Computação Fiável
Indução - exercícios básicos

Simão Melo de Sousa

17 de Outubro de 2011

Conteúdo

1 Indução Estrutural

Exercício 1 *Demonstre por indução estrutural que:*

- $\forall n \in \mathbb{N}.\ (3^{3.n+2} + 2^{n+4})$ *é divisível por 5.* **Resposta**

- $\forall n \in \mathbb{N}^*, 1.2.3 + 2.3.4 + \ldots + n.(n+1).(n+2) = \frac{n.(n+1).(n+2).(n+3)}{4}$. **Resposta**

- $n^4 - 4.n^2$ *é divisível por* 3 *para todo o* $n \geq 0$. **Resposta**

- $\sum_{k=0}^{n} k = \frac{n(n+1)}{2}$. **Resposta**

- $\sum_{k=0}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$. **Resposta**

- $\sum_{k=0}^{n} k^3 = \frac{n^2(n+1)^2}{4}$. **Resposta**

- $\sum_{k=0}^{n} (a.r)^k = \frac{a(1-r^{n+1})}{(1-r)}$. **Resposta**

- $\forall n \in \mathbb{N},\ n^2(n^2-1)$ *é divisível por* 12. **Resposta**

$\square$

Exercício 2 *Demonstre por inducão estrutural sobre* n *que* $\forall n \in \mathbb{N}, n^2(n^2-1)$ *é divisível por* 12. $\square$

Exercício 3 *Vamos aqui considerar o conjunto* $\mathbb{N}^*$ *(os naturais sem o* 0*). Considere a seguinte sequência de somas:*

$$\frac{1}{1\times 2};\quad \frac{1}{1\times 2}+\frac{1}{2\times 3};\quad \frac{1}{1\times 2}+\frac{1}{2\times 3}+\frac{1}{3\times 4};\quad \cdots$$

- *Calcule as somas do exemplo e apresente um padrão geral para esta sequência de somas.*

- *Demonstre* ***por indução estrutural*** *a conjectura apresentada na alínea anterior.*

$\square$

Exercício 4 *Seja* $f : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$*, a função recursiva definida por:*

$$f(m,n) \triangleq \begin{cases} n+1 & se\ m=0 \\ f(m-1,1) & se\ n=0 \wedge m\neq 0 \\ f(m-1,f(m,n-1)) & se\ n>0 \wedge m>0 \end{cases}$$

Demonstre por indução que $\forall k \in \mathbb{N},\ f(1,k) = k+2$ **Resposta**

demonstre por indução que $\forall k \in \mathbb{N},\ f(2,k) = 2\times k+3$. $\square$

Exercício 5

1. *Defina de forma indutiva o conjunto* bin_A *das árvores binárias* não vazias *de elementos dum conjunto* A*. Por árvores não vazias, entendemos que as mais pequenas árvores deste conjunto são folhas (árvores com um só elemento do conjunto* A*);*

2. *Dê o princípio de indução associada a esta definição indutiva;*
3. *Defina a função arestas que calcula o número de vértice da árvore em parâmetro;*
4. *Defina a função nodos que calcula o número de nodos da árvore em parâmetro;*
5. *Demonstre que* $\forall a \in bin_A, nodos(a) = arestas(a) + 1$.

Resposta

□

Exercício 6 *Explique brevemente a diferença entre a noção de função recursiva e a noção de função estruturalmente recursiva.*

Resposta

□

Exercício 7 *Neste exercício vamos considerar uma definição indutiva das fórmulas da lógica proposicional.*

Seja $\mathcal{V} \triangleq \{P, Q, R, S, \ldots\}$ *um conjunto numerável de variáveis chamadas variáveis proposicionais. Seja* $\mathcal{C}$ *o conjunto de conectivas* $\{\wedge, \vee, \rightarrow, \neg, \perp, \top\}$. *O conjunto* $\mathcal{P}rop$ *das fórmulas proposicionais é definido como o menor subconjunto* X *do monoíde livre* $(\mathcal{V} \cup \mathcal{C} \cup \{"(", ")"\})^*$ *verificando os (B) e (I) seguintes:*

(B):
1. *Para todo o* $x \in \mathcal{V}$, x *pertence a* X
2. $\top$ *pertence a* X
3. $\perp$ *pertence a* X

