Especificação de algoritmos com Isabelle

Bruno Carvalho, Guilherme Schmitt, Henri Dias,
João Fanti, Matheus Santos

Junho de 2016

1 Primeiro problema

1.1 Especificação

mult: N x N → N
requer T
garante mult(x,y)= x * y
mult(x,y) = multacc(x,y,0)
onde
multacc: N x N x N → N
requer T
garante multacc(x,y,z) = x * y + z
invariante ∀k ∈ N. multacc(m0,n0,a0) = mk * nk + ak
multacc01: multacc(0,n,a) = a
multacc02: multacc(k+1,n,a) = multacc(k,n,a + n)

1.2 Exemplo de execução

mult(2, 3)
= multacc(2, 3, 0)
= multacc(1, 3, (0+3))
= multacc(0, 3, (3+3))
= 6

1.3 Teorema th01: multacc(x,y,z) = x * y + z

P ≜ ∀ x,y,z ∈ N. multacc(x,y,z) = x * y + z
P(x) ≜ ∀ y,z ∈ N. multacc(x,y,z) = x * y + z

1.3.1 Base:

P(0) $\triangleq$ $\forall$ y,z $\in$ N. multacc(0,y,z) = 0 * y + z

Seja y0 uma variável arbitrária
Seja a0 uma variável arbitrária
Mostrar multacc(0,y0,a0) = 0 * y0 + a0

Prova:
multacc(0,y0,a0) (por definição)
0 * y0 + a0 (por simp)
a0 (por multacc01)
(0 * y0) + a0 (por aritmética)
0 * y0 + a0 (por simplificação)

1.3.2 Indução:

Seja x0 uma variável arbitrária
Assumir como hipótese P(x0) $\triangleq$ $\forall$ y,z $\in$ N. multacc(x0,y,z) = x0 * y + z
Demonstrar P(x0 + 1) $\triangleq$ $\forall$y,z $\in$ N. multacc(x0 + 1,y,z) = (x0 + 1) * y + z

Seja y0 uma variável arbitrária
Seja a0 uma variável arbitrária
Mostrar multacc(x0 + 1,y0,a0) = (x0 + 1) * y0 + a0

Prova:
multacc x0 y0 (a0 + y0) (por multacc02)
x0 * y0 + (a0 + y0) (por hipótese)
x0 * y0 + a0 + y0 (por aritmética)
(x0 * y0) + (y0 * 1) + a0 (por álgebra)
(y0 * x0) + (y0 * 1) + a0 (por álgebra)
y0 * (x0 + 1) + a0 (por álgebra)
(x0 + 1) * y0 + a0 (por álgebra)

1.4 Teorema th02: mult(x,y) = x * y

P $\triangleq$ $\forall$x,y $\in$ N. mult(x,y) = x * y
P(x) $\triangleq$ $\forall$y $\in$ N. mult(x,y) = x * y

1.4.1 Base:

P(0) $\triangleq$ $\forall$y $\in$ N. mult(0,y) = 0 * y

Seja y0 uma variável arbitrária
Mostrar mult(0,y0) = 0 * y0

Prova:
multacc 0 y0 0 (por mult01)
0 (por multacc01)
0 * y0 (por aritmética)

1.4.2 Indução:

Seja x0 uma variável arbitrária
Assumir como hipótese P(x0) $\triangleq$ $\forall$y $\in$ N. mult(x0,y) = x0 * y
Demonstrar P(x0 + 1) $\triangleq$ $\forall$y $\in$ N. mult(x0 + 1,y) = (x0 + 1) * y

Seja y0 uma variável arbitrária
Mostrar mult(x0 + 1,y0) = (x0 + 1) * y0

Prova:
mult (Suc x0) y0 (por simplificação)
multacc (Suc x0) y0 0 (por mult01)
multacc x0 y0 (0 + y0) (por multacc02)
multacc x0 y0 y0 (por simplificação)
x0 * y0 + y0 (por th01)
y0 * x0 + y0 (por algebra)
(y0 * x0) + (y0 * 1) (por algebra)
y0 * (x0 + 1) (por algebra)
(x0 + 1) * y0 (por algebra)
(Suc x0) * y0 (por simplificação)

