CORRETUDE DE ALGORITMOS ITERATIVOS

Prof. André Vignatti

CORRETUDE DE ALGORITMO ITERATIVOS

Para provar a corretude de algoritmos iterativos:

CORRETUDE DE ALGORITMO ITERATIVOS

Para provar a corretude de algoritmos iterativos:

Identificar laços: Analisar um laço por vez, começando do mais interno

CORRETUDE DE ALGORITMO ITERATIVOS

Para provar a corretude de algoritmos iterativos:

Identificar laços: Analisar um laço por vez, começando do mais interno

Achar invariantes: Para cada laço, achar um *invariante de laço* que permanece verdade em toda repetição e que captura o "progresso" feito pelo laço. (Achar o invariante é quase sempre a parte mais difícil)

CORRETUDE DE ALGORITMO ITERATIVOS

Para provar a corretude de algoritmos iterativos:

Identificar laços: Analisar um laço por vez, começando do mais interno

Achar invariantes: Para cada laço, achar um *invariante de laço* que permanece verdade em toda repetição e que captura o "progresso" feito pelo laço. (Achar o invariante é quase sempre a parte mais difícil)

Provar invariantes: Provar que os invariantes de laço são verdadeiros

CORRETUDE DE ALGORITMO ITERATIVOS

Provar término: Usar os invariantes de laço para provar que o algoritmo termina.

CORRETUDE DE ALGORITMO ITERATIVOS

Provar término: Usar os invariantes de laço para provar que o algoritmo termina.

Provar corretude: Usar os invariantes de laço para provar que o algoritmo obtém o resultado correto.

NOTAÇÃO

Iremos nos concentrar em algoritmos de um único laço

NOTAÇÃO

Iremos nos concentrar em algoritmos de um único laço

O valor da variável x **imediatamente após** a i-ésima iteração do laço é denotado por x_i

NOTAÇÃO

Iremos nos concentrar em algoritmos de um único laço

O valor da variável x **imediatamente após** a i-ésima iteração do laço é denotado por x_i

($i = 0$ significa imediatamente antes de entrar no laço pela primeira vez)

NOTAÇÃO

Iremos nos concentrar em algoritmos de um único laço

O valor da variável x **imediatamente após** a i-ésima iteração do laço é denotado por x_i

($i = 0$ significa imediatamente antes de entrar no laço pela primeira vez)

Por exemplo, x_6 é o valor da variável x depois da sexta vez que o laço foi executado.

FIBONACCI

Algoritmo $fib(n)$

- **se** $n = 0$ **então retorna** 0
- $a \leftarrow 0; b \leftarrow 1; i \leftarrow 2$
- **enquanto** $i \leq n$ **faça**
  - $c \leftarrow a + b; a \leftarrow b; b \leftarrow c; i \leftarrow i + 1$
- **retorna** b

FIBONACCI

Algoritmo $fib(n)$

se $n = 0$ **então retorna** 0

$a \leftarrow 0; b \leftarrow 1; i \leftarrow 2$

enquanto $i \leq n$ **faça**

$c \leftarrow a + b; a \leftarrow b; b \leftarrow c; i \leftarrow i + 1$

retorna b

Teorema. $\mathrm{fib}(n)$ devolve F_n.

Fatos sobre o Algoritmo

$$i_0 = 2$$
$$i_{j+1} = i_j + 1$$

$$a_0 = 0$$
$$a_{j+1} = b_j$$

$$b_0 = 1$$
$$b_{j+1} = c_{j+1}$$

$$c_{j+1} = a_j + b_j$$

Algoritmo $fib(n)$
  se $n = 0$ **então retorna** 0
  $a \leftarrow 0; b \leftarrow 1; i \leftarrow 2$
  enquanto $i \leq n$ **faça**
    $c \leftarrow a + b; a \leftarrow b; b \leftarrow c; i \leftarrow i + 1$
  retorna b

O Invariante de Laço

Teorema. Para todo natural $j \geq 0, i_j = j + 2, a_j = F_j$ e $b_j = F_{j+1}$.

O Invariante de Laço

Teorema. Para todo natural $j \geq 0, i_j = j + 2, a_j = F_j$ e $b_j = F_{j+1}$.

faz sentido?

Algoritmo $\mathit{fib}(n)$
  se $n = 0$ **então retorna** 0
  $a \leftarrow 0; b \leftarrow 1; i \leftarrow 2$
  enquanto $i \leq n$ **faça**
    $c \leftarrow a + b; a \leftarrow b; b \leftarrow c; i \leftarrow i + 1$
  retorna b

Teorema. Para todo natural $j \geq 0, i_j = j + 2, a_j = F_j$ e $b_j = F_{j+1}$.