(I):
1. *Seja* F *uma fórmula de* X $(F \in X)$, *então* $\neg F \in X$
2. *Sejam* F *e* G *duas fórmulas de* X *(i.e.* $F, G \in X$*), então* $(F \wedge G) \in X$
3. *Sejam* F *e* G *duas fórmulas de* X, *então* $(F \vee G) \in X$
4. *Sejam* F *e* G *duas fórmulas de* X, *então* $(F \rightarrow G) \in X$

1. *Dê o princípio de indução associada a esta definição indutiva;*

2. *Seja npe* : $\mathcal{P}rop \to \mathbb{N}$, *a função que devolve o número de parêntesis esquerdos da fórmula em parâmetro. De forma semelhante, seja npd* : $\mathcal{P}rop \to \mathbb{N}$, *a função que devolve o número de parêntesis direitos da fórmula em parâmetro. Demonstre que* $\forall F \in \mathcal{P}rop, npe(F) = npd(F)$.

Resposta

□

Exercício 8 *O objectivo deste exercício é a definição indutiva do conjunto das datas válidas. Uma data válida é um terno* (d, m, a) *onde d e a são inteiros que representam respectivamente o dia e o ano, e m uma palavra representando um mês (como a palavra "Fevereiro" que representa o mês de Fevereiro). Imagine que exista uma relação ternária* $val(d, m, a)$ *que seja verdade se o dia d é um dia possível para o mês m e o ano a. Por exemplo não temos* $val(29, Fevereiro, 2001)$ *porque* 2001 *não é um ano bissexto, também não temos* $val(31, Setembro, 2001)$ *nem* $val(0, Setembro, 2001)$ *porque Setembro só tem* 30 *dias e porque não existe o dia* 0.

1. *Defina indutivamente o conjunto* Mês *dos meses.*

2. *Defina indutivamente o conjunto* Data *das datas válidas.*

□

Exercício 9 *Considere o tipo expr das expressões aritméticas simples seguintes:*

```
(* tipo expr onde:
   I = constante inteira, V = variável, A = +, S = −, M = *, D = /
*)
type expr = I of int | V of string | A of expr*expr
            | S of expr*expr | M of expr*expr | D of expr*expr

(* tipo dos ambientes (associação variável−valor)*)
type env = (string*int) list
```

Considere igualmente a função eval seguinte:

```
let rec eval (e:expr) (ambiente: env)=
  match e with
      I i → i
    | V v → assoc v ambiente
```

```
    | A (e1,e2) → eval e1 ambiente + eval e2 ambiente
    | S (e1,e2) → eval e1 ambiente - eval e2 ambiente
    | M (e1,e2) → eval e1 ambiente * eval e2 ambiente
    | D (e1,e2) → eval e1 ambiente / eval e2 ambiente
```

onde assoc é a função que devolve o valor inteiro associado a parâmetro v no ambiente ambiente, se este existir.

- *Qual é o princípio de indução associada a definição indutiva de expr?*
- *Defina uma função* `simplify : expr →expr` *que execute sobre toda a estrutura do seu parâmetro as transformações seguintes:*

  para uma qualquer expressão e, $\begin{array}{ll} e+0=e & e-0=e \\ e*1=e & e/1=e \\ e+e=2*e & e-e=0 \end{array}$

  Por exemplo a expressão $\frac{((x+0)+x)}{1}$ *se simplifica em* $2*x$ *porque, pelas regras definidas,* $\frac{((x+0)+x)}{1}$ *se transforma em* $((x+0)+x)$, $x+0$ *se transforma em* x *e* $x+x$ *se transforma em* $2*x$. *Repare que a ordem de aplicação destas simplificações é irrelevante se todas elas são de facto executadas.*
- *Demonstre por indução que* $\forall e : expr, \forall a : env, eval\ (simplify\ e)\ a = eval\ e\ a$. *Assuma para esse efeito e* se necessário *que o ambiente a tem todas as propriedades desejadas. Por exemplo, o ambiente tem todas as variáveis presentes na expressão considerada.*

□

Exercício 10 *Definir os seguintes conjuntos indutivos:*

1. *A parte* $\mathbb{N}_3$ *de* $\mathbb{N}$ *dos inteiros múltiplos de três.*
2. *A parte* L_A *do monoide livre* B^* *(onde o alfabeto B é* $A \cup \{'[', ']', ':'\}$*) das listas de elementos de um conjunto A. Fornecer igualmente a definição constructiva do conjunto considerado.*
3. *A parte* AB_A *do monoide livre* B^* *(onde o alfabeto B é* $A \cup \{'(', ')', ','\}$*) das árvores de elementos de um conjunto A.*
4. *A parte D do monoide livre* A^* *(onde o alfabeto A é* $\{(,)\}$*) das expressões bem "parenteseadas"(conhecida por* Linguagem de Dyck. *Por exemplo* ()(()()), ()() *e* ((())) *são palavras da linguagems de Dyck, mas* ((), ()) *e* ())(*não são palavras da linguagem.*

