Inducción estructural

Clase 2

IIC 1253

Prof. Pedro Bahamondes

Outline

Equivalencia de principios de inducción

Teorema

Las siguientes condiciones son equivalentes:

1. Principio del buen orden.
2. Principio de inducción simple.
3. Principio de inducción fuerte.

Equivalencia de principios de inducción

Teorema

Las siguientes condiciones son equivalentes:

1. Principio del buen orden.
2. Principio de inducción simple.
3. Principio de inducción fuerte.

La demostración 1. $\Rightarrow$ 2. quedó propuesta y
se encuentra desarrollada en las diapos de la clase pasada.

Un tipo especial de definición

PIS (Primera formulación)

Para A un subconjunto de $\mathbb{N}$. Si se cumple que

1. $0 \in A$
2. Si $n \in A$, entonces $n + 1 \in A$

entonces $A = \mathbb{N}$.

Un tipo especial de definición

PIS (Primera formulación)

Para A un subconjunto de $\mathbb{N}$. Si se cumple que

1. $0 \in A$
2. Si $n \in A$, entonces $n + 1 \in A$

entonces $A = \mathbb{N}$.

Nos sugiere una **definición inductiva** de $\mathbb{N}$

Un tipo especial de definición

PIS (Primera formulación)

Para A un subconjunto de $\mathbb{N}$. Si se cumple que

1. $0 \in A$
2. Si $n \in A$, entonces $n + 1 \in A$

entonces $A = \mathbb{N}$.

Nos sugiere una **definición inductiva** de $\mathbb{N}$

Notemos que posee

- Elemento base (el cero)
- Operador para construir nuevos elementos (operador sucesor)

Definición inductiva de $\mathbb{N}$

"*Definición*" (naturales)

El conjunto de los números **naturales** $\mathbb{N}$ se define según

1. $0 \in \mathbb{N}$
2. Si $n \in \mathbb{N}$, entonces $n + 1 \in \mathbb{N}$

Definición inductiva de $\mathbb{N}$

"*Definición*" (naturales)

El conjunto de los números **naturales** $\mathbb{N}$ se define según

1. $0 \in \mathbb{N}$
2. Si $n \in \mathbb{N}$, entonces $n + 1 \in \mathbb{N}$

¿Es suficiente? ¿Es el único conjunto que cumple ambas reglas?

Definición inductiva de $\mathbb{N}$

Definición (naturales)

El conjunto de los números **naturales** $\mathbb{N}$ es el **menor conjunto** que cumple

1. $0 \in \mathbb{N}$
2. Si $n \in \mathbb{N}$, entonces $n + 1 \in \mathbb{N}$

Esta **regla de exclusión** exige solo elementos formados con las reglas 1. y 2.

Más allá de $\mathbb{N}$

Con esta idea pudimos definir **inductivamente** a los naturales

- Caso base
- Operador sucesor

Más allá de $\mathbb{N}$

Con esta idea pudimos definir **inductivamente** a los naturales

- Caso base
- Operador sucesor

¿Se pueden definir otros conjuntos de forma inductiva?

Objetivos de la clase

- Comprender definiciones inductivas
- Definir operadores inductivamente
- Demostrar propiedades mediante inducción estructural

Outline

Definiciones inductivas

Estrategia

Para **definir inductivamente** un conjunto necesitamos:

1. Un conjunto (no necesariamente finito) de elementos base, que se supondrá que inicialmente pertenecen al conjunto que se quiere definir.
2. Un conjunto finito de reglas de construcción de nuevos elementos del conjunto a partir de elementos que ya están en él.
3. Establecer que el conjunto es el menor que cumple las reglas.

Definiciones inductivas

Estrategia

Para **definir inductivamente** un conjunto necesitamos:

1. Un conjunto (no necesariamente finito) de elementos base, que se supondrá que inicialmente pertenecen al conjunto que se quiere definir.
2. Un conjunto finito de reglas de construcción de nuevos elementos del conjunto a partir de elementos que ya están en él.
3. Establecer que el conjunto es el menor que cumple las reglas.

