Основы математической логики и логического программирования

ЛЕКТОР: В.А. Захаров

Лекция 20.

Правильные программы. Императивные программы. Задача верификации программ. Логика Хоара. Автоматическая проверка правильности программ.

ПРАВИЛЬНЫЕ ПРОГРАММЫ

Какая компьютерная программа считается хорошей?

Та, которая работает ПРАВИЛЬНО и **эффективно** .

А какая программа считается правильной?

Правильной считается та программа, которая выполняет в точности то, что от нее требуется.

А как убедиться, что программа выполняет то, что от нее требуется?

Для этого нужно

1. Описать строго (формально) требования правильности вычислений;
2. Проверить, что все вычисления программы удовлетворяют этим требованиям.

ПРАВИЛЬНЫЕ ПРОГРАММЫ

Описание требований правильности функционирования программы называется спецификацией программы.

Проверка соблюдения вычислениями программы требований правильности функционирования называется верификацией программы.

Если спецификации программ записать на формальном логическом языке и строго определить операционную семантику программ, то для доказательства правильности программ можно использовать методы математической логики (логический вывод).

ПРАВИЛЬНЫЕ ПРОГРАММЫ

Формальная верификация программ

Преимущества	Проблемы
1. Абсолютно точная проверка правильности программ.	1. Как заставить программистов писать формальные спецификации?
2. Возможность автоматизации построения логического вывода.	2. Как заставить прувер работать эффективно?

И, тем не менее, попробуем...

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

Определим синтаксис и семантику императивных программ.

Пусть задана сигнатура $\sigma = \langle Const, Func, Pred \rangle$, в которой определно множество термов *Term* и множество атомарных формул *Atom*.
Условимся, что $\Leftarrow$ — это служебный символ, не принадлежащий сигнатуре σ.

Определение

присваивание ::= «переменная» $\Leftarrow$ «терм»
условие ::= «атом» | ($\neg$условие) |
(условие & условие) | (условие $\vee$ условие)
программа ::= присваивание |
программа ; программа |
if «условие» **then** программа **else** программа **fi** |
while условие **do** программа **od**

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

Пример

Программа вычисления наибольшего общего делителя двух натуральных чисел.
$Const = \{0, 1, 2, \dots\}$,
$Func = \{+^{(2)}, -^{(2)}\}$,
$Pred = \{=^{(2)}, >^{(2)}, <^{(2)}\}$

while $\neg(x = y)$
    do
        if $x > y$
            then $x \Leftarrow x - y$
            else $y \Leftarrow y - x$
        fi
    od

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

Операционная семантика императивных программ

Семантика задает смысл (значение) синтаксических конструкций (слов, формул, программ и пр.).

Значением императивной программы является **отношение вход–выход** между входными данными и результатом вычисления.

Отношение вход–выход программы определяется при помощи отношения переходов между **состояниями вычисления** программы.

Состояние вычисления программы определяется двумя компонентами — **состоянием управления** и **состоянием данных** .

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

Определение (состояния вычисления)

Пусть **Var** — это множество переменных, а *GTerm* — это множество основных термов сигнатуры σ.

Оценкой переменных (состоянием данных) будем называть всякое отображение (подстановку) $\theta : \textbf{Var} \rightarrow \mathit{GTerm}$.

Состоянием управления будем называть всякую программу, а также специальный символ $\varnothing$.

Состоянием вычисления будем называть всякую пару $\langle \pi, \theta \rangle$, где π — состояние управления, а θ — оценка переменных.

Запись State_σ будет обозначать множество всевозможных состояний вычислений сигнатуры σ.

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

Определение (отношения переходов)

Пусть I — это интерпретация сигнатуры σ.