Teorema. Para todo natural $j \geq 0, i_j = j + 2, a_j = F_j$ e $b_j = F_{j+1}$.

Demonstração. (Indução em j)

Teorema. Para todo natural $j \geq 0, i_j = j + 2, a_j = F_j$ e $b_j = F_{j+1}$.

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $i_0 = 2, a_0 = 0 = F_0$ e $b_0 = 1 = F_1$

Teorema. Para todo natural $j \geq 0, i_j = j + 2, a_j = F_j$ e $b_j = F_{j+1}$.

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $i_0 = 2, a_0 = 0 = F_0$ e $b_0 = 1 = F_1$

Hipótese: Para $j \geq 0$, $i_j = j + 2, a_j = F_j$ e $b_j = F_{j+1}$.

Teorema. Para todo natural $j \geq 0, i_j = j + 2, a_j = F_j$ e $b_j = F_{j+1}$.

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $i_0 = 2, a_0 = 0 = F_0$ e $b_0 = 1 = F_1$

Hipótese: Para $j \geq 0$, $i_j = j + 2, a_j = F_j$ e $b_j = F_{j+1}$.

Passo: Queremos mostrar que $i_{j+1} = j + 3$, $a_{j+1} = F_{j+1}$ e $b_{j+1} = F_{j+2}$

Passo: Queremos mostrar que $i_{j+1} = j + 3$, $a_{j+1} = F_{j+1}$ e $b_{j+1} = F_{j+2}$

Passo: Queremos mostrar que $i_{j+1} = j + 3$, $a_{j+1} = F_{j+1}$ e $b_{j+1} = F_{j+2}$

$$\begin{aligned} i_{j+1} &= i_j + 1 \\ \text{(hip. de indução)} &= (j + 2) + 1 \\ &= j + 3 \end{aligned}$$

Passo: Queremos mostrar que $i_{j+1} = j+3$, $a_{j+1} = F_{j+1}$ e $b_{j+1} = F_{j+2}$

$$\begin{aligned} i_{j+1} &= i_j + 1 \\ \text{(hip. de indução)} &= (j+2) + 1 \\ &= j + 3 \end{aligned}$$

$$\begin{aligned} a_{j+1} &= b_j \\ \text{(hip. de indução)} &= F_{j+1} \end{aligned}$$

Passo: Queremos mostrar que $i_{j+1} = j+3$, $a_{j+1} = F_{j+1}$ e $b_{j+1} = F_{j+2}$

$$\begin{aligned} i_{j+1} &= i_j + 1 \\ \text{(hip. de indução)} &= (j+2)+1 \\ &= j+3 \end{aligned}$$

$$\begin{aligned} a_{j+1} &= b_j \\ \text{(hip. de indução)} &= F_{j+1} \end{aligned}$$

$$\begin{aligned} b_{j+1} &= c_{j+1} \\ &= a_j + b_j \\ \text{(hip. de indução)} &= F_j + F_{j+1} \\ &= F_{j+2} \end{aligned}$$

$\square$

Prova de Corretude

Teorema. O algoritmo termina com b contendo F_n.

Prova de Corretude

Teorema. O algoritmo termina com b contendo F_n.

Demonstração.

A afirmação é claramente verdadeira se $n = 0$

Prova de Corretude

Teorema. O algoritmo termina com b contendo F_n.

Demonstração.

A afirmação é claramente verdadeira se $n = 0$

Se $n > 0$, então entramos no laço

Prova de Corretude

Teorema. O algoritmo termina com b contendo F_n.

Demonstração.

A afirmação é claramente verdadeira se $n = 0$

Se $n > 0$, então entramos no laço

Término: Como $i_{j+1} = i_j + 1$, eventualmente i será igual a $n + 1$ e o laço irá terminar.

Prova de Corretude

Teorema. O algoritmo termina com b contendo F_n.

Demonstração.

A afirmação é claramente verdadeira se $n = 0$

Se $n > 0$, então entramos no laço

Término: Como $i_{j+1} = i_j + 1$, eventualmente i será igual a $n + 1$ e o laço irá terminar.

Suponha que isso acontece após t iterações.

Prova de Corretude

Teorema. O algoritmo termina com b contendo F_n.

Demonstração.

A afirmação é claramente verdadeira se $n = 0$

Se $n > 0$, então entramos no laço

Término: Como $i_{j+1} = i_j + 1$, eventualmente i será igual a $n + 1$ e o laço irá terminar.