2 Segundo problema

2.1 Fatorial

2.1.1 Especificação

fat: N x N $\rightarrow$ N
requer x $\geq$ 0
garante fat(x) = x!
fat(x) = fataux(x,1)
onde
fataux: N x N $\rightarrow$ N
requer x $\geq$ 0
garante fataux(x,y) = x! * y
invariante $\forall$k $\in$ N. fataux(m0,n0) = mk! * nk

fataux01: fataux 0 a = a
fataux02: fataux(n,a) = fataux((n - 1), (a * n)), se n ¿ 0

2.1.2 Exemplo de execução

fat(3)
= fataux (3, 1)
= fataux(2, (1 * 3))
= fataux(1, ((1 * 3) * 2))
= fataux(0, (((1 * 3) * 2) * 1))
= (((1 * 3) * 2) * 1)
= ((3 * 2) * 1)
= (6 * 1)
= 6

2.1.3 Teorema th01: ∀cont. fataux n cont = n! * cont

P $\triangleq$ ∀x,y ∈ N. fataux(x,y) = x! * y
P(x) $\triangleq$ ∀y ∈ N. fataux(x,y) = x! * y

Base: P(0) $\triangleq$ ∀ y ∈ N. fataux(0,y) = 0! * y

Seja cont0 uma variável arbitrária
Mostrar fataux(0, cont0) = 0! * cont0

Prova:
cont0 (por fataux01)
1 * cont0 (por aritmética)
0! * cont0 (por fatorial)

Indução:
Seja m0 uma variável arbitrária
Assumir como hipótese P(m0) $\triangleq$ ∀y ∈ N. fataux(m0,y) = m0! * y
Demonstrar P(m0 + 1) $\triangleq$ ∀y ∈ N. fataux(x0 + 1,y) = (x0 + 1)! * y

Seja a0 uma variável arbitrária
Mostrar fataux(m0 + 1, a0) = (m0 + 1)! * a0

Prova:
fataux (Suc m0) a0 (por simplificação)
fataux m0 ((Suc m0) * a0) (por fataux02)
m0! * ((Suc m0) * a0) (por hipótese)
m0! * (Suc m0) * a0 (por aritmética)
(Suc m0)! * a0 (por simplificação)
(m0 + 1)! * a0 (por simplificação)

2.1.4 Teorema th02: fat n = n!

P $\triangleq \forall$x $\in$ N. fat(x) = x!
P(x) $\triangleq$ fat(x) = x!

Base: P(0) $\triangleq$ fat(0) = 0!
Mostrar fat(0) = 0!

Prova:
1 (por fat01)
0! (por th01)

Indução:
Seja m0 uma variável arbitrária
Assumir como hipótese P(m0) $\triangleq$ fat(m0) = m0!
Demonstrar P(m0 + 1) $\triangleq$ fat(m0 + 1) = (m0 + 1)!

Mostrar fat(m0 + 1) = (m0 + 1)!

Prova:
fat (Suc m0) (por simplificação)
fataux (Suc m0) 1 (por fat02)
(Suc m0)! * 1 (por th01)
(Suc m0)! (por aritmética)
(m0 + 1)! (por simplificação)

2.2 Somatório

2.2.1 Especificação

somatorio: N $\rightarrow$ N
requer: x $\geq$ 0
garante somatorio(n) =

$$\sum_{i=1}^{n} i^2 = (n/6) * (n+1) * (2*n+1)$$

somatorio(n) = sumaux(n,0)
onde
sumaux: N x N $\rightarrow$ N
requer x $\geq$ 0
garante sumaux(n,t) =

$$\sum_{i=1}^{n} i^2 + t$$

invariante ∀k ∈ N. sumaux(w0,u0) =

$$\sum_{i=1}^{wk} i^2 + uk$$

sumaux01: sumaux(0,a) = a
sumaux02: sumaux(n,a) = sumaux(n - 1,n^2 + a), se n > 0

