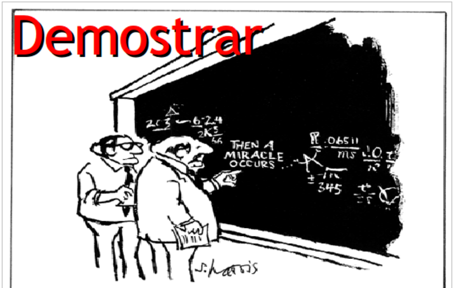
Inducción Estructural

Departamento de Computación,
Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires

Algoritmos y Estructuras de Datos II

¿Para qué?



"I think you should be more explicit here in step two."

Queremos demostrar propiedades que se cumplen en las operaciones de los TADs, estructuras que son construidas recursivamente.

Construcción de un TAD

- Generador(es) base (no reciben instancias del TAD)
- Generador(es) recursivos (reciben instancias del mismo TAD)

Nat

- Único generador base: 0
- Generador recursivo: suc(n)

Secuencia

- Generador base: $<>$
- Generador recursivo: $a \bullet s$

Árbol binario

- Generador base: *nil*
- Generador recursivo: $bin(I, r, D)$

Construcción de un TAD

Polinomio

- Generadores Base

  $X \quad : \longrightarrow \text{polinomio}$

  $\text{Cte} \quad : \text{nat} \longrightarrow \text{polinomio}$

- Generadores recursivos

  $\bullet + \bullet \quad : \text{polinomio} \times \text{polinomio} \longrightarrow \text{polinomio}$

  $\bullet * \bullet \quad : \text{polinomio} \times \text{polinomio} \longrightarrow \text{polinomio}$

Inducción estructural

¿Se acuerdan de Álgebra 1?

Para demostrar que $(\forall n)P(n)$

- Probábamos que valía P(0)
- Asumiendo que valía P(n), probábamos P($n+1$)

- ¿Cuál es el caso base?
- ¿Cuál es el paso inductivo?
- ¿Cuál es la hipótesis inductiva?
- ¿Cuál es la tesis inductiva?

Generalicemos

- En lugar de Nat, un TAD arbitrario.

¿Qué cambia?

Esquemas de inducción

El esquema de la demostración:

- uno o más **casos base**, y
- uno o más **pasos inductivos**
  - cada PI con su **hipótesis inductiva** y su **tesis inductiva**

- ¿Cuántos CB podríamos necesitar?
- ¿Cuántos PI podríamos necesitar?
- ¿Qué pinta podría tener cada CB?
- ¿Qué pinta podría tener cada PI?

Esquemas de inducción

Para demostrar que $(\forall n) P(n)$

- Probábamos que valía P(0)
- Asumiendo que valía P(n), probábamos P($n+1$)

Esquema de inducción en TAD Nat

$$\mathrm{P}(0) \quad \wedge \quad (\forall\, n : nat)\ (\mathrm{P}(n) \Rightarrow \mathrm{P}(suc(n)))$$

Para un TAD más general, llamémosle T

- $\mathrm{P}(GB_1) \ \wedge\ \mathrm{P}(GB_2) \cdots \ \wedge\ \mathrm{P}(GB_k)$
  Probamos los generadores base...
- $\wedge\ (\forall\, t : T)\ \mathrm{P}(t) \Rightarrow \mathrm{P}(GR_1)$
- $\wedge\ \cdots$
- $\wedge\ (\forall\, t : T)\ \mathrm{P}(t) \Rightarrow \mathrm{P}(GR_n)$

... y **todos** los generadores recursivos.

Esquema de inducción

En palabras...

- Probar los constructores que **NO** reciben instancias del mismo TAD.
- Suponer que hay una instancia que lo cumple y probar que al aplicarle un generador recursivo, la propiedad sigue valiendo

Esquema de inducción

¿Cuál es el esquema de inducción en TAD Secuencia?