Suponha que isso acontece após t iterações.

Como $i_t = n + 1$, e $i_t = t + 2$, conclui-se que $t = n - 1$.

Prova de Corretude

Teorema. O algoritmo termina com b contendo F_n.

Demonstração.

A afirmação é claramente verdadeira se $n = 0$

Se $n > 0$, então entramos no laço

Término: Como $i_{j+1} = i_j + 1$, eventualmente i será igual a $n + 1$ e o laço irá terminar.

Suponha que isso acontece após t iterações.

Como $i_t = n + 1$, e $i_t = t + 2$, conclui-se que $t = n - 1$.

Resultado: Pelo invariante de laço, $b_t = F_{t+1} = F_n$.

$\square$

MULTIPLICAÇÃO

Algoritmo $multiplica(y, z)$

$x \leftarrow 0$

enquanto $z > 0$ **faça**

  se z *é ímpar* **então** $x \leftarrow x + y$

  $y \leftarrow 2y$

  $z \leftarrow \lfloor z/2 \rfloor$

retorna x

MULTIPLICAÇÃO

Algoritmo *multiplica*(y, z)

$x \leftarrow 0$

enquanto $z > 0$ **faça**

 se z *é ímpar* **então** $x \leftarrow x + y$

 $y \leftarrow 2y$

 $z \leftarrow \lfloor z/2 \rfloor$

retorna x

Teorema. Se $y, z \in \mathbb{N}$, então multiplica(y, z) devolve yz.

Um Resultado Preliminar

Teorema. Para todo $n \in \mathbb{N}$,

$$2\lfloor n/2 \rfloor + (n \bmod 2) = n.$$

Um Resultado Preliminar

Teorema. Para todo $n \in \mathbb{N}$,

$$2\lfloor n/2 \rfloor + (n \bmod 2) = n.$$

Demonstração.

(Caso 1: n par) Então $\lfloor n/2 \rfloor = n/2$ e $n \bmod 2 = 0$, e o resultado segue.

Um Resultado Preliminar

Teorema. Para todo $n \in \mathbb{N}$,

$$2\lfloor n/2 \rfloor + (n \bmod 2) = n.$$

Demonstração.

(Caso 1: n par) Então $\lfloor n/2 \rfloor = n/2$ e $n \bmod 2 = 0$, e o resultado segue.

(Caso 2: n ímpar) Então $\lfloor n/2 \rfloor = (n-1)/2$ e $n \bmod 2 = 1$, e o resultado segue. $\square$

Fatos sobre o Algoritmo

$$y_{j+1} = 2y_j$$

Algoritmo *multiplica*(y, z)
- $x \leftarrow 0$
- **enquanto** $z > 0$ **faça**
  - **se** z *é ímpar* **então** $x \leftarrow x + y$
  - $y \leftarrow 2y$
  - $z \leftarrow \lfloor z/2 \rfloor$
- **retorna** x

Fatos sobre o Algoritmo

$$y_{j+1} = 2y_j$$

$$z_{j+1} = \lfloor z_j/2 \rfloor$$

Algoritmo $multiplica(y, z)$
- $x \leftarrow 0$
- **enquanto** $z > 0$ **faça**
  - **se** z *é ímpar* **então** $x \leftarrow x + y$
  - $y \leftarrow 2y$
  - $z \leftarrow \lfloor z/2 \rfloor$
- **retorna** x

Fatos sobre o Algoritmo

$$y_{j+1} = 2y_j$$

$$z_{j+1} = \lfloor z_j/2 \rfloor$$

$$x_0 = 0$$

$$x_{j+1} = x_j + y_j(z_j \bmod 2)$$

Algoritmo *multiplica*(y, z)
- $x \leftarrow 0$
- **enquanto** $z > 0$ **faça**
  - **se** z *é ímpar* **então** $x \leftarrow x + y$
  - $y \leftarrow 2y$
  - $z \leftarrow \lfloor z/2 \rfloor$
- **retorna** x

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

Demonstração. (Indução em j)

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $y_j z_j + x_j = y_0 z_0 + x_0 = y_0 x_0$

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $y_j z_j + x_j = y_0 z_0 + x_0 = y_0 x_0$

Hipótese: Para $j \geq 0$, $y_j z_j + x_j = y_0 z_0$

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $y_j z_j + x_j = y_0 z_0 + x_0 = y_0 x_0$

Hipótese: Para $j \geq 0$, $y_j z_j + x_j = y_0 z_0$

Passo: Queremos provar que $y_{j+1} z_{j+1} + x_{j+1} = y_0 z_0$.

