Lista de Exercícios 0.9

Igor Lacerda Faria

[1]Departamento de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG) - Belo Horizonte - MG - Brasil

igorlfs@ufmg.br

Parte da lista está em papel no final.

Revisão

1. (a) Uma definição recursiva é uma definição estabelece a priori um conjunto de casos "base" e uma regra (ou mais) para gerar seus outros elementos, com base nos iniciais. Os elementos essenciais de uma definição recursiva são justamente os casos base e a regra para estender a definição.

Exercícios

5. (a) $\begin{cases} \bigwedge_{i=1}^{0} p_i = T, \\ \bigwedge_{i=1}^{n} p_i = (\bigwedge_{i=1}^{n-1} p_i) \wedge p_n, n \geq 1 \end{cases}$

   (b) $\begin{cases} \bigvee_{i=1}^{0} p_i = F, \\ \bigvee_{i=1}^{n} p_i = (\bigvee_{i=1}^{n-1} p_i) \vee p_n, n \geq 1 \end{cases}$

6. (a) $f(n) = f(n-1) + 2; f(1) = 0$[1]

   (b) $f(n) = f(n-1) + 3; f(1) = 2$

   (c) $f(n) = f(n-1) + 5; f(1) = 1, f(2) = 2, f(3) = 3, f(4) = 4$

7. (a)
   - Primeira iteração: (0,1), (1,1), (2,1)
   - Segunda iteração: (0,2), (2,1), (2,1); (1,2), (2,2), (3,1); (2,2), (3,2), (4,2)
   - Terceira[2] iteração: (0,3), (1,2), (2,3)

[1]Os não negativos, pelo menos

[2]Eu vou continuar em um "galho" só, porque isso é muito chato

- Quarta iteração: (0,4), (1,4), (2,4)

(b) Caso base: $0 \leq 2 \cdot 0 = 0$ é válido. Seja um $(a, b) \in S$, tal que (a, b) possui a propriedade. Vamos mostrar que ela se mantém ao iterar. Se, por hipótese, $a < 2b$, então $a < 2(b+1)$, $a+1 < b+1$ e $a+2 \leq b+1$. Assim concluímos que os resultados da iteração também atendem à regra. Terminamos nossa prova por indução estrutural.

8. (a) Com base no plot dos gráficos:
   Passo base: $(1, 1) \in S$,
   Passo recursivo: Se $(a, b) \in S$, então $(a, b + 2) \in S, (a + 1, b + 1) \in S \wedge (a + 2, b) \in S$.

   (b) Com base no plot dos gráficos:
   Passo base: $(1, 1) \in S, (2, 1) \in S, (1, 2) \in S$
   Passo recursivo: Se $(a, b) \in S$, então $(a + 2, b) \in S \wedge (a, b + 2) \in S$.

9. Seja r uma função que retorna a string reversa de uma dada string. Podemos definir r recursivamente como: *Passo base:* $r(\lambda) = \lambda$. *Passo recursivo:* Seja w uma string de tamanho $n + 1$, tal que $w = xy$, em que x é uma string de tamanho n (e y uma string unitária), então $r(w) = yr(x)$.

10. Começamos definindo recursivamente tanto o número vértices $n(T)$ como a altura $h(T)$:

$$h(T) = \begin{cases} 0, & \text{se o único vértice da árvore binária completa } T \text{ é a própria raiz} \\ 1 + \max(h(T_1), h(T_2)), & \text{se a árvore binária completa } T \text{ é formada por uma raiz conectada a duas sub-árvores } T_1 \text{ e } T_2 \end{cases}$$

$$n(T) = \begin{cases} 1, & \text{se o único vértice da árvore binária completa } T \text{ é a própria raiz} \\ 1 + n(T_1) + n(T_2), & \text{se a árvore binária completa } T \text{ é formada por uma raiz conectada a duas sub-árvores } T_1 \text{ e } T_2 \end{cases}$$

Passo base: para uma árvore binária completa T consistindo apenas num vértice raiz, note que, por definição, $n(T) = 1$ e $h(T) = 0$, logo, a desigualdade é satisfeita pois:

$$1 = n(T) \geq 2h(T) + 1 = 2 \cdot 0 + 1 = 1$$

Passo indutivo: A nossa hipótese de indução é que temos:

$$n(T_1) \geq 2h(T_1) + 1 \land n(T_2) \geq 2h(T_2) + 1$$

Sempre que T_1 e T_2 forem árvores binárias completas. Assuma que T é uma árvore binária completa tendo T_1 e T_2 como sub-árvores imediatas. As fórmulas recursivas de $n(T)$ e $h(T)$ determinam que:

$$n(T) = 1 + n(T_1) + n(T_2) \land h(T) = 1 + \max(h(T_1), h(T_2))$$

Assim, calculamos:

$$\begin{aligned}
n(T) &= \\
1 + n(T_1) + n(T_2) &\geq \\
1 + 2h(T_1) + 1 + 2h(T_2) + 1 &= \\
2(1 + h(T_1) + h(T_2)) + 1 &\geq \\
2(1 + \max\left(h(T_1), h(T_2)\right)) + 1 &= \\
2(h(T)) + 1 &
\end{aligned}$$

Sendo que a primeira desigualdade decorre da hipótese e a segunda desigualdade vale porque $a + b \geq \max(a, b) \forall a, b \geq 0$.

11. *Passo base:* o menor elemento em um conjunto com um elemento é o próprio elemento. *Passo indutivo:* o menor elemento em um conjunto com $n + 1$ elementos é o menor elemento entre o conjunto com n elementos e o elemento novo.

    Para encontrar o menor elemento no conjunto $\{3, 5, 1, 2, 4\}$, fazemos:

$$\begin{aligned}
\min(\{3, 5, 1, 2, 4\}) &= \\
\min(\min(\{3, 5, 1, 2\}), 4) &= \\
\min(\min(\min(\{3, 5, 1\}), 2), 4) &= \\
\min(\min(\min(\min((\{3, 5\}), 1), 2), 4) &= \\
\min(\min(\min(\min(\min(\{3\}), 5), 1), 2), 4) &= \\
\min(\min(\min(\min(3, 5), 1), 2), 4) &= \\
\min(\min(\min(3, 1), 2), 4) &= \\
\min(\min(1, 2), 4) &= \\
\min(1, 4) &= \\
1 &
\end{aligned}$$