Pueden haber infinitos casos base y más de una regla recursiva

Definiciones inductivas

Ejemplo

El conjunto de los **números pares** es el menor conjunto tal que

1. El 0 es un número par.
2. Si n es número par, $n + 2$ es un número par.

Definiciones inductivas

Ejemplo

El conjunto de los **números pares** es el menor conjunto tal que

1. El 0 es un número par.
2. Si n es número par, $n + 2$ es un número par.

¿Podemos definir inductivamente algo que no sea un número?

Definiciones inductivas

Definición ($\mathcal{L}_{\mathbb{N}}$)

El conjunto $\mathcal{L}_{\mathbb{N}}$ es el menor conjunto que cumple las siguientes reglas:

1. $\varnothing \in \mathcal{L}_{\mathbb{N}}$.
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces $L \to k \in \mathcal{L}_{\mathbb{N}}$.

Definiciones inductivas

Definición ($\mathcal{L}_{\mathbb{N}}$)

El conjunto $\mathcal{L}_{\mathbb{N}}$ es el menor conjunto que cumple las siguientes reglas:

1. $\varnothing \in \mathcal{L}_{\mathbb{N}}$.
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces $L \to k \in \mathcal{L}_{\mathbb{N}}$.

¿Qué representan los elementos de $\mathcal{L}_{\mathbb{N}}$?

Ejemplo

Los siguientes son elementos de $\mathcal{L}_{\mathbb{N}}$

- $\varnothing$
- $\varnothing \to 6$ o análogamente, $\to 6$ (omitiremos $\varnothing$ cuando hay más elementos)
- $\to 6 \to 5 \to 6 \to 0$

Definiciones inductivas

Definición (listas enlazadas)

El conjunto de las **listas enlazadas** sobre los naturales $\mathcal{L}_{\mathbb{N}}$ es el menor conjunto que cumple las siguientes reglas:

1. $\varnothing \in \mathcal{L}_{\mathbb{N}}$.
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces $L \to k \in \mathcal{L}_{\mathbb{N}}$.

Definiciones inductivas

Definición (listas enlazadas)

El conjunto de las **listas enlazadas** sobre los naturales $\mathcal{L}_{\mathbb{N}}$ es el menor conjunto que cumple las siguientes reglas:

1. $\varnothing \in \mathcal{L}_{\mathbb{N}}$.
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces $L \to k \in \mathcal{L}_{\mathbb{N}}$.

El operador 2. para $\mathcal{L}_{\mathbb{N}}$ es "*agregar flechita y natural al final de una lista*"

Definiciones inductivas

Además de conjuntos, podemos definir **operaciones o funciones** sobre elementos de conjuntos recursivos

Definiciones inductivas

Además de conjuntos, podemos definir **operaciones o funciones** sobre elementos de conjuntos recursivos

Ejemplo

El operador factorial se define sobre $\mathbb{N}$ según

1. $0! = 1$
2. $(n+1)! = (n+1) \cdot n!$

Definiciones inductivas

Además de conjuntos, podemos definir **operaciones o funciones** sobre elementos de conjuntos recursivos

Ejemplo

El operador factorial se define sobre $\mathbb{N}$ según

1. $0! = 1$
2. $(n+1)! = (n+1) \cdot n!$

Además de operadores, ¿se pueden definir propiedades?

Definiciones inductivas

¿Cuándo dos listas enlazadas son iguales?

Definiciones inductivas

¿Cuándo dos listas enlazadas son iguales?

1. Si alguna es vacía, son iguales si y solo si la otra también es vacía
2. Si ninguna es vacía, entonces estamos en un escenario

   $$L_1 \to k_1 \quad \text{versus} \quad L_2 \to k_2$$

   En este caso, resulta natural considerar

   $$L_1 \to k_1 = L_2 \to k_2 \quad \text{si y solo si} \quad L_1 = L_2 \text{ y } k_1 = k_2$$

Definiciones inductivas

¿Cuándo dos listas enlazadas son iguales?