Тогда отношение переходов для императивных программ — это бинарное отношение $\longrightarrow_I$ на множестве состояний вычисления $State_\sigma$, удовлетворяющее следующим требованиям:

ASS: $\langle x \Rightarrow t,\ \theta\rangle \ \longrightarrow_I \ \langle \varnothing,\ \{x/t\}\theta\rangle$;

COMP_$\varnothing$: $\langle \pi_1; \pi_2,\ \theta\rangle \ \longrightarrow_I \ \langle \pi_2,\ \eta\rangle$
тогда и только тогда, когда $\langle \pi_1,\ \theta\rangle \ \longrightarrow_I \ \langle \varnothing,\ \eta\rangle$;

COMP: $\langle \pi_1; \pi_2,\ \theta\rangle \ \longrightarrow_I \ \langle \pi_1'; \pi_2,\ \eta\rangle$
тогда и только тогда, когда $\langle \pi_1,\ \theta\rangle \ \longrightarrow_I \ \langle \pi_1',\ \eta\rangle$ и $\pi_1' \neq \varnothing$;

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

Определение (отношения переходов)

IF_1: $\langle \textbf{if } C \textbf{ then } \pi_1 \textbf{ else } \pi_2 \textbf{ fi},\ \theta\rangle \ \longrightarrow_I\ \langle \pi_1,\ \theta\rangle$
тогда и только тогда, когда $I \models C\theta$;

IF_0: $\langle \textbf{if } C \textbf{ then } \pi_1 \textbf{ else } \pi_2 \textbf{ fi},\ \theta\rangle \ \longrightarrow_I\ \langle \pi_2,\ \theta\rangle$
тогда и только тогда, когда $I \not\models C\theta$;

WHILE_1: $\langle \textbf{while } C \textbf{ do } \pi \textbf{ od},\ \theta\rangle \ \longrightarrow_I\ \langle \pi;\ \textbf{while } C \textbf{ do } \pi \textbf{ od},\ \theta\rangle$
тогда и только тогда, когда $I \models C\theta$;

WHILE_0: $\langle \textbf{while } C \textbf{ do } \pi \textbf{ od},\ \theta\rangle \ \longrightarrow_I\ \langle \varnothing,\ \theta\rangle$
тогда и только тогда, когда $I \not\models C\theta$.

Отношение переходов $\longrightarrow_I$ определяет, как изменяется состояние вычисления за один шаг работы интерпретатора императивных программ.

Императивные программы

Определение (вычисления программы)

Пусть π_0 — это императивная программа, θ_0 — оценка переменных.

Частичным вычислением программы π_0 на оценке переменных θ_0 в интерпретации I называется последовательность (конечная или бесконечная) состояний вычисления

$$\langle \pi_0, \theta_0 \rangle, \langle \pi_1, \theta_1 \rangle, \dots, \langle \pi_{n-1}, \theta_{n-1} \rangle, \langle \pi_n, \theta_n \rangle, \dots,$$

в которой для любого n, $n \geq 1$, выполняется отношение $\langle \pi_{n-1}, \theta_{n-1} \rangle \longrightarrow_I \langle \pi_n, \theta_n \rangle$.

Вычислением программы π_0 на оценке переменных θ_0 в интерпретации I называется всякое частичное вычисление, которое нельзя продолжить.

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

Пример

Пусть I — интерпретация сигнатуры $\sigma = \langle Const, Func, Pred\rangle$:
$Const = \{0, 1, 2, \dots, \}$, $Func = \{+^{(2)}, -^{(2)}\}$,
$Pred = \{=^{(2)}, >^{(2)}, <^{(2)}\}$,
предметной областью которой является множество натуральных чисел $\mathcal{N}_0$ с обычными арифметическими операциями и отношениями.

Рассмотрим вычисление программы

π_0: **while** $\neg(x = y)$
do if $x > y$ **then** $x \Leftarrow x - y$ **else** $y \Leftarrow y - x$ **fi od**

на оценке переменных $\theta_0 = \{x/4, y/6\}$.