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $y_j z_j + x_j = y_0 z_0 + x_0 = y_0 x_0$

Hipótese: Para $j \geq 0$, $y_j z_j + x_j = y_0 z_0$

Passo: Queremos provar que $y_{j+1} z_{j+1} + x_{j+1} = y_0 z_0$.

Pelo Fatos do Algoritmo,

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $y_j z_j + x_j = y_0 z_0 + x_0 = y_0 x_0$

Hipótese: Para $j \geq 0$, $y_j z_j + x_j = y_0 z_0$

Passo: Queremos provar que $y_{j+1} z_{j+1} + x_{j+1} = y_0 z_0$.

Pelo Fatos do Algoritmo,

$$y_{j+1} z_{j+1} + x_{j+1} = 2 y_j \lfloor z_j / 2 \rfloor + x_j + y_j (z_j \bmod 2)$$

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $y_j z_j + x_j = y_0 z_0 + x_0 = y_0 x_0$

Hipótese: Para $j \geq 0$, $y_j z_j + x_j = y_0 z_0$

Passo: Queremos provar que $y_{j+1} z_{j+1} + x_{j+1} = y_0 z_0$.

Pelo Fatos do Algoritmo,

$$\begin{aligned} y_{j+1} z_{j+1} + x_{j+1} &= 2 y_j \lfloor z_j/2 \rfloor + x_j + y_j (z_j \bmod 2) \\ &= y_j \big(2 \lfloor z_j/2 \rfloor + (z_j \bmod 2)\big) + x_j \end{aligned}$$

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $y_j z_j + x_j = y_0 z_0 + x_0 = y_0 x_0$

Hipótese: Para $j \geq 0$, $y_j z_j + x_j = y_0 z_0$

Passo: Queremos provar que $y_{j+1} z_{j+1} + x_{j+1} = y_0 z_0$.

Pelo Fatos do Algoritmo,

$$\begin{aligned} y_{j+1} z_{j+1} + x_{j+1} &= 2 y_j \lfloor z_j/2 \rfloor + x_j + y_j (z_j \bmod 2) \\ &= y_j \big(2 \lfloor z_j/2 \rfloor + (z_j \bmod 2) \big) + x_j \\ \text{(resultado preliminar)} &= y_j z_j + x_j \end{aligned}$$

O Invariante de Laço

Teorema. Para todo natural $j \geq 0$,

$$y_j z_j + x_j = y_0 z_0.$$

Demonstração. (Indução em j)

Base: para $j = 0$ é trivial, pois $y_j z_j + x_j = y_0 z_0 + x_0 = y_0 x_0$

Hipótese: Para $j \geq 0$, $y_j z_j + x_j = y_0 z_0$

Passo: Queremos provar que $y_{j+1} z_{j+1} + x_{j+1} = y_0 z_0$.

Pelo Fatos do Algoritmo,

$$\begin{aligned} y_{j+1} z_{j+1} + x_{j+1} &= 2y_j \lfloor z_j/2 \rfloor + x_j + y_j (z_j \bmod 2) \\ &= y_j \big(2\lfloor z_j/2 \rfloor + (z_j \bmod 2)\big) + x_j \\ \text{(resultado preliminar)} &= y_j z_j + x_j \\ \text{(hipótese)} &= y_0 z_0. \end{aligned}$$

$\blacksquare$

Prova de Corretude

Teorema. O algoritmo termina com x contendo yz.

Prova de Corretude

Teorema. O algoritmo termina com x contendo yz.

Demonstração.

Prova de Corretude

Teorema. O algoritmo termina com x contendo yz.

Demonstração.

Término: Em cada iteração z é dividido pela metade (e arrendondado para baixo se for ímpar). Logo, para alguma iteração t, $z_t = 0$ e o laço termina.

Prova de Corretude

Teorema. O algoritmo termina com x contendo yz.

Demonstração.

Término: Em cada iteração z é dividido pela metade (e arrendondado para baixo se for ímpar). Logo, para alguma iteração t, $z_t = 0$ e o laço termina.

Resultado: Pelo invariante de laço,

$$y_t z_t + x_t = y_0 z_0.$$

Prova de Corretude

Teorema. O algoritmo termina com x contendo yz.

Demonstração.

Término: Em cada iteração z é dividido pela metade (e arrendondado para baixo se for ímpar). Logo, para alguma iteração t, $z_t = 0$ e o laço termina.

Resultado: Pelo invariante de laço,

$$y_t z_t + x_t = y_0 z_0.$$

Como $z_t = 0$, temos que $x_t = y_0 z_0 = yz$.

∎