$\mathrm{P}(\ <>\ )\ \ \wedge\ \ (\forall\, s : secu(\alpha))(P(s) \Rightarrow (\forall\, a : \alpha)(P(a \bullet s)))$

¿Estaría bien plantear así el paso inductivo?

$((\forall\, s : secu(\alpha))\ \mathrm{P}(s)) \Rightarrow (\forall\, s : secu(\alpha))((\forall\, a : \alpha)\ \mathrm{P}(a \bullet s))$ ¡¡¡NO!!!

¿Cambia el esquema si la propiedad es otra?

¡¡¡NO!!! El esquema se mantiene dentro del mismo TAD.

Escribamos los esquemas de inducción de:

Árbol binario

- Generador base: *nil*
- Generador recursivo: $bin(I, r, D)$

Polinomio

- Generadores Base

  $$\begin{array}{lll} X & : & \longrightarrow \text{polinomio} \\ Cte & : & \text{nat} \longrightarrow \text{polinomio} \end{array}$$

- Generadores recursivos

  $$\begin{array}{lll} \bullet + \bullet & : & \text{polinomio} \times \text{polinomio} \longrightarrow \text{polinomio} \\ \bullet * \bullet & : & \text{polinomio} \times \text{polinomio} \longrightarrow \text{polinomio} \end{array}$$

¡Al pizarrón!

Receta

1. Leer y entender la propiedad.
2. (Re)escribirla como un predicado unario.
3. Plantear el esquema de inducción.
4. Demostrar el/los caso(s) base.
5. Demostrar el/los paso(s) inductivo(s).
6. Demostrar los lemas, de haberlos.

Ejercicio 1

Enunciado

Se quiere probar por inducción estructural la siguiente propiedad:
$(\forall\ s : secu(\alpha))\ (long(duplicar(s)) \equiv 2 * long(s))$

longitud

$long : secu(\alpha) \longrightarrow nat$
$l1)\ long(<>)\ \equiv 0$
$l2)\ long(a \bullet s) \equiv 1 + long(s)$

duplicar

$duplicar : secu(\alpha) \longrightarrow secu(\alpha)$
$d1)\ duplicar(<>)\ \equiv <>$
$d2)\ duplicar(a \bullet s) \equiv a \bullet (a \bullet duplicar(s))$

Veamos

Enunciado

Se quiere probar por inducción estructural la siguiente propiedad:
$(\forall\ s : secu(\alpha))\ (long(duplicar(s)) \equiv 2 * long(s))$

Predicado unario

$\mathrm{P}(s) = (\mathrm{long}(\mathrm{duplicar}(\mathrm{s})) \equiv 2{*}\mathrm{long}(\mathrm{s}))$

Esquema de inducción

$\mathrm{P}(\ <>) \quad \wedge \quad (\forall\ s : secu(\alpha))(P(s) \Rightarrow (\forall\ a : \alpha)(P(a \bullet s)))$

Analicemos

Esquema de inducción

$$P(<>) \quad \wedge \quad (\forall\, s : secu(\alpha))(P(s) \Rightarrow (\forall\, a : \alpha)(P(a \bullet s)))$$

Nos interesa distinguir ciertas cosas

- Caso Base: $P(<>)$
- Paso inductivo: $(\forall\, s : secu(\alpha))(P(s) \Rightarrow (\forall\, a : \alpha)(P(a \bullet s)))$
- Hipótesis Inductiva: $P(s)$
- Tesis Inductiva: $(\forall\, a : \alpha)(P(a \bullet s))$

Caso Base

¿Qué queremos probar?

$$\begin{aligned} \mathrm{P}(s) &= (\mathrm{long}(\mathrm{duplicar}(\mathrm{s})) \equiv 2{*}\mathrm{long}(\mathrm{s})) \\ \mathrm{P}(<>) &= (\mathrm{long}(\mathrm{duplicar}(<>)) \equiv 2{*}\mathrm{long}(<>)) \end{aligned}$$

longitud

$long : secu(\alpha) \longrightarrow nat$
$l1)\ long(<>)\ \equiv 0$
$l2)\ long(a \bullet s) \equiv 1 + long(s)$

duplicar

$duplicar : secu(\alpha) \longrightarrow secu(\alpha)$
$d1)\ duplicar(<>)\ \equiv <>$
$d2)\ duplicar(a \bullet s) \equiv a \bullet (a \bullet duplicar(s))$

¡Al pizarrón!

