

# Математические методы верификации схем и программ

mk.cs.msu.ru → Лекционные курсы  
→ Математические методы верификации схем и программ

## Семинар 1

Дедуктивная верификация  
императивных программ

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

ВМК МГУ, 2022/2023, осенний семестр

# Соглашение по умолчанию

Каждое упражнение сформулировано для **целочисленной арифметической интерпретации**  $\mathcal{I}_{ar}$ , содержащей:

- ▶ константы  $0, 1, -1, 2, -2, \dots$
- ▶ функциональные символы  $+, -, \cdot, /, \%$ , обозначающие обычные арифметические операции
- ▶ предикатные символы  $=, \neq, >, <, \geq, \leq$ , обозначающие обычные арифметические отношения

# Упражнение 1

Построить вычисление заданной программы на заданной оценке

```
x := z;  
while x < y do  
  if x%2 > 0 then  
    x := 3 · x + 1;  
  else  
    x := x/2;  
  fi  
od
```

Оценка:

1. {x/15, y/4, z/1}
2. {x/3, y/1, z/0}

## Упражнение 2

Является ли программа в тройке Хоара

- ▶ частично корректной
- ▶ totally корректной

относительно предусловия и постусловия тройки?

$$\{\text{t}\} \quad x := \mathbf{100}; \quad \{\text{t}\}$$

$$\{\text{t}\} \quad x := \mathbf{100}; \quad \{\text{f}\}$$

$$\{\text{f}\} \quad x := \mathbf{100}; \quad \{\text{t}\}$$

$$\{\text{f}\} \quad x := \mathbf{100}; \quad \{\text{f}\}$$

$$\{\text{t}\} \quad x := \mathbf{100}; \quad \{x = \mathbf{100}\}$$

$$\{x = \mathbf{50}\} \quad x := \mathbf{100}; \quad \{x = \mathbf{50}\}$$

$$\{\text{f}\} \quad x := \mathbf{100}; \quad \{x = \mathbf{50}\}$$

$$\{y = \mathbf{50}\} \quad x := \mathbf{100}; \quad \{y = \mathbf{50}\}$$

## Упражнение 2

Является ли программа в тройке Хоара

- ▶ частично корректной
- ▶ totally корректной

относительно предусловия и постусловия тройки?

$\{\text{t}\}$  **while**  $x \neq 0$  **do**  $x := x - 1$ ; **od**  $\{\text{t}\}$

$\{\text{t}\}$  **while**  $x \neq 0$  **do**  $x := x - 1$ ; **od**  $\{\text{f}\}$

$\{\text{f}\}$  **while**  $x \neq 0$  **do**  $x := x - 1$ ; **od**  $\{\text{t}\}$

$\{\text{f}\}$  **while**  $x \neq 0$  **do**  $x := x - 1$ ; **od**  $\{\text{f}\}$

$\{x > 3\}$  **while**  $x \neq 0$  **do**  $x := x - 1$ ; **od**  $\{x = 0\}$

$\{x > 3\}$  **while**  $x \neq 0$  **do**  $x := x - 1$ ; **od**  $\{x > -5\}$

$\{x < 3\}$  **while**  $x \neq 0$  **do**  $x := x - 1$ ; **od**  $\{x = 0\}$

$\{x < -3\}$  **while**  $x \neq 0$  **do**  $x := x - 1$ ; **od**  $\{x = 5\}$

## Упражнение 2

Является ли программа в тройке Хоара

- ▶ частично корректной
- ▶ totally корректной

относительно предусловия и постусловия тройки?

$$\{\text{tt}\} \quad x := E; \quad \{x = E\}$$

( $E$  — произвольный терм)

## Упражнение 3

Предложить предусловие и постусловие, отвечающие требованию корректности программы  $\pi$  (в предположении, что  $\pi$  не содержит переменные  $u_1, u_2, \dots$ ):

1. выходное значение переменной `prod` — это произведение выходных значений переменных `x` и `y`
2. выходное значение переменной `prod` — это произведение входных значений переменных `x` и `y`
3.  $\pi$  меняет местами значения переменных `x` и `y`
4. входные значения переменных `x` и `y` и выходные значения `quo` и `rem` — это, соответственно, делимое, делитель, частное и остаток от деления
5. если входное значение  $n$  переменной `x` неотрицательно, то её выходное значение — это целая часть числа  $\sqrt{n}$

