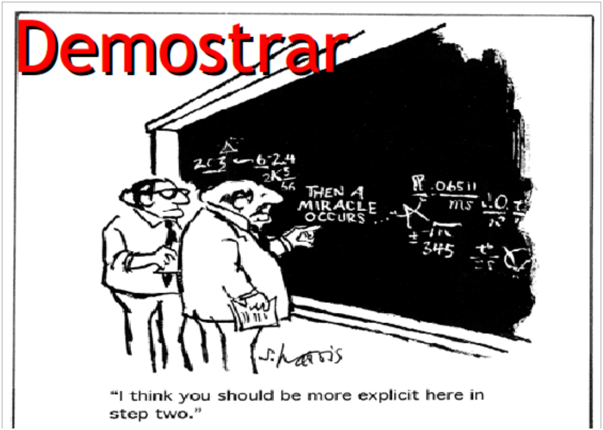
Inducción Estructural

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

Algo 2

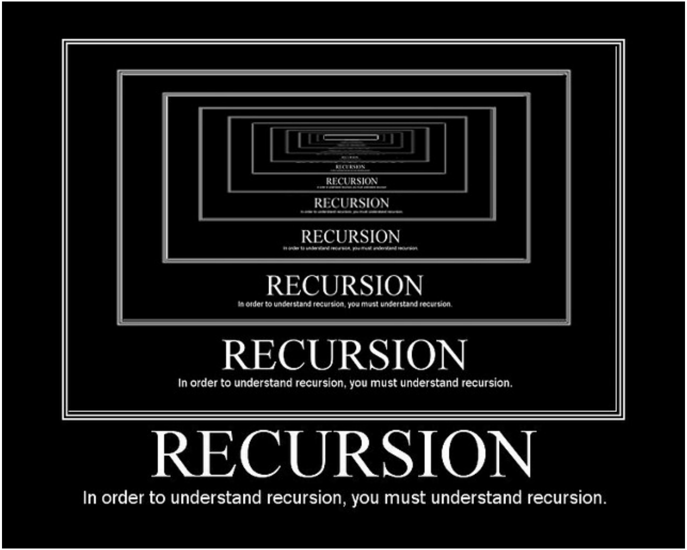
Contenido

¿Para qué?



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

Recursión


RECURSION
In order to understand recursion, you must understand recursion.
RECURSION
In order to understand recursion, you must understand recursion.
RECURSION
In order to understand recursion, you must understand recursion.
RECURSION
RECURSION
RECURSION

Construcción de un TAD

- Constructor(es) Base (no reciben instancias del TAD)
- Constructor(es) recursivos (reciben instancias del mismo TAD)

Construcción de un TAD

- Constructor(es) Base (no reciben instancias del TAD)
- Constructor(es) recursivos (reciben instancias del mismo TAD)

Nat

- Único constructor base: 0
- Constructor recursivo: suc

Construcción de un TAD

- Constructor(es) Base (no reciben instancias del TAD)
- Constructor(es) recursivos (reciben instancias del mismo TAD)

Nat

- Único constructor base: 0
- Constructor recursivo: suc

Secuencia

- Único constructor base: $<>$
- Constructor recursivo: $a \bullet s$

Inducción

¿Se acuerdan de Álgebra 1?

Probábamos que valía P(0), y asumiendo que valía P(n), probábamos P($n+1$). Eso demostraba $(\forall n)\ P(n)$

Inducción

¿Se acuerdan de Álgebra 1?

Probábamos que valía P(0), y asumiendo que valía P(n), probábamos P($n+1$). Eso demostraba $(\forall n)\ P(n)$

Inducción en TAD $\textsc{Nat}$

- P(0)
- $(\forall n\colon \text{nat})\ \mathrm{P}(n) \Rightarrow \mathrm{P}(suc(n)))$

Inducción

¿Se acuerdan de Álgebra 1?