Paso inductivo

¿Qué queremos probar?

$$(\forall\, s : secu(\alpha))(P(s) \Rightarrow (\forall\, a : \alpha)(P(a \bullet s)))$$

$$\begin{aligned} \text{HI: vale P}(s) &= (\text{long}(\text{duplicar}(\text{s})) \equiv 2{*}\text{long}(\text{s})) \\ \text{Probar P}(a \bullet s) &= (\text{long}(\text{duplicar}(a \bullet s))) \equiv 2{*}\text{long}(a \bullet s))) \end{aligned}$$

longitud

$long : secu(\alpha) \longrightarrow nat$
$l1)\ long(<>)\ \equiv 0$
$l2)\ long(a \bullet s) \equiv 1 + long(s)$

duplicar

$duplicar : secu(\alpha) \longrightarrow secu(\alpha)$
$d1)\ duplicar(<>)\ \equiv <>$
$d2)\ duplicar(a \bullet s) \equiv a \bullet (a \bullet duplicar(s))$

¡Al pizarrón!

Ejercicio 2

Enunciado

Se quiere probar por inducción estructural la siguiente propiedad:
$(\forall\ s : secu(nat))\ (\forall\ t : secu(nat))\ (ord(s\ \&\ t) \Rightarrow ord(s))$

ord

$ord : secu(\alpha) \longrightarrow bool$
$or_1)\ ord(<>)\ \equiv true$
$or_2)\ ord(a \bullet s) \equiv \textit{if } \text{vacía?}(s) \textit{ then true else } a < prim(s) \wedge ord(s) \textit{ fi}$

&

$\bullet \,\&\, \bullet\ : secu(\alpha) \times secu(\alpha) \longrightarrow secu(\alpha)$
$\&_1)\ <>\ \&\ t \quad \equiv t$
$\&_2)\ (a \bullet s)\ \&\ t\ \equiv a\ \bullet\ (s\ \&\ t)$

Ejercicio 2

Enunciado

Se quiere probar por inducción estructural la siguiente propiedad:
$(\forall\ s : secu(nat))\ (\forall\ t : secu(nat))\ (ord(s \,\&\, t) \Rightarrow ord(s))$

Predicado unario

$\mathrm{P}(s) = (\forall\ t : secu(nat))\ (ord(s \,\&\, t) \Rightarrow ord(s))$

Esquema de inducción

$\mathrm{P}(\ <>)\ \ \wedge\ \ (\forall\ s : secu(\alpha))(P(s) \Rightarrow (\forall\ a : \alpha)(P(a \bullet s)))$
Es sospechosamente parecido al del ejercicio anterior... ¿Por qué?

Caso Base

¿Qué queremos probar?

$$\mathrm{P}(s) = (\forall\, t : secu(nat))\ (ord(s \,\&\, t) \Rightarrow ord(s))$$
$$\mathrm{P}(<>) = (\forall\, t : secu(nat))\ (ord(<>\ \&\, t) \Rightarrow ord(<>))$$

ord

$ord : secu(\alpha) \longrightarrow bool$
$or_1)\ ord(<>) \equiv true$
$or_2)\ ord(a \bullet s) \equiv if\ \text{vacía?}(s)\ then\ true\ else\ a < prim(s) \wedge ord(s)\ fi$

&

$\bullet \,\&\, \bullet\ : secu(\alpha) \times secu(\alpha) \longrightarrow secu(\alpha)$
$\&_1)\ <> \&\ t \equiv t$
$\&_2)\ (a \bullet s)\ \&\ t \equiv a \bullet (s \,\&\, t)$

¡Al pizarrón!

