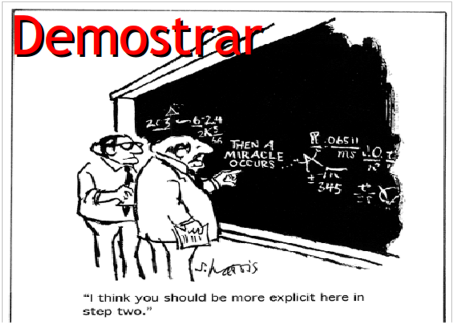
Inducción Estructural

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

Algo 2

¿Para qué?



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

Generadores de un TAD

- Generador(es) Base (no reciben instancias del TAD) $(g_1, \ldots g_k)$
- Generador(es) recursivos (reciben instancias del mismo TAD) $(g_{k+1}, \ldots g_n)$

Generadores de un TAD

- Generador(es) Base (no reciben instancias del TAD) $(g_1, \ldots g_k)$
- Generador(es) recursivos (reciben instancias del mismo TAD) $(g_{k+1}, \ldots g_n)$

Nat

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

Generadores de un TAD

- Generador(es) Base (no reciben instancias del TAD) $(g_1, \ldots g_k)$
- Generador(es) recursivos (reciben instancias del mismo TAD) $(g_{k+1}, \ldots g_n)$

Nat

- Único generador base: 0
- Generador recursivo: suc(n)
- Esquema inductivo: $P(0) \wedge (\forall n)\, \underbrace{P(n)}_{HI} \Rightarrow \underbrace{P(suc(n))}_{TI}$

Generadores de un TAD

- Generador(es) Base (no reciben instancias del TAD) $(g_1, \dots g_k)$
- Generador(es) recursivos (reciben instancias del mismo TAD) $(g_{k+1}, \dots g_n)$

Nat

- Único generador base: 0
- Generador recursivo: suc(n)
- Esquema inductivo: $P(0) \wedge (\forall n) \underbrace{P(n)}_{HI} \Rightarrow \underbrace{P(suc(n))}_{TI}$

Secuencia

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

Generadores de un TAD

- Generador(es) Base (no reciben instancias del TAD) $(g_1, \ldots g_k)$
- Generador(es) recursivos (reciben instancias del mismo TAD) $(g_{k+1}, \ldots g_n)$

Nat

- Único generador base: 0
- Generador recursivo: suc(n)
- Esquema inductivo: $P(0) \wedge (\forall n)\, \underbrace{P(n)}_{HI} \Rightarrow \underbrace{P(suc(n))}_{TI}$

Secuencia

- Generador base: $<>$
- Generador recursivo: $a \bullet s$
- Esquema inductivo: $(\forall s : \mathrm{secu}(\alpha))(\underbrace{P(s)}_{HI} \Rightarrow \underbrace{(\forall a : \alpha) P(a \bullet s)}_{TI})$

Generadores de un TAD

Árbol binario

- Generador base: nil
- Generador recursivo: $bin(A, r, B)$
- Esquema inductivo:

Generadores de un TAD

Árbol binario

- Generador base: *nil*
- Generador recursivo: $bin(A, r, B)$
- Esquema inductivo:
$P(Nil) \wedge ((\forall Izq, Der : Ab(Nat))(\underbrace{(P(Izq) \wedge P(Der))}_{HI} \Rightarrow (\forall N : Nat)\, \underbrace{P(Ab(Izq, N, Der))}_{TI})))$

Generadores de un TAD

Polinomio

- Generadores Base

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

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

- Generadores recursivos

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

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

- Esquema inductivo:

Generadores de un TAD

Polinomio

- Generadores Base

  Cte : nat $\longrightarrow$ polinomio

  X : $\longrightarrow$ polinomio

- Generadores recursivos

  $\bullet + \bullet$: polinomio $\times$ polinomio $\longrightarrow$ polinomio

  $\bullet * \bullet$: polinomio $\times$ polinomio $\longrightarrow$ polinomio

- Esquema inductivo:

  $(\forall n)P(Cte(N)) \wedge P(X) \wedge ((\forall p1, p2 : polinomio)((P(p1) \wedge P(p2)) \Rightarrow P(p1 + p2))) \wedge ((\forall p1, p2 : polinomio)((P(p1) \wedge P(p2)) \Rightarrow P(p1 * p2)))$

Lo más importante

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?