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

π_0: **while** $\neg(x = y)$
do if $x > y$ **then** $x \Leftarrow x - y$ **else** $y \Leftarrow y - x$ **fi od**

$\theta_0 = \{x/4, y/6\}$

Пример

$$\langle \pi_0,\ \{x/4, y/6\}\rangle$$
$$\downarrow_I$$
$$\langle \textbf{if } x > y \textbf{ then } x \Leftarrow x - y \textbf{ else } y \Leftarrow y - x \textbf{ fi } ; \pi_0,\ \{x/4, y/6\}\rangle$$
$$\downarrow_I$$
$$\langle y \Leftarrow y - x; \pi_0,\ \{x/4, y/6\}\rangle$$
$$\downarrow_I$$
$$\langle \pi_0,\ \{x/4, y/6 - 4\}\rangle$$

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

π_0: **while** $\neg(x = y)$

do if $x > y$ **then** $x \Leftarrow x - y$ **else** $y \Leftarrow y - x$ **fi od**

$\theta_0 = \{x/4, y/6\}$

Пример

$$\langle \pi_0,\ \{x/4, y/6-4\}\rangle$$
$$\downarrow_I$$
$$\langle \textbf{if } x > y \textbf{ then } x \Leftarrow x - y \textbf{ else } y \Leftarrow y - x \textbf{ fi }; \pi_0,\ \{x/4, y/6-4\}\rangle$$
$$\downarrow_I$$
$$\langle x \Leftarrow x - y; \pi_0,\ \{x/4, y/6-4\}\rangle$$
$$\downarrow_I$$
$$\langle \pi_0,\ \{x/4-(6-4), y/6-4\}\rangle$$
$$\downarrow_I$$
$$\langle \varnothing,\ \{x/4-(6-4), y/6-4\}\rangle$$

ИМПЕРАТИВНЫЕ ПРОГРАММЫ

Как следует из определения, любое вычисление либо является бесконечной последовательностью, либо завершается состоянием $\langle \varnothing, \eta \rangle$. В последнем случае оценка η называется результатом вычисления.

Будем использовать запись $\longrightarrow^*_I$ для обозначения рефлексивного и транзитивного замыкания отношения переходов $\longrightarrow_I$.

Тогда оценка переменных η является результатом вычисления программы π на оценке переменных θ в интерпретации I в том и только том случае, когда выполняется отношение

$$\langle \pi, \theta \rangle \ \longrightarrow^*_I \ \langle \varnothing, \eta \rangle.$$

ЗАДАЧА ВЕРИФИКАЦИИ ПРОГРАММ

Неформальная постановка.

Программа π считается (частично) корректной , если для любых начальных данных, удовлетворяющих определенному условию φ, результат вычисления (если вычисление завершается) удовлетворяет определенному условию ψ.

Ограничение φ, которое налагается на начальные данные, называется предусловием , а требование ψ, которому должны удовлетворять результаты вычисления, называется постусловием программы.

Задача верификации программы π заключается в проверке частичной корректности программы π относительно заданного предусловия φ и заданного постусловия ψ.

ЗАДАЧА ВЕРИФИКАЦИИ ПРОГРАММ

Формальная постановка.

Расширим множество формул логики предикатов, введя в рассмотрение в качестве формул выражения нового специального вида.

Определение

Триплетом Хоара (тройкой Хоара) называется всякое выражение вида

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

где φ, ψ — формулы логики предикатов,
а π — императивная программа.

Обозначим HT_σ множество триплетов Хоара сигнатуры σ.

ЗАДАЧА ВЕРИФИКАЦИИ ПРОГРАММ

Выполнимость триплетов Хоара в интерпретациях определяется так:

$I \models \varphi\{\pi\}\psi \iff$ для любых оценок переменных θ, η, если $I \models \varphi\theta$ и $\langle \pi, \theta \rangle \longrightarrow_I^* \langle \varnothing, \eta \rangle$, то $I \models \psi\eta$.

Определение (частичной корректности программы)

Пусть φ, ψ — формулы логики предикатов, а π — императивная программа.