Probábamos que valía P(0), y asumiendo que valía P(n), probábamos P($n+1$). Eso demostraba $(\forall n)\ P(n)$

Inducción en TAD $\textsc{Nat}$

- P(0)
- $(\forall n\colon \text{nat})\ \mathrm{P}(n) \Rightarrow \mathrm{P}(suc(n)))$

Para un TAD más general, llamémoslo T

- **Para cada generador base** $g_b :\to T$ **probar que vale** $P(g_b)$
- **Para cada generador recursivo** $g_r : T \times \ldots \times T \to T$

$$\textbf{probar que vale } (\forall t_1 \ldots t_k : T) \left(\begin{array}{r} (P(t_1) \wedge \ldots \wedge P(t_k)) \quad \Rightarrow \\ P(g_r(t_1, \ldots, t_k)) \quad \end{array} \right)$$

Inducción

En palabras...

- Probar los constructores que **NO** reciben instancias del mismo TAD.
- Para cada constructor recursivo, suponer que vale para todas las instancias del mismo TAD que toma, y luego probar que vale para el constructor aplicado a ellas.

Receta

Receta

1. Convencerse

Receta

Receta

1. Convencerse
2. Escribir predicado unario

Receta

Receta

1. Convencerse
2. Escribir predicado unario
3. Plantear esquema de inducción

Receta

Receta

1. Convencerse
2. Escribir predicado unario
3. Plantear esquema de inducción
4. Probar caso(s) base

Receta

Receta

1. Convencerse
2. Escribir predicado unario
3. Plantear esquema de inducción
4. Probar caso(s) base
5. Probar paso(s) inductivo

Ejercicio 1

Ejercicio

Dada la siguiente axiomatización:

$$\text{duplicar} : \text{secu}(\alpha) \longrightarrow \text{secu}(\alpha)$$
$$\text{duplicar}(<>) \equiv <>$$
$$\text{duplicar}(a \bullet s) \equiv a \bullet (a \bullet \mathit{duplicar}(s))$$

Probar que para toda secuencia 'S', la longitud de 'duplicar(s)' es el doble que la de 'S'

¿Y ahora?

Veamos

Formalmente...

Queremos ver que:

$(\forall s\colon \text{secu}(\alpha))\ (\text{Long}(\text{duplicar}(s)) \equiv 2 * \text{Long}(s))$

Veamos

Formalmente...

Queremos ver que:

$(\forall s\colon \text{secu}(\alpha))\ (\text{Long}(\text{duplicar(s)}) \equiv 2{*}\text{Long(s)})$

Predicado unario

$\text{P}(s) = (\text{Long}(\text{duplicar(s)}) \equiv 2{*}\text{Long(s)})$

Veamos

Formalmente...

Queremos ver que:

$(\forall s\colon \text{secu}(\alpha))\ (\text{Long}(\text{duplicar}(s)) \equiv 2{*}\text{Long}(s))$

Predicado unario

$\text{P}(s) = (\text{Long}(\text{duplicar}(s)) \equiv 2{*}\text{Long}(s))$

Esquema de inducción

$\text{P}(<>) \wedge (\forall s\colon \text{secu}(\alpha))(\text{P}(s) \Rightarrow (\forall a\colon \alpha)\ \text{P}(a \bullet s))$

Analicemos

Nos interesa distinguir ciertas cosas

- Caso Base:

Analicemos

Nos interesa distinguir ciertas cosas

- Caso Base: P(<>)
- Paso inductivo:

Analicemos

Nos interesa distinguir ciertas cosas

- Caso Base: P($<>$)
- Paso inductivo: $(\forall s\colon \text{secu}(\alpha))$ (P(s) $\Rightarrow$ $(\forall a\colon \alpha)$ P($a \bullet s$))
- Hipótesis Inductiva:

Analicemos

Nos interesa distinguir ciertas cosas

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

Analicemos

Nos interesa distinguir ciertas cosas

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

Preguntas

¿Estaría bien plantear así el paso inductivo?

$((\forall s\colon \mathrm{secu}(\alpha))\ \mathrm{P}(s)) \Rightarrow (\forall s\colon \mathrm{secu}(\alpha))\ (\forall a\colon\ \alpha)\ \mathrm{P}(a \bullet s))$

Preguntas

¿Estaría bien plantear así el paso inductivo?

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

¿Cambia el esquema si la propiedad es otra?

Preguntas

¿Estaría bien plantear así el paso inductivo?