5. *A parte A^* do monoide livre A^* (sendo A um alfabeto qualquer).*

□

Exercício 11 *Seja A um alfabeto, define-se $AB_{A,n}$, $(n \in \mathbb{N})$ por*

$$AB_{A,0} = \{\emptyset\} \tag{1}$$
$$AB_{A,n+1} = AB_{A,n} \cup \{(a,g,d) |\ a \in A \text{ e } g,d \in AB_{A,n}\} \tag{2}$$

1. *Mostrar que $AB_{A,\omega} (= \bigcup_{n\in\mathbb{N}} AB_{A,n})$ é o conjunto AB_A.*

2. *Para $A = \{a,b,c\}$, exibir $AB_{A,2}$.*

3. *Exibir uma árvore binária que não pertença a AB_A. Porque nunca será ela gerado por $AB_{A,\omega}$?*

□

Exercício 12 *Explique brevemente a importância da noção de não ambiguidade aquando da definição de funções por recursividade estrutural*

Resposta □

Exercício 13 *Qual é a altura de 18 nos conjuntos indutivo dos inteiros, dos inteiros pares, dos inteiros múltiplos de três?*

□

Exercício 14 *Seja o polinómio $p(x) = \frac{1}{3}x^3 - \frac{1}{2}x^2 + \frac{1}{6}x$*

1. *Definir $p(x+1) - p(x)$.*

2. *Mostrar que para todo o $n \in \mathbb{N}$, $p(x) \in \mathbb{N}$.*

□

Exercício 15 *Mostrar que qualquer palavra da linguagem de Dyck tem tantas parêntesis esquerdas como parêntesis direitas.*

□

Exercício 16 *Encontrar o erro no raciocínio seguinte:*

"Em qualquer grupo de pessoas, todas as pessoas têm a mesma idade"

Demonstração: *Por indução sobre o número de pessoas no grupo (em $\mathbb{N}^*$).*

Caso base *num grupo de uma pessoa, todas as pessoas tem a mesma idade, trivialmente.*

Caso do passo indutivo *Hipótese de indução: todas as pessoas tem a mesma idade em qualquer grupo de n pessoas.*
Seja G um grupo de $n+1$ pessoas, sejam G_1 e G_2 os grupos das n primeiras e últimas pessoas de G. Todas as pessoas de G_1 e de G_2 têm a mesma idade, por hipótese. Logo a primeira pessoa de G_1 tem a mesma idade do que a segunda pessoa de G_1, essa segunda pessoa de G_1 é a primeira pessoa de G_2 e tem a mesma idade do que a última de G_2, logo a primeira pessoa de G ($\in G_1, \notin G_2$) tem a mesma idade do que a última pessoa de G ($\in G_2, \notin G_1$). Logo, todas as pessoas de G têm a mesma idade.

Conclusão *Fica então demonstrado que em qualquer grupo de pessoas, todas as pessoas têm a mesma idade*

Resposta

□

Exercício 17

1. *Definir a função subtermo(t) que devolve o conjuntos dos subtermos de um termo $t \in T$.*

2. *Definir a função altura(a) que devolve a altura de uma árvore $a \in AB_A$.*

3. *Definir a função nós(a) que devolve o número de nós da árvore parâmetro $a \in AB_A$.*

4. *Definir a função $folhas(a)$ que devolve o número de folhas da árvore parâmetro $a \in AB_A$.*

5. *Definir a função pot(x, n) que devolve x^n ($n \in \mathbb{N}$).*

6. *Definir a função pertence(x, l) que devolve 1 se $x(\in A)$ pertence a $l(\in L_A)$, 0 senão.*

7. *Definir a função comprimento(l) que devolve o tamanho da lista $l(\in L_A)$.*

8. *Definir a função* $inverte(l)$ *que devolve a lista inversa a lista* $l(\in L_A)$ *(a lista em que os elementos estão em ordem inversa em comparação a* l*).*

9. *Definir a função* $concat(l_1, l_2)$ *que concatenas a lista* l_2 *a lista* l_1 $(l_1, l_2 \in L_A)$.