Программа π называется частично корректной в интерпретации I относительно предусловия φ и постусловия ψ, если триплет $\varphi\{\pi\}\psi$ выполним в интерпретации I, т. е.

$$I \models \varphi\{\pi\}\psi \ .$$

ЛОГИКА ХОАРА

Как же доказать частичную корректность программы?

Попробуем построить систему правил вывода, аналогичных правилам вывода для семантических таблиц.

Такую систему предложил в 1968 г. Хоар. Правила вывода Хоара имеют вид

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

где $\Phi, \Psi, \Psi_1, \Psi_2$ — триплеты Хоара,
φ, ψ — формулы логики предикатов.

ЛОГИКА ХОАРА

Правила вывода Хоара

ASS: $$\frac{\varphi\{x/t\}\ \{x \Leftarrow t\}\ \varphi}{\textbf{true}},$$

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

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

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

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

ЛОГИКА ХОАРА

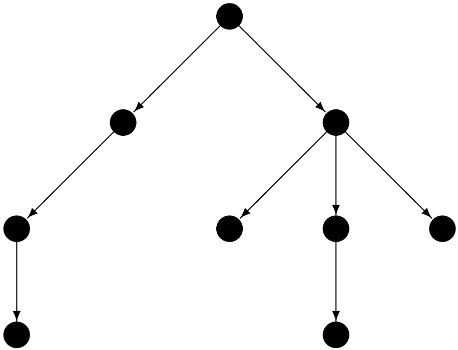
Определение вывода в логике Хоара

Вывод в логике Хоара триплета $\Phi_0 = \varphi_0 \ \{\pi_0\} \ \psi_0$ — это корневое дерево,

ЛОГИКА ХОАРА

Определение вывода в логике Хоара

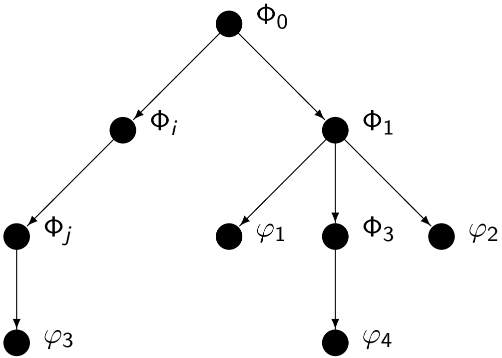
Вывод в логике Хоара триплета $\Phi_0 = \varphi_0 \ \{\pi_0\} \ \psi_0$ — это корневое дерево,



ЛОГИКА ХОАРА

Определение вывода в логике Хоара

Вывод в логике Хоара триплета $\Phi_0 = \varphi_0 \ \{\pi_0\} \ \psi_0$ — это корневое дерево, вершинами которого служат триплеты и формулы логики предикатов и при этом

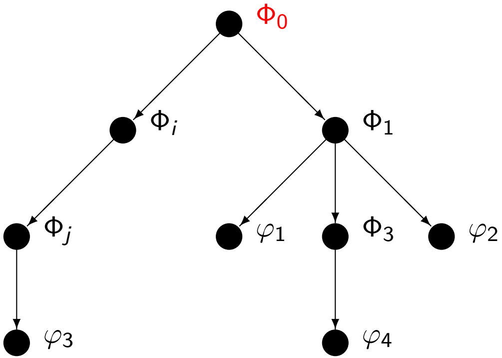


ЛОГИКА ХОАРА

Определение вывода в логике Хоара

Вывод в логике Хоара триплета $\Phi_0 = \varphi_0 \ \{\pi_0\} \ \psi_0$ — это корневое дерево, вершинами которого служат триплеты и формулы логики предикатов и при этом

1) корнем дерева является триплет Φ_0;