$((\forall s\colon \text{secu}(\alpha))\ \text{P}(s)) \Rightarrow (\forall s\colon \text{secu}(\alpha))\ (\forall a\colon \alpha)\ \text{P}(a \bullet s))$ ¡¡¡NO!!!

¿Cambia el esquema si la propiedad es otra?

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

Preguntas

¿Estaría bien plantear así el paso inductivo?

$((\forall s\colon \text{secu}(\alpha))\ \text{P}(s)) \Rightarrow (\forall s\colon \text{secu}(\alpha))\ (\forall a\colon\ \alpha)\ \text{P}(a \bullet s))$ ¡¡¡NO!!!

¿Cambia el esquema si la propiedad es otra?

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

¿Estaría bien agregar esto al esquema de inducción?

$(\forall a\colon\ \alpha)\ (\text{P}(a \bullet <>)))$

Preguntas

¿Estaría bien plantear así el paso inductivo?

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

¿Cambia el esquema si la propiedad es otra?

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

¿Estaría bien agregar esto al esquema de inducción?

$(\forall a\colon\ \alpha)\ (\mathrm{P}(a\bullet <>)))$ ¡¡¡NO!!!

Caso base

¿Qué queremos probar?

$$P(s) \;=\; (Long(duplicar(s)) \;\equiv\; 2 * Long(s))$$

Caso base

¿Qué queremos probar?

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

Caso base

¿Qué queremos probar?

$$P(s) = (Long(duplicar(s)) \equiv 2 * Long(s))$$
$$P(<>) = (Long(duplicar(<>)) \equiv 2 * Long(<>))$$

Duplicar

$$\text{duplicar} : \text{secu}(\alpha) \longrightarrow \text{secu}(\alpha)$$
$$\text{duplicar}(<>) \equiv <>$$
$$\text{duplicar}(a \bullet s) \equiv a \bullet (a \bullet duplicar(s))$$

Longitud

$$\text{long} : \text{secu}(\alpha) \longrightarrow \text{nat}$$
$$\text{long}(<>) \equiv 0$$
$$\text{long}(a \bullet s) \equiv 1 + long(s)$$

¡Al pizarrón!

Paso inductivo

¿Qué queremos probar?

$(\forall s\colon \text{secu}(\alpha))\ (\text{P}(s) \Rightarrow (\forall a\colon\ \alpha)\ \text{P}(a \bullet s))$.

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

Paso inductivo

¿Qué queremos probar?

$(\forall s\colon \mathrm{secu}(\alpha))\ (\mathrm{P}(s) \Rightarrow (\forall a\colon\ \alpha)\ \mathrm{P}(a \bullet s)).$

$$\begin{array}{rcl} HI : vale\ P(s) & = & (Long(duplicar(s)) \equiv 2 * Long(s)) \\ Probar\ P(a \bullet s) & = & (Long(duplicar(a \bullet s)) \equiv 2 * Long(a \bullet s)) \end{array}$$

Duplicar

$$\begin{array}{lcl} \mathrm{duplicar} : \mathrm{secu}(\alpha) & \longrightarrow & \mathrm{secu}(\alpha) \\ \mathrm{duplicar}(<>) & \equiv & <> \\ \mathrm{duplicar}(a \bullet s) & \equiv & a \bullet (a \bullet duplicar(s)) \end{array}$$

Longitud

$$\begin{array}{lcl} \mathrm{long} : \mathrm{secu}(\alpha) & \longrightarrow & \mathrm{nat} \\ \mathrm{long}(<>) & \equiv & 0 \\ \mathrm{long}(a \bullet s) & \equiv & 1 + long(s) \end{array}$$

¡Al pizarrón!

IF

Propiedad 1

$$\textbf{if } {\color{blue}p} \textbf{ then } {\color{red}q} \textbf{ else } {\color{red}q} \textbf{ fi} \equiv$$

IF

Propiedad 1

$\textbf{if } p \textbf{ then } q \textbf{ else } q \textbf{ fi} \equiv q$

Si x es par entonces $2x$ es par y si no... también.

IF

Propiedad 1

$\textbf{if } p \textbf{ then } q \textbf{ else } q \textbf{ fi} \equiv q$

Si x es par entonces $2x$ es par y si no... también.