- **¿De qué depende todo eso?**

Lo más importante

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?

- **¿De qué depende todo eso?**
- **¿Depende de la propiedad a demostrar?**

Inducción

Receta

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

Recordemos

Quiero probar $(\forall x)P(x)$

- Puedo olvidarme del cuantificador siempre que no suponga nada en particular sobre x.
- $P(x)$ es el predicado unario.

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

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

Ejercicio 1

Receta

1. Leer y entender la propiedad.
2. Convencerse de que es cierta

Ejercicio 1

Receta

1. Leer y entender la propiedad.
2. Convencerse de que es cierta

$\text{Duplicar}(1 \bullet 2 \bullet <>)$	por d_2
$1 \bullet 1 \bullet \text{Duplicar}(2 \bullet <>)$	nuevamente usando d_2
$1 \bullet 1 \bullet 2 \bullet 2 \bullet \text{Duplicar}(<>)$	y ahora por d_1
$1 \bullet 1 \bullet 2 \bullet 2 \bullet <>$	

Ejercicio 1

Receta

(Re)escribirla como un predicado unario.
Es lo que vamos a demostrar, sin el cuantificador

Ejercicio 1

Receta

(Re)escribirla como un predicado unario.
Es lo que vamos a demostrar, sin el cuantificador

$$(\forall\ s : secu(\alpha))\ (long(duplicar(s)) \equiv 2 * long(s))$$

$$(\forall\ s : secu(\alpha))\ P(s)$$

Ejercicio 1

Receta

(Re)escribirla como un predicado unario.
Es lo que vamos a demostrar, sin el cuantificador

$$(\forall\ s : secu(\alpha))\ (long(duplicar(s)) \equiv 2 * long(s))$$

$$(\forall\ s : secu(\alpha))\ P(s)$$

$$P(s) \equiv (long(duplicar(s)) \equiv 2 * long(s))$$

Ejercicio 1

Receta

Plantear el esquema de inducción.

Casos bases y pasos inductivos

Ejercicio 1

Receta

Plantear el esquema de inducción.
Casos bases y pasos inductivos

1. un generador basico ($<>$)
2. un generador recursivo ($\bullet$), que toma una instancia de secuencia.

Ejercicio 1

Receta

Plantear el esquema de inducción.

Casos bases y pasos inductivos

1. un generador basico ($<>$)
2. un generador recursivo ($\bullet$), que toma una instancia de secuencia.

$$P(<>) \wedge (\forall s : \text{secu}(\alpha))(\underbrace{P(s)}_{HI} \Rightarrow \underbrace{(\forall a : \alpha)P(a \cdot s)}_{TI})$$

Ejercicio 1

1. Demostrar el/los caso(s) base.
2. Demostrar el/los paso(s) inductivo(s).
3. Demostrar los lemas, de haberlos.

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

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

Ejercicio 2

Enunciado

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

ord

$ord : secu(\alpha) \longrightarrow bool$
$ord_1)\ ord(<>)\ \equiv true$
$ord_2)\ ord(a \bullet s) \equiv if\ \text{vacía?}(s)\ then\ true\ else\ a < prim(s) \wedge ord(s)\ fi$

&

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

Ejercicio 2

Receta

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

Ejercicio 2

Receta

1. Leer y entender la propiedad.
2. Convencerse de que es cierta

Ejercicio 2

Receta

1. Leer y entender la propiedad.
2. Convencerse de que es cierta

$$
\begin{array}{ll}
y = 1 \bullet 2 \bullet 3 \bullet <> & x = <> \\
y = 1 \bullet 2 \bullet 3 \bullet <> & x = 1 \bullet <> \\
y = 10 \bullet 2 \bullet 3 \bullet <> & x = <> \\
y = 10 \bullet 2 \bullet 3 \bullet <> & x = 1 \bullet <>
\end{array}
$$

Ejercicio 2

Receta

(Re)escribirla como un predicado unario.