Propiedades con implicaciones

- ¿Cómo hago para probar que $p \Rightarrow q$?
- Recordamos: $(p \Rightarrow q) \equiv (\neg p \vee q)$
  - Si no vale p, entonces vale la implicación.
  - Si vale p, entonces debe valer q.
- Si lo logro, ¿probé que vale p? ¡NO!
- Si lo logro, ¿probé que vale q? ¡NO!
- ¿Qué demostré, exactamente?

Paso inductivo

¿Qué queremos probar?

$(\forall\, s : secu(\alpha))(P(s) \Rightarrow (\forall\, a : \alpha)(P(a \bullet s)))$

HI: vale P(s) $= (\forall\, t : secu(nat))\ (ord(s \,\&\, t) \Rightarrow ord(s))$

Probar P($a \bullet s$) $= (\forall\, t : secu(nat))\ (ord((a \bullet s) \,\&\, t) \Rightarrow ord(a \bullet s))$

ord

$ord : secu(\alpha) \longrightarrow bool$

$or_1)\ ord(<>) \equiv true$

$or_2)\ ord(a \bullet s) \equiv if$ vacía?(s) $then\ true\ else\ a < prim(s) \wedge ord(s)\ fi$

&

$\bullet \,\&\, \bullet\ : secu(\alpha) \times secu(\alpha) \longrightarrow secu(\alpha)$

$\&_1)\ <> \,\&\, t \equiv t$

$\&_2)\ (a \bullet s) \,\&\, t \equiv a \bullet (s \,\&\, t)$

¡Al pizarrón!

Separación en casos

- ¿Cuándo aparece la necesidad de separar en casos?
  - if G then A else B fi
- ¿Cuáles son las características de una buena separación en casos?
  - Disjunta y completa (por ej: G y $\neg G$)

Lemas

Lemas

Muchas veces necesitamos hacer uso de una propiedad que intuimos es cierta para poder avanzar en la demostración. En estos casos podemos enunciar lemas auxiliares, utilizarlos y dejar su demostración para el final... pero ¡OJO!, salvo que se diga lo contrario, HAY que demostrarlos.

Lema 1

$(\forall\ s : secu(nat))\ \neg \text{vacía?}(s) \Rightarrow (\forall\ t : secu(nat))\ \neg \text{vacía?}(s \& t)$

Lema 2

$(\forall\ s : secu(nat))\ \neg \text{vacía?}(s) \Rightarrow_L (\forall\ t : secu(nat))\ \text{prim}(s) = \text{prim}(s \& t)$

La demostración les queda de tarea

Cuidado con...

- HI que son implicaciones
  - Para utilizar el consecuente, debe valer el antecedente.
- Operaciones con restricciones
  - Hay que asegurarse de que siempre se satisfagan las restricciones.

Propiedades del IF

- Propiedad 1
  IF p THEN q ELSE q FI $\equiv q$
  Si x es par entonces $2x$ es par y si no... también.

- Propiedad 2
  IF p THEN (IF p THEN q ELSE r FI) ELSE t FI $\equiv$
  IF p THEN q ELSE t FI

- Propiedad 3
  F(IF p THEN q ELSE r FI) $\equiv$ IF p THEN F(q) ELSE F(r) FI

Ejercicio 3

Enunciado

Llamaremos *árbol binario estricto* (ABE) a todo árbol binario donde cada nodo tiene 0 ó 2 hijos. En otras palabras, se trata de un árbol binario donde cada nodo ó bien está saturado ó bien es una hoja. Demostrar que en un árbol binario estricto no nil, las hojas son más de 50% de los nodos totales.