Propiedad 2

$\textbf{if } p \textbf{ then } (\textbf{if } p \textbf{ then } q \textbf{ else } r \textbf{ fi}) \textbf{ else } t \textbf{ fi} \equiv$

IF

Propiedad 1

$\textbf{if } p \textbf{ then } q \textbf{ else } q \textbf{ fi} \equiv q$

Si x es par entonces $2x$ es par y si no... también.

Propiedad 2

$\textbf{if } p \textbf{ then } (\textbf{if } p \textbf{ then } q \textbf{ else } r \textbf{ fi}) \textbf{ else } t \textbf{ fi} \equiv$
$\textbf{if } p \textbf{ then } q \textbf{ else } t \textbf{ fi}$

IF

Propiedad 1

$\textbf{if } p \textbf{ then } q \textbf{ else } q \textbf{ fi} \equiv q$

Si x es par entonces $2x$ es par y si no... también.

Propiedad 2

$\textbf{if } p \textbf{ then } (\textbf{if } p \textbf{ then } q \textbf{ else } r \textbf{ fi}) \textbf{ else } t \textbf{ fi} \equiv$
$\textbf{if } p \textbf{ then } q \textbf{ else } t \textbf{ fi}$

Propiedad 2 (bis)

$\textbf{if } p \textbf{ then } (\textbf{if } \neg p \textbf{ then } q \textbf{ else } r \textbf{ fi}) \textbf{ else } t \textbf{ fi} \equiv$

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 2 (bis)

if p then (if $\neg p$ then q else r fi) else t fi $\equiv$

if p then r else t fi

Propiedad 3

F(if p then q else r fi) $\equiv$

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 2 (bis)

if p then (if $\neg p$ then q else r fi) else t fi $\equiv$
if p then r else t fi

Propiedad 3

F(if p then q else r fi) $\equiv$ if p then F(q) else F(r) fi

Ejercicio 2 (Más secuencias)

Enunciado

Se quiere probar por inducción estructural la siguiente propiedad:
$(\forall s\colon \text{secu}(nat))\ (\forall t\colon \text{secu}(nat))\ [\ \text{ord?}(s \& t) \Rightarrow \text{ord?}(s)\ ]$

Ejercicio 2 (Más secuencias)

Enunciado

Se quiere probar por inducción estructural la siguiente propiedad:
$(\forall s\colon \text{secu}(nat))\ (\forall t\colon \text{secu}(nat))\ [\ \text{ord?}(s\&t) \Rightarrow \text{ord?}(s)\ ]$

ord

$$
\begin{array}{lcl}
\text{ord} : \text{secu}(\alpha) & \longrightarrow & \text{bool} \\
\text{ord}(<>) & \equiv & \textit{true} \\
\text{ord}(a \bullet s) & \equiv & \textbf{if } \text{vacía?}(s) \textbf{ then } \textit{true} \textbf{ else } a < \text{prim}(s) \wedge \text{ord?}(s) \textbf{ fi}
\end{array}
$$

&

$$
\begin{array}{lcl}
\bullet\&\bullet : \text{secu}(\alpha) \times \text{secu}(\alpha) & \longrightarrow & \text{secu}(\alpha) \\
<> \&t & \equiv & t \\
(a \bullet s)\ \&t & \equiv & a \bullet (s\&t)
\end{array}
$$

Ejemplos

ord

- $\mathrm{ord}([2, 1, 3, 6, 17]) \equiv$

Ejemplos

ord

- ord([2, 1, 3, 6, 17]) $\equiv$ *false*
- ord([]) $\equiv$

Ejemplos

ord

- $\text{ord}([2, 1, 3, 6, 17]) \equiv \textit{false}$
- $\text{ord}([]) \equiv \textit{true}$
- $\text{ord}([5, 10, 14]) \equiv$

Ejemplos

ord

- $\mathrm{ord}([2, 1, 3, 6, 17]) \equiv \mathit{false}$
- $\mathrm{ord}([]) \equiv \mathit{true}$
- $\mathrm{ord}([5, 10, 14]) \equiv \mathit{true}$

Ejemplos

ord

- $\text{ord}([2, 1, 3, 6, 17]) \equiv \textit{false}$
- $\text{ord}([]) \equiv \textit{true}$
- $\text{ord}([5, 10, 14]) \equiv \textit{true}$

