \frac{\{\varphi\}\,\pi_1\;\pi_2\,\{\psi\}}{\{\varphi\}\,\pi_1\,\{\chi\}\,,\{\chi\}\,\pi_2\,\{\psi\}} \qquad R_{inf} : \frac{\{\varphi\}\,\pi\,\{\psi\}}{\varphi \to \varphi', \{\varphi'\}\,\pi\,\{\psi'\}\,, \psi' \to \psi}$$
  может и не найтись подходящих формул φ', ψ', χ, 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 построение триплетов Хоара, описывающих свойства правильности программ?
2. Можно ли автоматизировать построение успешного вывода для заданных триплетов 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 π и постусловия ψ 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 φ, такой что $\mathcal{I} \models \{\varphi\}\,\pi\,\{\psi\}$, верно соотношение $\mathcal{I} \models \varphi \to wpr(\pi, \psi, \mathcal{I})$

Замечание: слабейших предусловий на самом деле бывает много, но все они *равносильны* в интерпретации, 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** $\mathcal{I} \models \varphi \to wpr(\pi, \psi, \mathcal{I})$

Доказательство. Следует из определения слабейшего предусловия ▼

Автоматизация проверки правильности программ

Теорема(*о слабейшем предусловии*).

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

Доказательство. Попробуйте самостоятельно

Таким образом,

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

Автоматизация проверки правильности программ

А как вычислить слабейшее предусловие для цикла?

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

Чтобы применить правило

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

требуется предварительно найти формулу φ, в которой записано свойство состояний данных, сохраняющееся при выполнении каждого витка цикла

Эта формула φ называется инвариантом цикла и явно или неявно используется практически всегда при анализе циклов

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