Formalmente

$(\forall\ a : ab(\alpha))\ ((\neg \text{nil?}(a) \wedge \text{esEstricto}(\text{a})) \Rightarrow$
$(2 * \text{cantHojas}(a) \geq \text{cantNodos}(a) + 1))$

Ejercicio 3 (cont.)

esEstricto

$$\begin{aligned}
&esEstricto : ab(\alpha) \longrightarrow bool\\
&E1)\ esEstricto(nil) \equiv true\\
&E2)\ esEstricto(bin(I, r, D)) \equiv (\text{nil?}(I) \land \text{nil?}(D)) \lor (\neg\text{nil?}(I) \land \neg\text{nil?}(D)\\
&\qquad \land \text{esEstricto}(I) \land \text{esEstricto}(D))
\end{aligned}$$

cantHojas

$$\begin{aligned}
&cantHojas : ab(\alpha) \longrightarrow nat\\
&H1)\ cantHojas(nil) \equiv 0\\
&H2)\ cantHojas(bin(I, r, D)) \equiv \text{if nil?}(I) \land \text{nil?}(D) \text{ then } 1\\
&\qquad \text{else cantHojas}(I) + \text{cantHojas}(D) \text{ fi}
\end{aligned}$$

cantNodos

$$\begin{aligned}
&cantNodos : ab(\alpha) \longrightarrow nat\\
&N1)\ cantNodos(nil) \equiv 0\\
&N2)\ cantNodos(bin(I, r, D)) \equiv 1 + \text{cantNodos}(I) + \text{cantNodos}(D)
\end{aligned}$$

Veamos

Propiedad

$(\forall\ a : ab(\alpha))((\neg \text{nil?}(a) \wedge \text{esEstricto(a)})$
$\Rightarrow (2 * \text{cantHojas}(a) \geq \text{cantNodos}(a) + 1))$

Predicado unario

$\text{P}(a) = (\neg \text{nil?}(a) \wedge \text{esEstricto(a)}) \Rightarrow (2 * \text{cantHojas}(a) \geq \text{cantNodos}(a) + 1)$

Esquema de inducción

$\text{P}(\ nil) \quad \wedge \quad (\forall\ I, D : ab(\alpha))(P(I) \wedge P(D) \Rightarrow (\forall\ r : \alpha)(P(bin(I, r, D))))$

Ejercicio 3

esEstricto

$$
\begin{aligned}
&\mathit{esEstricto} : \mathit{ab}(\alpha) \longrightarrow \mathit{bool}\\
&E1)\ \mathit{esEstricto}(\mathit{nil}) \equiv \mathit{true}\\
&E2)\ \mathit{esEstricto}(\mathit{bin}(I, r, D)) \equiv (\text{nil?}(I) \wedge \text{nil?}(D)) \vee (\neg\text{nil?}(I) \wedge \neg\text{nil?}(D)\\
&\qquad \wedge \text{esEstricto}(I) \wedge \text{esEstricto}(D))
\end{aligned}
$$

cantHojas

$$
\begin{aligned}
&\mathit{cantHojas} : \mathit{ab}(\alpha) \longrightarrow \mathit{nat}\\
&H1)\ \mathit{cantHojas}(\mathit{nil}) \equiv 0\\
&H2)\ \mathit{cantHojas}(\mathit{bin}(I, r, D)) \equiv \text{if nil?}(I) \wedge \text{nil?}(D) \text{ then } 1\\
&\qquad \text{else cantHojas}(I) + \text{cantHojas}(D) \text{ fi}
\end{aligned}
$$

cantNodos

$$
\begin{aligned}
&\mathit{cantNodos} : \mathit{ab}(\alpha) \longrightarrow \mathit{nat}\\
&N1)\ \mathit{cantNodos}(\mathit{nil}) \equiv 0\\
&N2)\ \mathit{cantNodos}(\mathit{bin}(I, r, D)) \equiv 1 + \text{cantNodos}(I) + \text{cantNodos}(D)
\end{aligned}
$$

¡Al pizarrón!

¿¿Preguntas??