## Упражнение 4

Построить слабое предусловие для следующей программы и постусловия:

$x := x + 10; \{x = 7\}$

$x := x + 10; \{\text{t}\}$

$x := x + 10; \{\text{f}\}$

$x := x + 10; \{x = x + 10\}$

$x := x + 10; y := x + y; \{x = A \ \& \ y = B\}$

**if**  $x = y$  **then**  $x := 7$ ; **else**  $x := x + y + 2$ ; **fi**  $\{x = A \ \& \ y = B \ \& \ z = C\}$



## Упражнение 5

Доказать, что заданная программа частично корректна относительно заданных требований

$$x := x + 1;$$
$$y := y + 1;$$

**Требования:** если входные значения переменных  $x$  и  $y$  равны, то выходные значения также равны

## Упражнение 5

Доказать, что заданная программа частично корректна относительно заданных требований

$z := x;$

$x := y;$

$y := z;$

**Требования:** программа меняет местами значения переменных  $x$  и  $y$

## Упражнение 5

Доказать, что заданная программа частично корректна относительно заданных требований

```
pr := 0; cou := y;  
while cou > 0 do  
  pr := pr + x;  
  cou := cou - 1;  
od
```

**Требования:** выходное значение переменной `pr` — это произведение неотрицательных входных значений переменных `x` и `y`

## Упражнение 5

Доказать, что заданная программа частично корректна относительно заданных требований

```
quo := 0; rem := x;  
while rem  $\geq$  y do  
    rem := rem - y;  
    quo := quo + 1;  
od
```

**Требования:** если входные значения переменных  $x$  и  $y$  соответственно положительны, то входные значения переменных  $x$  и  $y$  и выходные значения  $quo$  и  $rem$  — это, соответственно, делимое, делитель, частное и остаток от деления

## Упражнение 5

Доказать, что заданная программа частично корректна относительно заданных требований

```
x := 0; y := 1; cou := n;  
while cou > 0 do  
  h := y;  
  y := x + y;  
  x := h;  
  cou := cou - 1;  
od
```

**Требования:** выходное значение переменной  $x$  — это  $n$ -е число Фибоначчи (для неотрицательного входного значения переменной  $n$ )

# Немного о тотальной корректности

Чтобы доказать, что частично корректная программа тотально корректна, по определению достаточно показать, что для каждого значения входных данных, удовлетворяющего предусловию, число итераций каждого цикла конечно

Популярный и достаточно простой для понимания способ это доказать — это предоставить **ограничивающую функцию** (**bound function**) для каждого цикла: выражение (терм)  $E$ , такое что:

- ▶ множество всевозможных значений  $E$ , вычисляющихся при проверке условия цикла, ограничено снизу величиной, зависящей только от значения  $E$  непосредственно перед первой итерацией
- ▶ значение  $E$  строго уменьшается на каждой итерации цикла и не может уменьшаться бесконечно долго

# Немного о тотальной корректности

## Пример

Рассмотрим цикл

**while**  $x < 0$  **do**  $x := x + 1$ ; **od**

и произвольное состояние данных до выполнения этого цикла

Тогда  $(-x)$  — ограничивающая функция для этого цикла:

- ▶ значение  $-x$  уменьшается на единицу на каждой итерации
- ▶ если начальное значение  $x$  отрицательно, то граница снизу — **0**
- ▶ если начальное значение  $n$  переменной  $x$  неотрицательно, то граница снизу —  $-n$

## Упражнение 6

Доказать тотальную корректность каждой программы из упражнения 5



# Массивы

$A[t]$  — это терм, обозначающий  $t$ -й элемент массива  $A$

Переменные программы поделены на **целочисленные** и **массивные**

Каждая массивная переменная  $A$  используется только в термах вида  $A[t]$

Никакая целочисленная переменная  $x$  не используется в термах вида  $x[t]$

Терм  $A[t]$  может быть записан в левой части присваивания

В **оценку** переменных программы включены все целочисленные переменные, а также значения переменных

$$\dots, A[-2], A[-1], A[0], A[1], A[2], \dots$$

для каждой массивной переменной  $A$

Отношение переходов (шаг вычисления программы) естественным образом обобщается на присваивания вида  $A[t] := t'$ ;