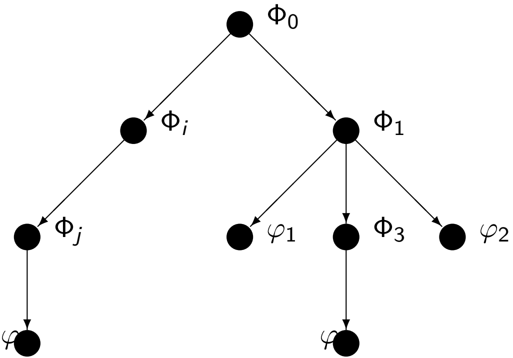
ЛОГИКА ХОАРА

Определение вывода в логике Хоара

2) из вершины Φ_i исходят дуги в вершину Φ_j

$\Longleftrightarrow$

$\dfrac{\Phi_i}{\Phi_j}$ — правило табличного вывода;



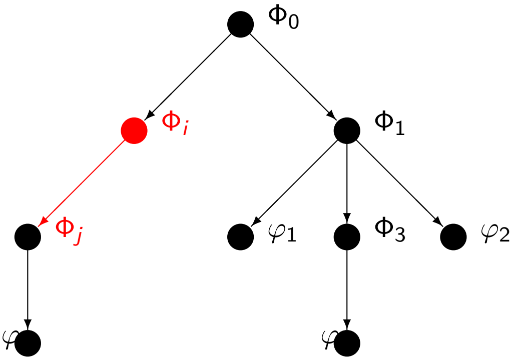
ЛОГИКА ХОАРА

Определение вывода в логике Хоара

2) из вершины Φ_i исходят дуги в вершину Φ_j

$\Longleftrightarrow$

$\dfrac{\Phi_i}{\Phi_j}$ — правило табличного вывода;



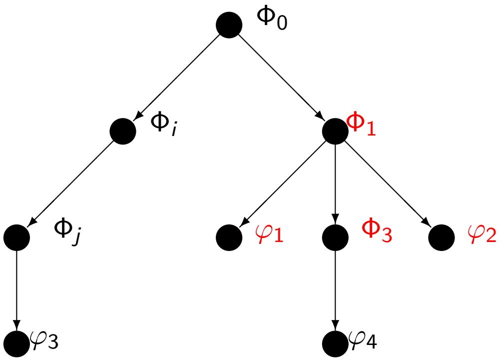
ЛОГИКА ХОАРА

Определение вывода в логике Хоара

2) из вершины Φ_1 исходят дуги в вершины φ_1, Φ_3, φ_2

$\Longleftrightarrow$

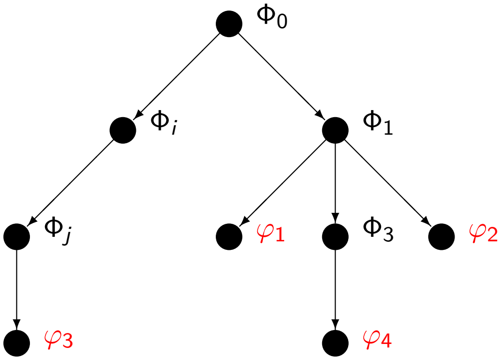
$$\frac{\Phi_1}{\varphi_1,\ \Phi_3,\ \varphi_2}$$ — правило табличного вывода;



ЛОГИКА ХОАРА

Определение вывода в логике Хоара

3) листьями дерева являются формулы логики предикатов.



ЛОГИКА ХОАРА

Определение вывода в логике Хоара

Вывод триплета $\Phi_0 = \varphi_0 \ \{\pi_0\} \ \psi_0$ в логике Хоара называется успешным в интерпретации I, если дерево вывода является конечным, и все его листовые вершины — это истинные в интерпретации I формулы логики предикатов.

ЛОГИКА ХОАРА

Пример

Покажем, что программа

π_0: **while** $\neg(x = y)$
do if $x > y$ **then** $x \Leftarrow x - y$ **else** $y \Leftarrow y - x$ **fi od**

правильно вычисляет наибольший общий делитель двух положительных целых чисел.

Для этого необходимо сформулировать предусловие φ_0 и постусловие ψ_0, соответствующее этому требованию, и построить успешный вывод триплета $\varphi_0\ \{\pi_0\}\ \psi_0$ в логике Хоара.