&

- $[]\&[1] \equiv$

Ejemplos

ord

- $\text{ord}([2,\ 1,\ 3,\ 6,\ 17]) \equiv \textit{false}$
- $\text{ord}([]) \equiv \textit{true}$
- $\text{ord}([5,\ 10,\ 14]) \equiv \textit{true}$

&

- $[]\&[1] \equiv [1]$
- $[31, 8]\&[2, 6, 17] \equiv$

Ejemplos

ord

- $\text{ord}([2,\ 1,\ 3,\ 6,\ 17]) \equiv \textit{false}$
- $\text{ord}([]) \equiv \textit{true}$
- $\text{ord}([5,\ 10,\ 14]) \equiv \textit{true}$

&

- $[]\&[1] \equiv [1]$
- $[31, 8]\&[2, 6, 17] \equiv [31, 8, 2, 6, 17]$

Planteo

Predicado unario a demostrar

Planteo

Predicado unario a demostrar

$\mathrm{P}(s) = (\forall t\colon \mathrm{secu}(nat))\ [\ \mathrm{ord?}(s\&t) \Rightarrow \mathrm{ord?}(s)\ ]$

Planteo

Predicado unario a demostrar

$P(s) = (\forall t\colon \text{secu}(nat))\ [\ \text{ord?}(s \& t) \Rightarrow \text{ord?}(s)\ ]$

Esquema de inducción

Planteo

Predicado unario a demostrar

$P(s) = (\forall t\colon \text{secu}(nat))\ [\ \text{ord?}(s \& t) \Rightarrow \text{ord?}(s)\ ]$

Esquema de inducción

$P(<>) \wedge (\forall s\colon \text{secu}(\alpha))(P(s) \Rightarrow (\forall a\colon \alpha)\ P(a \bullet s))$

Planteo

Predicado unario a demostrar

$\mathrm{P}(s) = (\forall t\colon \mathrm{secu}(nat))\ [\ \mathrm{ord?}(s \& t) \Rightarrow \mathrm{ord?}(s)\ ]$

Esquema de inducción

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

Es sospechosamente parecido al del ejercicio anterior... ¿Por qué?

Caso Base

$\mathrm{P}(<>) = (\forall t\colon \mathrm{secu}(nat))\ [\ \mathrm{ord?}(<> \& t) \Rightarrow \mathrm{ord?}(<>)\ ]$

Planteo

Predicado unario a demostrar

$P(s) = (\forall t\colon \text{secu}(nat))\ [\ \text{ord?}(s\&t) \Rightarrow \text{ord?}(s)\ ]$

Esquema de inducción

$P(<>) \wedge (\forall s\colon \text{secu}(\alpha))(P(s) \Rightarrow (\forall a\colon \alpha)\ P(a \bullet s))$

Es sospechosamente parecido al del ejercicio anterior... ¿Por qué?

Caso Base

$P(<>) = (\forall t\colon \text{secu}(nat))\ [\ \text{ord?}(<> \&t) \Rightarrow \text{ord?}(<>)\ ]$

- Uno podría ponerse a ver qué pasa con el antecedente, o... ?

Planteo

Predicado unario a demostrar

$P(s) = (\forall t\colon \text{secu}(nat))\ [\ \text{ord?}(s\&t) \Rightarrow \text{ord?}(s)\ ]$

Esquema de inducción

$P(<>) \wedge (\forall s\colon \text{secu}(\alpha))(P(s) \Rightarrow (\forall a\colon \alpha)\ P(a \bullet s))$

Es sospechosamente parecido al del ejercicio anterior... ¿Por qué?

Caso Base

$P(<>) = (\forall t\colon \text{secu}(nat))\ [\ \text{ord?}(<> \&t) \Rightarrow \text{ord?}(<>)\ ]$

- Uno podría ponerse a ver qué pasa con el antecedente, o... ?
- Recordemos que $p \Rightarrow q = q \vee \neg p$. Ahora, qué pasa con ord?($<>$)

Paso inductivo

¿Qué queremos probar?