□

Exercício 18 *Uma árvore binária é estrita se não tiver um nó com um só filho e se não for a árvore vazia.*

1. *Dar a definição indutiva de* ABs_A*, o conjunto das árvores binárias estritas.*

2. *Mostrar que em* ABs_A *se tem* $n(x) = 2 * f(x) - 1$ *em que* x *é um elemento de* ABs_A*,* n *a função que devolve o número de nós de uma árvore, e* f *a função que devolve o número de folhas de uma árvore.*

□

Exercício 19

a) *Definir a linguagem* T *de termos baseada no alfabeto* $F = F_0 \cup F_2$ *onde* $F_0 = \{c\}$ *e* $F_2 = \{f\}$.

b) *Dado a interpretação* h *em* $\mathbb{N}$ *seguinte,*

$$h(c) = 1 \tag{3}$$

$$h_f(n, m) = n + m \tag{4}$$

que conjunto X *é definido por* $X = \{h^*(t) |\ t \in T\}$*?*

c) *Será a definição de* X*, correspondente a* T*, ambígua?*

□

Exercício 20 *Seja* $\mathbb{N}^* = \mathbb{N}/\{0\}$ *Seja a definição seguinte de* $modulo(n, m)$ *no conjunto indutivo* $\mathbb{N} \times \mathbb{N}^*$ *por*

$$modulo(n, m) = \begin{cases} n & se\ n < m \\ modulo((n-m), m) & senão \end{cases}$$

Esta definição define modulo como uma função apesar de $\mathbb{N} \times \mathbb{N}^*$ *ser definido ambiguamente. Extrair uma definição não ambígua de* $\mathbb{N} \times \mathbb{N}^*$ *que corresponde a função modulo.*

□

Exercício 21 • *Definir* $maxelem(a)$ *e* $minelem(a)$, $a \in AB_A$, *as funções que devolvem o maior e menor elemento* ($\in A$) *da árvore* a.

- *Definir o conjunto das árvores* ABo_A *das árvores ordenadas.*
- *Provar que o percurso infixo de uma árvore ordenada* a *devolve sempre a lista crescente dos elementos de* a.

□

2 Indução Bem Fundada

Exercício 22 *Considere a sequência (de Fibonacci) de inteiros seguintes:*

- $F_1 = F_2 = 1$
- $F_n = F_{n-1} + F_{n-2}$

Demonstre por indução bem fundada que $F_n = \frac{1}{\sqrt{5}}(\Phi^n - \bar{\Phi}^n)$, *onde* $\Phi = \frac{1+\sqrt{5}}{2}$ *e* $\bar{\Phi} = \frac{1-\sqrt{5}}{2}$.

□

Exercício 23 *Demonstre que:*

- *a relação de inclusão é uma relação bem fundada;*
- *a relação* $\leq$ *sobre* $\mathbb{N}$ *é uma relação bem fundada*
- *a relação* $\leq_{div}$ *sobre* $\mathbb{N}^\star$ *é uma relação bem fundada*

Resposta

- *a relação* $\leq$ *sobre* $\mathbb{Z}$ *não é uma relação bem fundada*

□

Exercício 24 *1. Mostre que* $\forall n \in \mathbb{N}.(n+1)^2 - (n+2)^2 - (n+3)^2 + (n+4) = 4$

2. *Deduzir que qualquer inteiro m pode ser escrito como soma e diferença dos quadrados $1^2, 2^2, 3^2, \ldots, n^2$ para um determinado n. Isto é:*

$$\forall m \in \mathbb{N}.\exists n \in \mathbb{N}.\exists \epsilon_1, \ldots \epsilon_n \in \{-1, 1\}.m = \epsilon_1.1^2 + \epsilon_2.2^2 + \ldots + \epsilon_n.n^2$$

□

Exercício 25 *Sejam $(A, \leq_A)$ e $(B, \leq_B)$ dois conjuntos* ***ordenados*** *por ordens largas* ***totais*** *e* ***bem fundadas*** *(diz-se, neste caso, que $\leq_A$ e $\leq_B$ são boas ordens). Seja $(A \times B, \leq^{\mathcal{L}})$ o conjunto $A \times B$ ordenado pela ordem lexicográfica $\leq^{\mathcal{L}}$ i.e.*

$$(a, b) \leq^{\mathcal{L}} (c, d) \triangleq \begin{cases} (a <_A c) \\ (a = c) \wedge (b \leq_B d) \end{cases}$$

onde $(a <_A c) \triangleq (a \leq_A c) \wedge (a \neq c)$. Demonstre que $(A \times B, \leq^{\mathcal{L}})$ é bem fundado.