ЛОГИКА ХОАРА

Пример

Для удобства обозначений введем некоторые вспомогательные формулы:

$$
\begin{array}{lcl}
DIV(x,z) & : & \exists u\ (u \times z = x), \\
GCD(x,y,z) & : & DIV(x,z)\ \&\ DIV(y,z)\ \& \\
 & & \forall u\ (DIV(x,u) \& DIV(y,u) \to (u \leq z)).
\end{array}
$$

Тогда

$$
\begin{array}{lcl}
\varphi_0(x,y,z) & : & (x > 0)\ \&\ (y > 0)\ \&\ GCD(x,y,z), \\
\psi_0(x,z) & : & z = x.
\end{array}
$$

π_0: **while** $\neg(x = y)$

do if $x > y$ **then** $x \Leftarrow x - y$ **else** $y \Leftarrow y - x$ **fi od**

$$\underbrace{(x > 0)\&(y > 0)\&GCD(x, y, z)}_{\varphi_0(x,y,z)}\ \{\pi_0\}\ z = x$$

$$\varphi_0(x, y, z) \rightarrow \varphi_0(x, y, z)$$

$$\varphi_0(x, y, z)\&\neg\neg(x = y) \rightarrow (z = x)$$

$$\varphi_0(x, y, z)\ \{\pi_0\}\ \varphi_0(x, y, z)\&\neg\neg(x = y)$$

$$\varphi_0(x, y, z)\&\neg(x = y)\{\textbf{if } x > y \textbf{ then } x \Leftarrow x - y \textbf{ else } y \Leftarrow y - x \textbf{ fi}\}\varphi_0(x, y, z)$$

$$\varphi_0(x, y, z)\&\neg(x = y)\&\neg(x > y)\{y \Leftarrow y - x\}\varphi_0(x, y, z)$$

$$\varphi_0(x, y, z)\&\neg(x = y)\&(x > y)\{x \Leftarrow x - y\}\varphi_0(x, y, z)$$

Левая ветвь

$$\varphi_0(x,y,z)\&\neg(x=y)\&(x>y)\{x\Leftarrow x-y\}\varphi_0(x,y,z)$$

$$\varphi_0(x,y,z)\&\neg(x=y)\&(x>y)\to$$
$$\to\varphi_0(x-y,y,z)$$

$$\varphi_0(x-y,y,z)\{x\Leftarrow x-y\}\varphi_0(x,y,z)$$

ASS

true

$$\varphi_0(x,y,z)\to\varphi_0(x,y,z)$$

Правая ветвь

$$\varphi_0(x,y,z)\&\neg(x=y)\&\neg(x>y)\{x \Leftarrow y-x\}\varphi_0(x,y,z)$$

$$\varphi_0(x,y,z)\&\neg(x=y)\&\neg(x>y) \to$$
$$\to \varphi_0(x,y-x,z)$$

$$\varphi_0(x,y,z) \to \varphi_0(x,y,z)$$

$$\varphi_0(x,y-x,z)\{y \Leftarrow y-x\}\varphi_0(x,y,z)$$

ASS

true

Пример

Покажем, что построенный вывод в логике Хоара является успешным для стандартной арифметической интерпретации $I_0 = \langle D_{I_0} = \{0, 1, 2, \dots\}, \{+, -, \times\}, <, >, =, \geq, \leq \rangle$.

Для этого достаточно установить истинность в интерпретации I_0 всех формул, стоящих в листьях построенного вывода.

1. $I_0 \models \varphi(x, y, z) \rightarrow \varphi(x, y, z)$ Очевидно.
2. $I_0 \models \varphi_0(x, y, z) \& \neg\neg(x = y) \rightarrow (z = x)$, т. е.
   $I_0 \models (x > 0) \& (y > 0) \& (x = y) \& GCD(x, y, z) \rightarrow (z = x)$.
   Верно.