Dada una secuencia s : secu(α) arbitraria:
HI: vale $\mathrm{P}(s) = (\forall t\colon \mathrm{secu}(nat))$ [$\mathrm{ord?}(s\&t) \Rightarrow \mathrm{ord?}(s)$]
Probar $\mathrm{P}(a \bullet s) = (\forall t\colon \mathrm{secu}(nat))$ [$\mathrm{ord?}((a \bullet s)\&t) \Rightarrow \mathrm{ord?}(a \bullet s)$]

Paso inductivo

¿Qué queremos probar?

Dada una secuencia s : secu(α) arbitraria:

HI: vale $\mathrm{P}(s) = (\forall t\colon \mathrm{secu}(nat))\ [\ \mathrm{ord?}(s \& t) \Rightarrow \mathrm{ord?}(s)\ ]$

Probar $\mathrm{P}(a \bullet s) = (\forall t\colon \mathrm{secu}(nat))\ [\ \mathrm{ord?}((a \bullet s) \& t) \Rightarrow \mathrm{ord?}(a \bullet s)\ ]$

ord

$\mathrm{ord} : \mathrm{secu}(\alpha) \longrightarrow \mathrm{bool}$

$\mathrm{ord}(<>) \equiv \mathit{true}$

$\mathrm{ord}(a \bullet s) \equiv \textbf{if } \mathrm{vacía?}(s) \textbf{ then } \mathit{true} \textbf{ else } a < \mathrm{prim}(s) \wedge \mathrm{ord?}(s) \textbf{ fi}$

&

$\bullet \& \bullet : \mathrm{secu}(\alpha) \times \mathrm{secu}(\alpha) \longrightarrow \mathrm{secu}(\alpha)$

$<> \& t \equiv t$

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

¡Al pizarrón!

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.

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

$\neg\text{vacía?}(s) \Rightarrow \neg\text{vacía?}(s \& t)$

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

$\neg\text{vacía?}(s) \Rightarrow \neg\text{vacía?}(s \& t)$

Lema 2

$\neg\text{vacía?}(s) \Rightarrow_{\mathrm{L}} \text{prim}(s) = \text{prim}(s \& t)$

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

$\neg\text{vacía?}(s) \Rightarrow \neg\text{vacía?}(s\&t)$

Lema 2

$\neg\text{vacía?}(s) \Rightarrow_{\mathrm{L}} \text{prim}(s) = \text{prim}(s\&t)$

La demostración les queda de tarea

$[a, b)$

¿¿Intervalo??

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 las hojas son más de 50% de los nodos totales.

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 las hojas son más de 50% de los nodos totales.

Formalmente

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

esEstricto

$$\begin{aligned}
&\mathrm{esEstricto} : \mathrm{ab}(\alpha) \longrightarrow \mathrm{bool}\\
&\mathrm{esEstricto}(nil) \equiv true\\
&\mathrm{esEstricto}(\mathrm{bin}(i, r, d)) \equiv (\mathrm{nil?}(i) \wedge \mathrm{nil?}(d)) \vee (\neg \mathrm{nil?}(i) \wedge \neg \mathrm{nil?}(d) \wedge \\
&\qquad \mathrm{esEstricto}(i) \wedge \mathrm{esEstricto}(d))
\end{aligned}$$

Ejercicio 3 (cont.)

cantHojas

$$
\begin{array}{lcl}
\text{cantHojas} : \text{ab}(\alpha) & \longrightarrow & \text{nat} \\
\text{cantHojas}(\mathit{nil}) & \equiv & 0 \\
\text{cantHojas}(\text{bin}(i, r, d)) & \equiv & \textbf{if } \text{nil?}(i) \wedge \text{nil?}(d) \textbf{ then} \\
& & \quad 1 \\
& & \textbf{else} \\
& & \quad \text{cantHojas}(i) + \text{cantHojas}(d) \\
& & \textbf{fi}
\end{array}
$$

cantNodos

$$
\begin{array}{lcl}
\text{cantNodos} : \text{ab}(\alpha) & \longrightarrow & \text{nat} \\
\text{cantNodos}(\mathit{nil}) & \equiv & 0 \\
\text{cantNodos}(\text{bin}(i, r, d)) & \equiv & 1 + \text{cantNodos}(i) + \text{cantNodos}(d)
\end{array}
$$

Planteo

Predicado unario a demostrar

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

Planteo

Predicado unario a demostrar

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

Esquema de inducción

Ahora tenemos un generador (*bin*) que recibe más de una instancia del tipo... ¿Modifica en algo la situación? ¿Sobre qué hacemos H.I.?