## Упражнение 7

Предложить предусловие и постусловие, отвечающие требованию корректности программы  $\pi$  (в предположении, что  $\pi$  не содержит переменные  $u_1, u_2, \dots$ ):

1.  $\pi$  обнуляет все значения  $s[0 : n - 1]$  (для заданного входного значения переменной  $n$ )
2.  $\pi$  вычисляет в переменной  $m$  максимальное число среди  $s[0 : n - 1]$  (для заданного входного значения переменной  $n$ ) и не меняет значение  $s$
3.  $\pi$  зеркально отражает набор  $s[0 : n - 1]$  (для заданного входного значения переменной  $n$ )

**Обозначение:**  $s[t_1 : t_2]$  — это набор значений  
( $s[t_1], s[t_1 + 1], \dots, s[t_2]$ )

## Упражнение 8

Доказать частичную корректность заданной программы относительно заданных требований и выяснить, является ли эта программа totally корректной

```
x := 1;  
a[1] := 2;  
a[x] := x;
```

**Требования:** выходное значение переменной  $a[1]$  обязательно равно 1

## Упражнение 8

Доказать частичную корректность заданной программы относительно заданных требований и выяснить, является ли эта программа totally correct

```
x := 0;  
while a[x] ≠ 0 do  
  x := x + 1;  
od
```

**Требования:** если входные значения  $a[0 : 1]$  — это набор  $(1, 0)$ , то

- ▶ выходные значения  $a[0]$  и  $a[1]$  равны входным и
- ▶ выходное значение  $a[x]$  равно 0

## Упражнение 8

Доказать частичную корректность заданной программы относительно заданных требований и выяснить, является ли эта программа totally correct

```
x := 2;  
while a[x] ≠ 0 do  
  x := x + 1;  
od
```

**Требования:** если входные значения  $a[0 : 1]$  — это набор  $(1, 0)$ , то

- ▶ выходные значения  $a[0]$  и  $a[1]$  равны входным и
- ▶ выходное значение  $a[x]$  равно 0

## Упражнение 8

Доказать частичную корректность заданной программы относительно заданных требований и выяснить, является ли эта программа totally корректной

```
i := 1; m := s[0];
while i < n do
  if s[i] < m then
    m := s[i];
  fi
  i := i + 1;
od
```

**Требования:** программа вычисляет в переменной  $m$  максимальное число среди  $s[0 : n - 1]$

**Обозначение:** **if**  $C$  **then**  $\pi$  **fi** — это синоним записи **if**  $C$  **then**  $\pi$  **else**  $\emptyset$  **fi**

## Упражнение 8

Доказать частичную корректность заданной программы относительно заданных требований и выяснить, является ли эта программа totally correct

```
sum := 0; i := 0;
while i < n do
  sum := sum + s[i];
  i := i + 1;
od
```

**Требования:** программа вычисляет в переменной `sum` сумму значений  $s[0 : n - 1]$ , если входное значение `n` положительно

## Упражнение 8

Доказать частичную корректность заданной программы относительно заданных требований и выяснить, является ли эта программа totally correct

```
i := 0;
while 2 · i < n - 1 do
  y := s[i];
  s[i] := s[n - i - 1];
  s[n - i - 1] := y;
  i := i + 1;
od
```

**Требования:** набор значений  $s[0 : n - 1]$  зеркально отражается программой, если  $n \geq 1$



## Challenge

Доказать тотальную корректность следующей программы

```
i := n - 1;
while i > 0 do
  k := i; j := i - 1;
  while j ≥ 0 do
    if s[j] > s[k] then
      k := j;
    fi
    j := j - 1;
  od
  y := s[k]; s[k] := s[i]; s[i] := y;
  i := i - 1;
od
```

**Требования:** программа сортирует набор значений  $s[0 : n - 1]$  по неубыванию, если  $n \geq 1$  перед началом выполнения программы