FMC III - Trabalho 11

Alexandre Ribeiro José Ivo Marina Leite

21 de novembro de 2025

1. Dado um número natural n, o seguinte programa calcula a soma dos primeiros n números naturais. Prove que a fórmula bem formada está totalmente correta. Dica: Seja o invariante de laço $(s = i(i+1)/2 \wedge (i \leq n)$

$\{n > 0\}$
  $i := 0$
  $s := 0$
  while $i < n$ **do**
    $i := i + 1$
    $s := s + i$
  end while
$\{s = n(n+1)/2\}$

Provaremos que a fbf está totalmente correta com invariante de laço $(s = i(i+1)/2) \wedge (i \leq n)$

Antes do laço:

1.	$\{(0 = i(i+1)/2) \wedge (i \leq n)\}\ s := 0\ \{(s = i(i+1)/2) \wedge (i \leq n)\}$	AA
2.	$\{0 = 0 \wedge 0 \leq n\}\ i := 0\ \{(0 = i(i+1)/2) \wedge (i \leq n)\}$	AA
3.	$n > 0$	P
4.	$0 \leq n$	$3, T$
5.	$\{n > 0\}\ i := 0; s := 0\ \{(s = i(i+1)/2) \wedge (i \leq n)\}$	1,2,4,Conseq

Durante a execução do laço:

6.	$\{(s + i = i(i+1)/2) \wedge (i \leq n)\}\ s := s + i\ \{(s = i(i+1)/2) \wedge (i \leq n)\}$	AA
7.	$\{(s + (i+1) = (i+1)(i+2)/2) \wedge (i+1 \leq n)\}\ i := i + 1\ \{(s + i = i(i+1)/2) \wedge (i \leq n)\}$	AA
8.	$(s = i(i+1)/2) \wedge (i \leq n) \wedge (i < n)$	Hipótese (I $\wedge$ B)
9.	$s = i(i+1)/2$	8, Simp
10.	$s + (i+1) = i(i+1)/2 + (i+1)$	9, T
11.	$s + (i+1) = (i+1)(i/2+1) = (i+1)(i+2)/2$	10, T
12.	$i < n$	8, Simp
13.	$i + 1 \leq n$	12, T
14.	$((s = i(i+1)/2) \wedge (i \leq n) \wedge (i < n)) \rightarrow ((s + (i+1) = (i+1)(i+2)/2) \wedge (i+1 \leq n))$	$11, 13$, PC
15.	$\{(s = i(i+1)/2) \wedge (i \leq n) \wedge (i < n)\}\ i := i + 1; s := s + i\ \{(s = i(i+1)/2) \wedge (i \leq n)\}$	$7, 14$, Conseq

Depois do laço:

15.	$(s = i(i+1)/2) \wedge (i \leq n) \wedge \neg(i < n)$	$I \wedge \neg B$
16.	$i \leq n \wedge i \geq n$	$15, T$
17.	$i = n$	$16, T$
18.	$s = n(n+1)/2$	$15, 17$, Subst
19.	$I \wedge \neg B \rightarrow (s = n(n+1)/2)$	15-18, PC

2. O programa a seguir implementa o algoirtmo de divisão para números naturais. Ele calcula o quociente e o resto da divisão de um número natural por um número natural positivo. PRove que a fórmula bem formada (fbf) está totalmente correta. Dica: seja o invariante de laço $(a = yb + x) \wedge (0 \leq x$

$\{(a \geq 0) \wedge (b > 0)\}$
  $x := a;$
  $y := 0;$
  while $b \leq x$ **do**
    $x := x - b;$
    $y := y + 1$
  end while
  $r := x;$
  $q := y;$
  $\{(a = qb + r) \wedge (0 \leq r < b)\}$

(Antes do laço)
Para isso utilizaremos o invariante $(a = yb + x) \wedge (0 \leq x)$

1. $\{(a = 0 \cdot b + x) \wedge (0 \leq x)\}\ y := 0\ \{(a = yb + x) \wedge (0 \leq x)\}$ AA
2. $\{(a = 0 \cdot b + a) \wedge (0 \leq a)\}\ x := a\ \{(a = 0 \cdot b + x) \wedge (0 \leq x)\}$ AA
3. $(a \geq 0) \wedge (b > 0)$ P
4. $(a = 0 \cdot b + a) \wedge (0 \leq a)$ 3, T
5. $\{a \geq 0 \wedge b > 0\}\ x := a; y := 0\ \{I\}$ 1, 2, 4, Conseq.