Planteo

Predicado unario a demostrar

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

Esquema de inducción

Ahora tenemos un generador (*bin*) que recibe más de una instancia del tipo... ¿Modifica en algo la situación? ¿Sobre qué hacemos H.I.?

$\mathrm{P}(nil) \wedge (\forall\ i, d : \mathrm{ab}(\alpha))(\ (\mathrm{P}(i) \wedge \mathrm{P}(d)) \Rightarrow (\forall\ r : \alpha)\ \mathrm{P}(\mathrm{bin}(i,r,d))$

Planteo

Predicado unario a demostrar

$P(a) = (\neg nil?(a) \land esEstricto(a)) \Rightarrow (2 * cantHojas(a) \geq cantNodos(a) + 1)$

Esquema de inducción

Ahora tenemos un generador (*bin*) que recibe más de una instancia del tipo... ¿Modifica en algo la situación? ¿Sobre qué hacemos H.I.?

$P(nil) \land (\forall\ i,\ d : ab(\alpha))(\ (P(i) \land P(d)) \Rightarrow (\forall\ r : \alpha)\ P(bin(i,r,d))$

Caso Base

P(*nil*) . ¿Qué dice el predicado unario? ¿Qué pasa con una implicación cuando el antecedente es falso?

Paso inductivo

¿Qué queremos probar?

HI: P(a) = (¬nil?(a) ∧ esEstricto(a))⇒(2 * cantHojas(a) ≥ cantNodos(a) + 1) vale para i y d.

Paso inductivo

¿Qué queremos probar?

HI: P(a) = (¬nil?(a) ∧ esEstricto(a))⇒(2 * cantHojas(a) ≥ cantNodos(a) + 1) vale para i y d.

Probar(∀ r : α) P(bin(i,r,d)), es decir que queremos probar que:

(¬nil?(bin(i,r,d)) ∧ esEstricto(bin(i,r,d))) ⇒
(2 * cantHojas(bin(i,r,d)) ≥ cantNodos(bin(i,r,d)) + 1)

Paso inductivo

¿Qué queremos probar?

HI: P(a) $=$ ($\neg$nil?(a) $\wedge$ esEstricto(a))$\Rightarrow$(2 * cantHojas(a) $\geq$ cantNodos(a) $+$ 1) vale para i y d.

Probar($\forall\ r : \alpha$) P(bin(i,r,d)), es decir que queremos probar que:

($\neg$nil?(bin(i,r,d)) $\wedge$ esEstricto(bin(i,r,d))) $\Rightarrow$
(2 * cantHojas(bin(i,r,d)) $\geq$ cantNodos(bin(i,r,d)) $+$ 1)

Preguntas...

- ¿Puede $\neg$nil?(bin(i,r,d)) ser falso?

Paso inductivo

¿Qué queremos probar?

HI: P(a) $=$ ($\neg$nil?(a) $\wedge$ esEstricto(a))$\Rightarrow$(2 * cantHojas(a) $\geq$ cantNodos(a) $+$ 1) vale para i y d.

Probar($\forall\ r : \alpha$) P(bin(i,r,d)), es decir que queremos probar que:

($\neg$nil?(bin(i,r,d)) $\wedge$ esEstricto(bin(i,r,d))) $\Rightarrow$
(2 * cantHojas(bin(i,r,d)) $\geq$ cantNodos(bin(i,r,d)) $+$ 1)

Preguntas...

- ¿Puede $\neg$nil?(bin(i,r,d)) ser falso?
- ¿Qué pasa si bin(i,r,d) no es estricto?

Paso inductivo

¿Qué queremos probar?

HI: P(a) $=$ ($\neg$nil?(a) $\wedge$ esEstricto(a))$\Rightarrow$(2 * cantHojas(a) $\geq$ cantNodos(a) $+$ 1) vale para i y d.