ЛОГИКА ХОАРА

Пример

3. $I_0 \models \varphi_0(x, y, z) \& \neg(x = y) \& (x > y) \rightarrow$
$\rightarrow \varphi_0(x - y, y, z)$, т. е.
$I_0 \models (x > 0) \& (y > 0) \& (x > y) \& GCD(x, y, z) \rightarrow$
$\rightarrow (x - y > 0) \& (y > 0) \& GCD(x - y, y, z)$.
Верно.
4. $I_0 \models \varphi_0(x, y, z) \& \neg(x = y) \& \neg(x > y) \rightarrow$
$\rightarrow \varphi_0(x, y - x, z)$, т. е.
$I_0 \models (x > 0) \& (y > 0) \& (y > x) \& GCD(x, y, z) \rightarrow$
$\rightarrow (x > 0) \& (y - x > 0) \& GCD(x, y - x, z)$.
Верно.
5. $I_0 \models$ **true**. Очевидно.

Таким образом, все листовые формулы вывода истинны в интерпретации I_0. Значит, вывод триплета $\varphi_0 \; \{\pi_0\} \; \psi_0$ является успешным выводом в интерпретации I_0.

ЛОГИКА ХОАРА

Теорема корректности

Для любой интерпретации I и для любого правила вывода логики Хоара

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

если $I \models \Psi, \quad I \models \varphi, \quad \begin{cases} I \models \Psi_1, \\ I \models \Psi_2, \end{cases} \quad \begin{cases} I \models \varphi, \\ I \models \Psi, \\ I \models \psi, \end{cases}$

то $I \models \Phi$.

Доказательство.

Рассмотрим поочередно все правила вывода логики Хоара.

ЛОГИКА ХОАРА

Доказательство.

Правило

ASS: $\dfrac{\varphi\{x/t\}\ \{x \Leftarrow t\}\ \varphi}{\textbf{true}}$.

Покажем, что в любой интерпретации I верно

$$I \models \varphi\{x/t\}\ \{x \Leftarrow t\}\ \varphi. \qquad (*)$$

Пусть θ — произвольная оценка переменных, и пусть $I \models \varphi\{x/t\}\theta$.
Тогда согласно операционной семантике императивных программ имеется единственное вычисление

$$\langle x \Leftarrow t,\ \theta\rangle \ \longrightarrow_I \ \langle\varnothing,\ \eta\rangle,$$

и при этом $\eta = \{x/t\}\theta$.
Очевидно, $I \models \varphi\eta$, и это доказывает $(*)$.

ЛОГИКА ХОАРА

Доказательство.

Для остальных правил доказательство корректности проводится по той же схеме, но более изощренно.
Попробуйте завершить доказательство самостоятельно.

$\square$

Следствие.

Если триплет $\varphi\{\pi\}\psi$ имеет успешный в интерпретации I вывод, то программа π частично корректна в интерпретации I относительно предусловия φ и постусловия ψ.

В частности, это означает, что исследованная нами программа вычисления наибольшего общего частично корректна в арифметической интерпретации I_0.

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОГРАММ

Как автоматизировать верификацию программ?

Для этого нужно выяснить

1. Полна ли система правил вывода логики Хоара?
2. Существует ли алгоритм построения успешного вывода?

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОГРАММ

Вопрос о полноте правил вывода Хоара.

На самом деле, здесь не один а три вопроса.

1. Верно ли, что для каждой интерпретации I существует система правил вывода, позволяющая для каждого триплета $\Phi = \varphi\{\pi\}\psi$ построить успешный вывод Φ в интерпретации I и **доказать его успешность** в случае $I \models \Phi$?

Ответ отрицательный . Следует из теоремы Геделя о неполноте.

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОГРАММ

Вопрос о полноте правил вывода Хоара.

2. Верно ли, что для каждой интерпретации I существует система правил вывода, позволяющая для каждого триплета $\Phi = \varphi\{\pi\}\psi$ построить успешный вывод Φ в интерпретации I (**но не гарантирующая доказательства его успешности**) в случае $I \models \Phi$?