(Durante a execução do laço).

Para isso faremos uma prova em cima da regra "Enquanto"

6.	$\{(a = (y+1)b + x) \wedge (0 \leq x)\}\ y := y + 1\ \{I\}$	AA
7.	$\{(a = (y+1)b + (x-b)) \wedge (0 \leq x - b)\}\ x := x - b\ \{\dots\}$	AA
8.	$(a = yb + x) \wedge (0 \leq x) \wedge (b \leq x)$	P para PC
9.	$x - b \geq 0$	$8, \mathrm{T}$
10.	$a = yb + x = yb + b + x - b = (y+1)b + (x-b)$	$8, \mathrm{T}$
11.	$(I \wedge B) \rightarrow ((a = (y+1)b + (x-b)) \wedge (0 \leq x - b))$	$8 - 10, \mathrm{PC}$
12.	$\{I \wedge B\}\ x := x - b; y := y + 1\ \{I\}$	$7, 11,$ Conseq

Depois do laço

13.	$\{(a = qb + x) \wedge (0 \leq x < b)\}\ r := x\ \{(a = qb + r) \wedge (0 \leq r < b)\}$	AA
14.	$\{(a = yb + x) \wedge (0 \leq x < b)\}\ q := y\ \{(a = qb + x) \wedge (0 \leq x < b)\}$	AA
15.	$(a = yb + x) \wedge (0 \leq x) \wedge \neg(b \leq x)$	P para PC
16.	$x < b$	$15,$ Simp
17.	$0 \leq x < b$	$15, 16,$ Conj
18.	$(I \wedge \neg B) \rightarrow ((a = yb + x) \wedge (0 \leq x < b))$	15-17, PC
19.	$\{I \wedge \neg B\}\ r := x; q := y\ \{\textit{Pós-condião}\}$	$14, 18,$ Conseq

Terminação: O laço termina pois a variante $t = x$ é estritamente decrescente em $\mathbb{N}$. Como $b > 0$ e $x := x - b$, temos que o novo valor do x sempre será menor que o antigo valor do x. Como $0 \leq x$ (pelo invariante), x é limitado inferiormente.

3. (Máximo Divisor Comum). O programa a seguir afirma encontrar o máximo divisor comum mdc(a,b) de dois inteiros positivos a e b. Prove que a fórmula bem formada (fbf) está totalmente correta.

$\{(a > 0) \wedge (b > 0)\}$

$x := a$
$y := b$
while $x \neq y$ **do**
    if $x > y$ **then**
        $x := x - y$
    else
        $y := y - x$
    end if
end while
$max := x$
$\{max = \text{mdc}(a, b)\}$

Provaremos que a fbf está totalmente correta usando o **Invariante de Laço (I):**

$$I \equiv (\text{mdc}(x, y) = \text{mdc}(a, b)) \wedge (x > 0) \wedge (y > 0)$$

e a **Função Variante (t):** $t = x + y$.

I. Inicialização (Antes do laço)

Devemos provar que a pré-condição implica o invariante após as atribuições iniciais.

1.	$\{(\text{mdc}(a, b) = \text{mdc}(a, b)) \wedge (a > 0) \wedge (b > 0)\}\ x := a; y := b\ \{I\}$	AA (múltiplo)
2.	$a > 0 \wedge b > 0$	Pré-condição (P)
3.	$\text{mdc}(a, b) = \text{mdc}(a, b)$	Reflexividade
4.	$(a > 0 \wedge b > 0) \rightarrow ((\text{mdc}(a, b) = \text{mdc}(a, b)) \wedge a > 0 \wedge b > 0)$	$2, 3$, Lógica
5.	$\{a > 0 \wedge b > 0\}\ x := a; y := b\ \{I\}$	$1, 4$, Conseq

II. Manutenção (Durante a execução do laço)

Devemos provar que $\{I \wedge B\}$ Corpo $\{I\}$. O corpo possui um condicional. Seja $M = \text{mdc}(a, b)$. O invariante é $mdc(x, y) = M \wedge x, y > 0$. A guarda é $x \neq y$.
Caso A: Se $(x > y)$ é verdadeiro (Ramo 'If')