Probar($\forall\ r : \alpha$) P(bin(i,r,d)), es decir que queremos probar que:

($\neg$nil?(bin(i,r,d)) $\wedge$ esEstricto(bin(i,r,d))) $\Rightarrow$
(2 * cantHojas(bin(i,r,d)) $\geq$ cantNodos(bin(i,r,d)) $+$ 1)

Preguntas...

- ¿Puede $\neg$nil?(bin(i,r,d)) ser falso?
- ¿Qué pasa si bin(i,r,d) no es estricto? Para continuar la demostración podemos asumirlo también como hipótesis (si no, toda la expresión es verdadera trivialmente)

Paso inductivo

¿Qué queremos probar?

HI: P(a) = ($\neg$nil?(a) $\wedge$ esEstricto(a))$\Rightarrow$(2 * cantHojas(a) $\geq$ cantNodos(a) + 1) vale para i y d.

Probar($\forall$ r : α) P(bin(i,r,d)), es decir que queremos probar que:

($\neg$nil?(bin(i,r,d)) $\wedge$ esEstricto(bin(i,r,d))) $\Rightarrow$
(2 * cantHojas(bin(i,r,d)) $\geq$ cantNodos(bin(i,r,d)) + 1)

Preguntas...

- ¿Puede $\neg$nil?(bin(i,r,d)) ser falso?
- ¿Qué pasa si bin(i,r,d) no es estricto? Para continuar la demostración podemos asumirlo también como hipótesis (si no, toda la expresión es verdadera trivialmente)
- Nos queda ver que (2 * cantHojas(bin(i,r,d)) $\geq$ cantNodos(bin(i,r,d)) + 1)

¡Al pizarrón!

Paso inductivo (cont.)

esEstricto

$$
\begin{aligned}
\text{esEstricto} &: \text{ab}(\alpha) \longrightarrow \text{bool} \\
\text{esEstricto}(\textit{nil}) &\equiv \textit{true} \\
\text{esEstricto}(\text{bin}(i, r, d)) &\equiv (\text{nil?}(i) \wedge \text{nil?}(d)) \vee (\neg\text{nil?}(i) \wedge \neg\text{nil?}(d) \wedge \text{esEstricto}(i) \wedge \\
&\quad\ \text{esEstricto}(d))
\end{aligned}
$$

cantHojas

$$
\begin{aligned}
\text{cantHojas} &: \text{ab}(\alpha) \longrightarrow \text{nat} \\
\text{cantHojas}(\textit{nil}) &\equiv 0 \\
\text{cantHojas}(\text{bin}(i, r, d)) &\equiv \textbf{if } \text{nil?}(i) \wedge \text{nil?}(d) \textbf{ then} \\
&\qquad 1 \\
&\quad\ \textbf{else} \\
&\qquad \text{cantHojas}(i) + \text{cantHojas}(d) \\
&\quad\ \textbf{fi}
\end{aligned}
$$

cantNodos

$$
\begin{aligned}
\text{cantNodos} &: \text{ab}(\alpha) \longrightarrow \text{nat} \\
\text{cantNodos}(\textit{nil}) &\equiv 0 \\
\text{cantNodos}(\text{bin}(i, r, d)) &\equiv 1 + \text{cantNodos}(i) + \text{cantNodos}(d)
\end{aligned}
$$

Conclusiones

Conclusiones

- Convencerse de que la propiedad vale y analizar por qué vale antes de encarar el ejercicio
- Seguir la estructura del esquema de inducción

Conclusiones

Conclusiones

- Convencerse de que la propiedad vale y analizar por qué vale antes de encarar el ejercicio
- Seguir la estructura del esquema de inducción
- Ser ordenado en la separación en casos

Conclusiones

Conclusiones

- Convencerse de que la propiedad vale y analizar por qué vale antes de encarar el ejercicio
- Seguir la estructura del esquema de inducción
- Ser ordenado en la separación en casos
- Si hay una implicación, probarla suponiendo que el antecedente es verdadero

Conclusiones

Conclusiones

- Convencerse de que la propiedad vale y analizar por qué vale antes de encarar el ejercicio
- Seguir la estructura del esquema de inducción
- Ser ordenado en la separación en casos
- Si hay una implicación, probarla suponiendo que el antecedente es verdadero

¿Preguntas?

¿¿Preguntas??

Eso es todo amigos