2.2.2 Exemplo de execução

somatorio(4)
= sumaux(4,0)
= sumaux(3,4 * 4 + 0)
= sumaux(2,3 * 3 + 4 * 4 + 0)
= sumaux(1,2 * 2 + 3 * 3 + 4 * 4 + 0)
= sumaux(0,1 * 1 + 2 * 2 + 3 * 3 + 4 * 4 + 0)
= 1 * 1 + 2 * 2 + 3 * 3 + 4 * 4 + 0
= 30

2.2.3 Teorema th01:

$$\sum_{i=1}^{n} i^2 = (n/6) * (n+1) * (2*n+1)$$

P ≜ ∀n ∈ N.

$$\sum_{i=1}^{n} i^2 = (n/6) * (n+1) * (2*n+1)$$

P(n) ≜

$$\sum_{i=1}^{n} i^2 = (n/6) * (n+1) * (2*n+1)$$

Base:
P(0) ≜

$$\sum_{i=1}^{0} i^2 = (0/6) * (0+1) * (2*0+1)$$

Mostrar

$$\sum_{i=1}^{0} i^2 = (0/6) * (0+1) * (2*0+1)$$

Prova:

0 (por aritmética)
(n/6) * (n + 1) * (2 * n + 1) (por aritmética)

Indução: Seja n0 uma variável arbitrária
Assumir como hipótese P(n0) $\triangleq$

$$\sum_{i=1}^{n0} i^2 = (n0/6) * (n0 + 1) * (2 * n0 + 1)$$

Demonstrar P(n0 + 1) $\triangleq$

$$\sum_{i=1}^{n0+1} i^2 = (n0 + 1/6) * (n0 + 1 + 1) * (2 * (n0 + 1) + 1)$$

Mostrar

$$\sum_{i=1}^{n0+1} i^2 = (n0 + 1/6) * (n0 + 1 + 1) * (2 * (n0 + 1) + 1)$$

Prova:

$$\sum_{i=1}^{n0+1} i^2 + (n0 + 1)^2$$

(por aritmética)
(n0/6) * (n0 + 1) * (2 * n0 + 1) + (n0 + 1)^2 (por hipótese)
((n0 + 1)/6) * (n0 + 2) * (2 * (n0 + 1) + 1) (por aritmética)

2.2.4 Teorema th02: sumaux(n, t) =

$$\sum_{i=1}^{n} i^2 + t$$

P $\triangleq$ $\forall$n,t $\in$ N. sumaux(n,t) =

$$\sum_{i=1}^{n} i^2 + t$$

P(n) $\triangleq$ $\forall$t $\in$ N. sumaux(n,t) =

$$\sum_{i=1}^{n} i^2 + t$$

Base:
P(0) $\triangleq$ $\forall$t $\in$ N. sumaux(0,t) =

$$\sum_{i=1}^{0} i^2 + t$$

Prova:
Seja t0 uma variável arbitrária
Mostrar sumaux(0,t0) =

$$\sum_{i=1}^{0} i^2 + t0$$

t0 (por sumaux01)
i=

$$\sum_{i=1}^{0} i^2 + t0$$

(por aritmética)

Indução:
Seja n0 uma variável arbitrária
Assumir como hipótese
P(n0) $\triangleq$ $\forall$t $\in$ N. sumaux(n0,t) =

$$\sum_{i=1}^{n0} i^2 + t$$

Demonstrar
P(n0 + 1) $\triangleq$ $\forall$t $\in$ N. sumaux(n0 + 1,t) =

$$\sum_{i=1}^{n0+1} i^2 + t$$

Prova:
Seja t0 uma variável arbitrária
Mostrar sumaux(n0 + 1,t0) =

$$\sum_{i=1}^{n0+1} i^2 + t0$$

sumaux(n0,(n0 + 1)^2 + t0) (por sumaux02)

$$\sum_{i=1}^{n0+1} (n0+1)^2 + t0$$

(por hipótese)

$$\sum_{i=1}^{n0+1} i^2 + t0$$

(por aritmética)