Resposta □

Exercício 26 *Seja* `d` *a função OCaml seguinte:*

```
let rec d x y =
    if x < y then 0
    else if y = 0 then x
            else (d  (x-y) y) +1 ;;
```

1. *Diga,* ***brevemente****, o que calcula a função* `d`*.*

2. *Demonstre, por indução bem fundada, que a função* `d` *termina.*

Resposta

□

Exercício 27 *Sejam misterio a seguinte função OCaml.*

```
let rec misterio f e l a =
match l with
 [] → a
| el::li →
    let (a1,a2)=a in
      if (f el e)
      then misterio f e li (el::a1,a2)
      else misterio f e li (a1,el::a2)
```

1. *Diga o que calcula a função misterio. Considere por exemplo a execução de*
   misterio $(<)\ 4\ [3;7;4;1;8]\ ([],[])$.

2. *Demonstre a terminação da função misterio. Para tal, asuma que a função parâmtero f termina.*

Resposta □

Exercício 28 *Seja* `f` *a função OCaml seguinte:*

```
let rec f x =
   if (x<1)
     then 0
     else
       if (x=1)
         then 1
         else (f (x-2))*(f (x-1))/2
```

Demonstre, usando a indução bem fundada, que $\forall n \in \mathbb{N}.\ (f\ n)$ *termina.*

□

Exercício 29 *Diga, dos conjuntos ordenados seguintes, quais são os conjuntos ordenados bem fundados. No caso negativo apresenta uma justificação formal (um contra-exemplo por exemplo). Considere* $\leq$ *como a relação de ordem habitual e* $|$ *como a relação de divisibilidade (i.e.* $a|b \triangleq$ *a divide b);*

1. $(\mathbb{N}, \leq)$;
2. $(\mathbb{Z}, \leq)$;
3. $(\mathbb{N}, |)$;
4. $(\mathbb{R}, \leq)$;
5. $(\{2n \mid n \in \mathbb{N}^*\}, |)$
6. $(\{2^n \mid n \in \mathbb{N}^*\}, |)$
7. $\forall C$ *conjunto,* $(\wp(C), \subseteq)$

□

Exercício 30 *Sejam* $(A, \leq_A)$ *e* $(B, \leq_B)$ *dois conjuntos* ***ordenados*** *por ordens largas* ***totais*** *e* ***bem fundadas*** *(diz-se, neste caso, que* $\leq_A$ *e* $\leq_B$ *são boas ordens). Seja* $(A \times B, \leq^{\mathcal{L}})$ *o conjunto* $A \times B$ *ordenado pela ordem lexicográfica* $\leq^{\mathcal{L}}$ *i.e.*

$$(a, b) \leq^{\mathcal{L}} (c, d) \triangleq \begin{cases} (a <_A c) \\ (a = c) \wedge (b \leq_B d) \end{cases}$$

onde $(a <_A c) \triangleq (a \leq_A c) \wedge (a \neq c)$. *Demonstre que* $(A \times B, \leq^{\mathcal{L}})$ *é bem fundado.*

□

Exercício 31 *Sejam* $(A, \leq_A)$ *e* $(B, \leq_B)$ *dois conjuntos* ***ordenados bem fundados****. Seja* $(A \times B, \leq_{A\times B})$ *o conjunto* $A \times B$ *ordenado pela ordem produto* $\leq_{A\times B}$ *i.e.* $(a, b) \leq_{A\times B} (c, d) \triangleq ((a \leq_A c) \wedge (b \leq_B d))$. *Demonstre que* $(A \times B, \leq_{A\times B})$ *é igualmente bem fundado. Para tal considere a definição da noção de ordem bem fundada e verifique que* $\leq_{A\times B}$ *respeita bem esta definição (uma demonstração possível é proceder por contradição).* □

Exercício 32 *Seja* $A^\star$ *o monóide livre gerado a partir do alfabeto* A. *Mostre que* $\forall u, v \in A^\star, u.v = v.u \longleftrightarrow \exists w \in A^\star.\exists p, q \in \mathbb{N}.u = w^p \wedge v = w^q$

□