Es lo que vamos a demostrar, sin el cuantificador

Ejercicio 2

Receta

(Re)escribirla como un predicado unario.
Es lo que vamos a demostrar, sin el cuantificador

$$(\forall\ x : secu(nat))\ (\forall\ y : secu(nat))\ (ord(y \,\&\, x) \Rightarrow ord(y))$$

$$(\forall\ s : secu(nat))\ P(s)$$

Ejercicio 2

Receta

(Re)escribirla como un predicado unario.

Es lo que vamos a demostrar, sin el cuantificador

$$(\forall\ x : secu(nat))\ (\forall\ y : secu(nat))\ (ord(y \mathbin{\&} x) \Rightarrow ord(y))$$

$$(\forall\ s : secu(nat))\ P(s)$$

$$P_1(s) \equiv (\forall\ y : secu(nat))\ (ord(y \mathbin{\&} s) \Rightarrow ord(y))$$

$$P_2(s) \equiv (\forall\ x : secu(nat))\ (ord(s \mathbin{\&} x) \Rightarrow ord(s))$$

Ejercicio 2

Receta

Plantear el esquema de inducción.

Casos bases y pasos inductivos

Ejercicio 2

Receta

Plantear el esquema de inducción.
Casos bases y pasos inductivos

1. un generador basico ($<>$)
2. un generador recursivo ($\bullet$), que toma una instancia de secuencia.

Ejercicio 2

Receta

Plantear el esquema de inducción.

Casos bases y pasos inductivos

1. un generador basico ($<>$)
2. un generador recursivo ($\bullet$), que toma una instancia de secuencia.

$$P(<>) \wedge (\forall s : \mathrm{secu}(nat))(\underbrace{P(s)}_{HI} \Rightarrow \underbrace{(\forall a : nat) P(a \cdot s)}_{TI})$$

Ejercicio 2

Receta

Demostrar el/los caso(s) base.

$$P_1(s) \equiv (\forall\ y : secu(nat))\ (ord(y \ \&\ s) \Rightarrow ord(b))$$

$$P_1(<>) \equiv (\forall\ y : secu(nat))\ (ord(y \ \&\ <>) \Rightarrow ord(y))$$

No tenemos ningún axioma que aplicar

Ejercicio 2

Receta

Demostrar el/los caso(s) base.

$$P_2(s) \equiv (\forall\ x : secu(nat))\ (ord(s\ \&\ x) \Rightarrow ord(s))$$

$$P_2(<>) \equiv (\forall\ x : secu(nat))\ (ord(<>\ \ \&\ x) \Rightarrow ord(<>))$$

Podemos aplicar $\&_1$

Ejercicio 2

1. Demostrar el/los caso(s) base.
2. Demostrar el/los paso(s) inductivo(s).
3. Demostrar los lemas, de haberlos.

Enunciado

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

ord

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

&

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

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?
- Ciudado al utilizar las HI
  - Para utilizar el consecuente, debe valer el antecedente.

Cuidado con...

¿Estaría bien plantear así el paso inductivo?

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

Cuidado con...

¿Estaría bien plantear así el paso inductivo?

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

¡¡¡NO!!!

- Porque la HI afirma lo que queremos probar.

Cuidado con...

- Las separaciones 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 $\neg G$)

Cuidado con...

- Operaciones con restricciones.
- ¿Por qué?
  - Porque complican las cosas.
  - Hay que asegurarse de que siempre se satisfagan las restricciones.

Cuidado con...

- Aprenderse esquemas de memoria, traerse esquemas anotados, etc.
  - En los ejercicios de parcial eso suele no servir para nada.
  - Hay que aprender a deducir los esquemas de los generadores.
- El alcance de los cuantificadores en el esquema.
- No saber identificar HI y TI de manera precisa.
- No asumir lo que queremos probar.

Cuidado con...

- Suponer "obvio" cualquier paso de reescritura sin justificarlo.
- Siempre hay que indicar por qué axioma se puede dar el paso.
- Única excepción: las propiedades elementales de Nat (aritmética, asociatividad, conmutatividad, ...)
  - ¿Por qué permitimos eso?
  - Porque para poder usar los axiomas de Nat habría que trabajar con los generadores 0 y suc.

¡Gracias!

¿Preguntas?