2.2.5 Teorema th03: somatorio(n) =

$$\sum_{i=1}^{n} i^2$$

P $\triangleq$ $\forall$n $\in$ N. somatorio(n) =

$$\sum_{i=1}^{n} i^2$$

P(n) $\triangleq$ $\forall$n $\in$ N. somatorio(n) =

$$\sum_{i=1}^{n} i^2$$

Base:
P(0) $\triangleq$ $\forall$n $\in$ N. somatorio(0) =

$$\sum_{i=1}^{0} i^2$$

Prova:
Mostrar somatorio(0) =

$$\sum_{i=1}^{0} i^2$$

sumaux(0,0) (por definição)
0 (por definição)

$$\sum_{i=1}^{0} i^2$$

(por aritmética)

Indução:

Seja n0 uma variável arbitrária
Assumir como hipótese P(n0) $\triangleq$ somatorio(n0) =

$$\sum_{i=1}^{n0} i^2$$

Demonstrar P(n0 + 1) $\triangleq$ somatorio(n0 + 1) =

$$\sum_{i=1}^{n0+1} i^2$$

Prova:
Mostrar somatorio(n0 + 1) =

$$\sum_{i=1}^{n0+1} i^2$$

sumaux(n0 + 1,0) (por definição)
sumaux(n0,(n0 + 1)^2 + 0) (por definição)

$$\sum_{i=1}^{n0} (n0+1)^2 + 0$$

(por definição da invariante sumaux)

$$\sum_{i=1}^{n0+1} i^2$$

(por aritmética)

2.2.6 Teorema: sum(n) = (n/6) * (n + 1) * (2 * n + 1)

P $\triangleq$ $\forall$n $\in$ N. sum(n) = (n/6) * (n + 1) * (2 * n + 1)
P(n) $\triangleq$ $\forall$n $\in$ N sum(n) = (n/6) * (n + 1) * (2 * n + 1)

Base:
P(0) $\triangleq$ sum(0) = (0/6) * (0 + 1) * (2 * 0 + 1)

Prova:
Mostrar sum(0) = (0/6) * (0 + 1) * (2 * 0 + 1)
sumaux(0,0) (por definição)
0 (por definição)
(0/6) * (0 + 1) * (2 * 0 + 1) (por aritmética)

Indução:
Seja n0 uma variável arbitrária
Assumir como hipótese P(n0) $\triangleq$ sum(n0) = (n0/6) * (n0 + 1) * (2 * n0 + 1)
Demonstrar P(n0 + 1) $\triangleq$ sum(n0 + 1) = ((n0 + 1)/6) * (n0 + 2) * (2 * (n0 + 1) + 1)

Prova:
Mostrar sum(n0 + 1) = ((n0 + 1)/6) * (n0 + 2) * (2 * (n0 + 1) + 1)
sumaux(n0 + 1,0) (por definição)
sumaux(n0,$(n0 + 1)^2$ + 0) (por definição)

$$\sum_{i=1}^{n0} i^2 + (n0+1)^2 + 0$$

(por definição da invariante sumaux)
sum(n0) + $(n0 + 1)^2$ (por definição sum(n) =

$$\sum_{i=1}^{n0} i^2$$

)
(n0/6) * (n0 + 1) * (2 * n0 + 1) + $(n0 + 1)^2$ (por hipótese)
((n0 + 1)/6) * (n0 + 2) * (2 * (n0 + 1) + 1) (por aritmética)

3 Terceiro problema

3.1 Especificação recursiva na cauda

tail_len_c: List → N
requer L
garante tail_len_c L = length L
tail_len_c01: tail_len_c [] = 0
tail_len_c02: tail_len_c T = tail_len_c_aux T 0
onde
tail_len_c_aux: List x N → N
requer L
tail_len_c_aux01: tail_len_c_aux [] a = a
tail_len_c_aux02: tail_len_c_aux (h#T) a = tail_len_c_aux T (Suc(a))