6.	$\{(\text{mdc}(x - y, y) = M) \wedge (x - y > 0) \wedge (y > 0)\}\ x := x - y\ \{I\}$	AA
7.	$I \wedge (x \neq y) \wedge (x > y)$	Hipótese no Ramo If
8.	$\text{mdc}(x, y) = M$	7, Simp de I
9.	$\text{mdc}(x - y, y) = \text{mdc}(x, y) = M$	Propriedade do MDC (Euclides)
10.	$x > y \implies x - y > 0$	7, Aritmética
11.	$(I \wedge x > y) \rightarrow ((\text{mdc}(x - y, y) = M) \wedge (x - y > 0) \wedge (y > 0))$	9, 10, Lógica
12.	$\{I \wedge x > y\}\ x := x - y\ \{I\}$	6, 11, Conseq

Caso B: Se $\neg(x > y)$ é verdadeiro (Ramo 'Else') Como $x \neq y$ (pela guarda) e não é $x > y$, então $y > x$. A prova é simétrica ao Caso A.

13.	$\{I \wedge y > x\}\ y := y - x\ \{I\}$	Análogo aos passos 6-12

Portanto, o invariante se mantém independente do caminho tomado no *If*.

III. Término (Correção Total)

Para provar que o laço não é infinito, usamos a função variante $t = x + y$ com domínio nos Naturais.

14.	$x > 0 \wedge y > 0 \implies x + y > 0$	Do Invariante I
15.	Se $x > y$: $t_{new} = (x - y) + y = x < x + y = t_{old}$	Decrescimento estrito
16.	Se $y > x$: $t_{new} = x + (y - x) = y < x + y = t_{old}$	Decrescimento estrito
17.	O laço termina pois t decresce e é limitado inferiormente por 0.	Conclusão

IV. Finalização (Depois do laço)

Quando o laço termina, a guarda é falsa $(x = y)$ e o invariante I ainda é verdadeiro.

18.	$\{max = \text{mdc}(a, b)\}\ max := x\ \{max = \text{mdc}(a, b)\}$	AA (inválido - atribuição direta)
	Correção do passo lógico final para atribuição:	
19.	$\{x = \text{mdc}(a, b)\}\ max := x\ \{max = \text{mdc}(a, b)\}$	AA
20.	$I \wedge \neg B$	Estado pós-laço
21.	$(\text{mdc}(x, y) = \text{mdc}(a, b)) \wedge (x = y)$	20, Subst
22.	$\text{mdc}(x, x) = x$	Propriedade Aritmética
23.	$x = \text{mdc}(a, b)$	$21, 22$, Transitividade
24.	$(I \wedge \neg B) \rightarrow (x = \text{mdc}(a, b))$	20-23, PC
25.	$\{I \wedge \neg B\}\ max := x\ \{max = \text{mdc}(a, b)\}$	$19, 24$, Conseq

Resolução de forma indutiva: Considere W como o conjunto dos números naturais e a função $f(x, y) = x + y$. Usaremos $<$. Assuma que $mdc(a, b) = mdc(x, y)$ e que $x \neq y$. Seja $s = (x, y)$. Então $f(s) = x + y$. Teremos duas possibilidades para t, de acordo com x e y na última iteração:

1. Se $x < y$, então $t = (x,\ y - x)$ e f(t) = x + (y - x) = y.
2. Se $x > y$, então $t = (x - y,\ y)$ e f(t) = (x - y) + y = x.

Como $a > 0$ e $b > 0$, temos $x > 0$ e $y > 0$. Em ambas as situações acima, temos $f(t) < f(s)$, pois $x < x + y$ e $y < x + y$.
Portanto, o laço termina.
O laço terminará quando x = y e, ao fim do laço, max := x. Ou seja, max := x - y ou max := a (caso x != y na primeira iteração).
O laço termina quando $(x = y)$. No momento da saída do enquanto, temos $(x = y > 0)$ e $mdc(x, y) = mdc(x, x) = mdc(x - y, y)$. Sabemos que $mdc(a, b) = mdc(x - y, y)$. O programa atribui então $(max := x)$, portanto $(max = mdc(a, b))$. Assim ({P} S {Q}) é verdadeiro: se a execução termina, a pós-condição $(mdc(a, b) = max)$ é satisfeita.