Lista de Exercícios 4

Gustavo Higuchi

August 18, 2016

Contents

Exercício 1

Teorema 1. *Para todo* $n \geq 0$*,* $g(n)$ *devolve* $3^n - 2^n$

Proof.
Base : para $n = 0$, g(n) devolve 0 e está correto,
para $n = 1$, g(n) devolve 1 e está correto também

Hipótese : para $n \geq 2$ e todo $0 \leq m < n$, g(m) devolve $3^m - 2^m$
Passo : queremos provar que para g(n), o algoritmo devolve $3^n - 2^n$.
assim temos

$$\begin{aligned}
&5 * g(n-1) - 6 * g(n-2) \\
&5 * (3^{n-1} - 2^{n-1}) - 6(3^{n-2} - 2^{n-2}) \\
&5 * 3^{n-1} - 5 * 2^{n-1} - 2 * 3 * 3^{n-2} + 3 * 2 * 2^{n-2} \\
&5 * 3^{n-1} - 2 * 3^{n-1} - 5 * 2^{n-1} + 3 * 2^{n-1} \\
&3 * 3^{n-1} - 2 * 2^{n-1} = 3^n - 2^n
\end{aligned} \tag{1}$$

□

Exercício 2

Teorema 1. *Para todo* $y, z \geq 0$*, mult(y,z) retorna* $y * z$

Proof.
Base : para $z = 0$, $mult(y, 0)$ devolve 0 e está correto,
para $z = 1$, $mult(y, 1)$ devolve
$mult(2y, \left\lfloor \frac{1}{2} \right\rfloor) + y(1 mod 2)$
$mult(2y, 0) + y * 1$
$0 + y = y$, o que está certo também.
Hipótese : para $z \geq 2$ e todo $0 \leq n \leq z$, $mult(y, n)$ devolve $y * n$
Passo : queremos provar que para $mult(y, z+1)$, o algoritmo devolve $y*(z+1)$.
assim temos
Para $z + 1$ ímpar

$$\begin{aligned}
mult(y, z+1) &= mult(2y, \left\lfloor \frac{(z+1)}{2} \right\rfloor) + y \\
&= 2y * \left\lfloor \frac{(z+1)}{2} \right\rfloor + y, \text{ da hipótese} \\
&= 2y * \frac{z}{2} + y \\
&= y * z + y = y * (z+1)
\end{aligned} \tag{2}$$

Para $z+1$ par

$$\begin{aligned} mult(y, z+1) &= mult(2y, \left\lfloor \frac{(z+1)}{2} \right\rfloor) \\ &= 2y * \frac{(z+1)}{2}, \text{ da hipótese} \\ &= y * (z+1) \end{aligned} \tag{3}$$

$\square$

Exercício 3

Teorema 1. *Para todo $y, z \geq 0$, o algoritmo $power(y, z)$ devolve y^z.*

Proof.

Base : para $z = 0$, $power(y, 0)$ devolve 1 e está correto,
para $z = 1$, $power(y, 1)$ devolve
$power(y^2, \left\lfloor \frac{1}{2} \right\rfloor) * y$
$1 * y = y$, o que está certo também.

Hipótese : para $z \geq 2$ e todo $0 \leq n \leq z$, $power(y, n)$ devolve y^n

Passo : queremos provar que para $power(y, z+1)$, o algoritmo devolve y^{z+1}.
Para $z+1$ ímpar, temos

$$\begin{aligned} power(y, z+1) &= power(y^2, \left\lfloor \frac{(z+1)}{2} \right\rfloor) * y \\ &= y^{2\left\lfloor \frac{z}{2} \right\rfloor} * y, \text{ pela hipótese} \\ &= y^{2*\frac{z}{2}} * y \\ &= y^z * y = y^{z+1} \end{aligned} \tag{4}$$

Para $z+1$ par, temos

$$\begin{aligned} power(y, z+1) &= power(y^2, \left\lfloor \frac{(z+1)}{2} \right\rfloor) \\ &= y^{2*\frac{(z+1)}{2}}, \text{ pela hipótese} \\ &= y^{z+1} \end{aligned} \tag{5}$$

$\square$

Exercício 4

Teorema 1. *Para todo $n > 0$, o algoritmo $sum(A, n)$ devolve $\sum_{i=1}^{n} A[i]$.*

Proof.
Base : para $n = 1$, $sum(A, 1)$ devolve $A[1]$ e está correto,
para $n = 2$, $sum(A, 2)$ devolve
$sum(A, 2 - 1) + A[2]$
$sum(A, 1) + A[2] = A[1] + A[2]$, o que está certo também.

Hipótese : para $n \geq 3$ e todo $0 \leq m \leq n$, $sum(A, m)$ devolve $\sum_{i=1}^{m} A[i]$

Passo : queremos provar que para $sum(A, n+1)$, o algoritmo devolve $\sum_{i=1}^{n+1} A[i]$.
Assim temos

$$\begin{aligned} sum(A, n+1) &= sum(A, n) + A[n+1] \\ &= \sum_{i=1}^{n} A[i] + A[n+1], \text{ pela hipótese} \\ &= \sum_{i=1}^{n+1} A[i] \end{aligned} \tag{6}$$

$\square$