3.1.1 Exemplo de execução

tail_len_c [4, 7]
= tail_len_c_aux [4, 7] 0
= tail_len_c_aux [7] 1
= tail_len_c_aux [] 2
= 2

3.2 Especificação não recursiva na cauda

tail_len_nc: List → N
requer L
garante tail_len_nc L = length L *tail_len_nc01*: tail_len_nc [] = 0 *tail_len_nc02*:
tail_len_nc (h#T) = 1 + tail_len_nc T

3.2.1 Exemplo de execução

tail_len_nc [4, 5]
= tail_len_nc (1 + tail_len_nc [5])
= 1 + (tail_len_nc (1 + tail_len_nc []))
= (1 + 1 + 0)
= (2 + 0)
= 2

3.3 Teorema th01: tail_len_nc l = length l

P ≜ tail_len_nc l = length l
P(l) ≜ tail_len_nc l = length l

3.3.1 Base:

P([]) ≜ tail_len_nc

Mostrar tail_len_nc [] = length []

Prova:
tail_len_nc [] = 0 (por tail_len_nc01)
length [] (por simplificação)

3.3.2 Indução:

Seja l0 uma variável arbitrária
Seja *e* uma variável arbitrária
Assumir como hipótese P(l0) ≜ tail_len_nc l0 = length l0
Demonstrar P(e#l0) ≜ tail_len_nc (e#l0) = length (e#l0)

Prova:
tail_len_nc l0 + 1 (por tail_len_nc02)
length l0 + 1 (por hipótese)
length (e#l0) (por simplificação)

3.4 Teorema th02: ∀a. tail_len_c_aux l a = length l + a

P ≜ ∀a ∈ N. tail_len_c_aux l a = length l + a
P(l) ≜ ∀a ∈ N. tail_len_c_aux l a = length l + a

3.4.1 Base:

P([]) ≜ ∀a ∈ N. tail_len_c [] a = length [] + a
Seja a0 uma variável arbitrária
Mostrar tail_len_c_aux [] a0 = length [] + a0

Prova:
a0 (por tail_len_c_aux01)
length [] + a0 (por simplificação)

3.4.2 Indução:

Seja l0 uma variável arbitrária
Seja *elem* uma variável arbitrária
Assumir como hipótese P(l0) ≜ ∀a. tail_len_c_aux l0 a = length l0 + a
Demonstrar P(elem#l0) ≜ ∀a. tail_len_c_aux (elem#l0) a = length (elem#l0) + a

Seja a0 uma variável arbitrária
Mostrar P(elem#l0) ≜ tail_len_c_aux (elem#l0) a0 = length (elem#l0) + a0

Prova:
tail_len_c_aux l0 (Suc a0) (por tail_len_c_aux02)
length l0 + (Suc a0) (por hipótese)
length (elem#l0) + a0 (por simplificação)

3.5 Teorema th03: tail_len_nc T = tail_len_c T

P ≜ tail_len_nc T = tail_len_c T
P(l) ≜ tail_len_nc l = tail_len_c l

3.5.1 Base:

P([]) ≜ tail_len_nc [] = tail_len_c []

Prova:

tail_len_nc [] = 0 (por tail_len_nc01)
tail_len_c [] (por tail_len_c01)

3.5.2 Indução:

Seja T0 uma variável arbitrária
Seja a0 uma variável arbitrária
Assumir como hipótese P(T0) $\triangleq$ tail_len_nc T0 = tail_len_c T0
Demonstrar P(a0#T0) $\triangleq$ tail_len_nc (a0#T0) = tail_len_c (a0#T0)

Prova:
length (a0#T0) (por th01)
= tail_len_c_aux (a0#T0) 0 (por th02)
= tail_len_c (a0#T0)" (por tail_len_c02)