Ответ отрицательный . Базовые предикаты сигнатуры σ могут быть недостаточно выразительными для представления всех тех отношений между переменными программы, которые нужны для построения успешного вывода.
В результате не найдется нужных формул φ', ψ' для применения правила

CONS: $$\frac{\varphi\{\pi\}\psi}{\varphi \to \varphi',\ \varphi'\{\pi\}\psi',\ \psi' \to \psi}\ .$$

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОГРАММ

Вопрос о полноте правил вывода Хоара.

3. Верно ли, что для некоторых интерпретаций I существует система правил вывода Хоара, которая позволяет для каждого триплета $\Phi = \varphi\{\pi\}\psi$ построить успешный вывод Φ в интерпретации I в случае $I \models \Phi$?

Ответ положительный . Достаточно, чтобы для любого цикла $\pi =$ **while** C **do** π' **od** существовал такой терм t_π, что для любой оценки переменных θ значение терма $t_\pi\theta$ равно $n + 1$ тогда и только тогда, когда цикл π в вычислении $\langle\pi, \theta\rangle$ совершает n итераций.

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОГРАММ

Что нужно для построения успешного вывода?

- Необходимо иметь эффективный прувер для проверки истинности формул в разных интерпретациях:

$$I \models \varphi\ .$$

.

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

поскольку неясно, какие формулы φ', ψ' нужно выбирать в каждом случае.

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОГРАММ

Стратегия вывода в логике Хоара.

Определение

Пусть заданы интерпретация I, императивная программа π и постусловие ψ. Тогда формула φ_0 называется слабейшим предусловием (weakest postcondition) для программы π и постусловия ψ, если

1. $I \models \varphi_0\{\pi\}\psi$,
2. для любой формулы φ, если $I \models \varphi\{\pi\}\psi$, то $I \models \varphi \to \varphi_0$.

Слабейшее предусловие для программы π и постусловия ψ условимся обозначать $wpr(\pi, \psi)$.

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОГРАММ

Какая польза от слабейшего предусловия?

Теорема

$$I \models \varphi\{\pi\}\psi \iff \begin{cases} I \models wpr(\pi, \psi)\{\pi\}\psi, \\ I \models \varphi \to wpr(\pi, \psi). \end{cases}$$

Таким образом, задача построения успешного вывода сводится к задаче вычисления $wpr(\pi, \psi)$.

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОГРАММ

А как вычислять слабейшее предусловие?

Теорема

$$
\begin{aligned}
& wpr(x \Leftarrow t,\ \psi) = \psi\{x/t\}, \\
& wpr(\pi_1;\ \pi_2,\ \psi) = wpr(\pi_1,\ wpr(\pi_2,\ \psi)), \\
& wpr(\textbf{if}\ C\ \textbf{then}\ \pi_1\ \textbf{else}\ \pi_2\ \textbf{fi}, \psi) = \\
& \qquad\qquad C \& wpr(\pi_1,\ \psi) \vee \neg C \& wpr(\pi_2,\ \psi),
\end{aligned}
$$

Доказательство

Самостоятельно.

Таким образом, для многих операторов (программ) слабейшее предусловие вычисляется автоматически.

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОГРАММ

Неужели все так просто?

Увы, нет. Главную трудность представляет оператор цикла **while** C **do** π **od**. Единственный способ верифицировать этот оператор — это воспользоваться производным правилом:

$$\textbf{WHILE-GEN:} \quad \frac{\varphi \; \{\textbf{while } C \textbf{ do } \pi \textbf{ od}\} \; (\psi)}{\varphi \to \chi, \; (\chi \& C) \; \{\pi\} \; \chi, \; (\chi \& \neg C) \to \psi} \, .$$

Это правило требует введения вспомогательной формулы χ, которая называется инвариантом цикла . Инвариант цикла зависит от программы π и условия C.
Автоматическая генерация инвариантов цикла — это ключевая задача в решении проблемы автоматической верификации программ.

КОНЕЦ ЛЕКЦИИ 20.