1. Si alguna es vacía, son iguales si y solo si la otra también es vacía
2. Si ninguna es vacía, entonces estamos en un escenario

$$L_1 \to k_1 \quad \text{versus} \quad L_2 \to k_2$$

   En este caso, resulta natural considerar

$$L_1 \to k_1 = L_2 \to k_2 \quad \text{si y solo si} \quad L_1 = L_2 \text{ y } k_1 = k_2$$

Es decir, la **igualdad de listas** se puede definir a partir de la def. de $\mathcal{L}_{\mathbb{N}}$

Solo nos falta ser capaces de **demostrar** propiedades inductivas

Outline

Demostración de propiedades inductivas

Consideremos una lista $L \in \mathcal{L}_{\mathbb{N}}$ y la propiedad

$$P(L): \quad L \text{ tiene el mismo número de flechas que de elementos}$$

¿Cómo abordamos esta demostración?

Inducción estructural

Principio de Inducción estructural

Sea A un conjunto definido inductivamente y P una propiedad sobre los elementos de A. Si se cumple que:

Inducción estructural

Principio de Inducción estructural

Sea A un conjunto definido inductivamente y P una propiedad sobre los elementos de A. Si se cumple que:

1. Todos los elementos base de A cumplen la propiedad P,

Inducción estructural

Principio de Inducción estructural

Sea A un conjunto definido inductivamente y P una propiedad sobre los elementos de A. Si se cumple que:

1. Todos los elementos base de A cumplen la propiedad P,
2. Para cada regla de construcción, si la regla se aplica sobre elementos en A que cumplen la propiedad P, entonces los elementos producidos por la regla también cumplen la propiedad P

Inducción estructural

Principio de Inducción estructural

Sea A un conjunto definido inductivamente y P una propiedad sobre los elementos de A. Si se cumple que:

1. Todos los elementos base de A cumplen la propiedad P,
2. Para cada regla de construcción, si la regla se aplica sobre elementos en A que cumplen la propiedad P, entonces los elementos producidos por la regla también cumplen la propiedad P

entonces todos los elementos en A cumplen la propiedad P.

Inducción estructural

Principio de Inducción estructural

Sea A un conjunto definido inductivamente y P una propiedad sobre los elementos de A. Si se cumple que:

1. Todos los elementos base de A cumplen la propiedad P,
2. Para cada regla de construcción, si la regla se aplica sobre elementos en A que cumplen la propiedad P, entonces los elementos producidos por la regla también cumplen la propiedad P

entonces todos los elementos en A cumplen la propiedad P.

¡El PIS es un caso particular de este principio!

Inducción estructural

Ejemplo

$$P(L): \quad L \text{ tiene el mismo número de flechas que de elementos}$$

Inducción estructural

Ejemplo

$$P(L): \quad L \text{ tiene el mismo número de flechas que de elementos}$$

BI: El único caso base es la lista vacía $\varnothing$, la cual no tiene flechas ni elementos, y por lo tanto $P(\varnothing)$ es verdadera.

Inducción estructural

Ejemplo

$$P(L): \quad L \text{ tiene el mismo número de flechas que de elementos}$$

BI: El único caso base es la lista vacía $\varnothing$, la cual no tiene flechas ni elementos, y por lo tanto $P(\varnothing)$ es verdadera.

HI: Supongamos que una lista cualquiera L cumple $P(L)$, es decir, tiene exactamente la misma cantidad de flechas que de elementos.

Inducción estructural

Ejemplo

$$P(L): \quad L \text{ tiene el mismo número de flechas que de elementos}$$

BI: El único caso base es la lista vacía $\varnothing$, la cual no tiene flechas ni elementos, y por lo tanto $P(\varnothing)$ es verdadera.

HI: Supongamos que una lista cualquiera L cumple $P(L)$, es decir, tiene exactamente la misma cantidad de flechas que de elementos.

¿Qué elemento tomamos para la **TI**?

Inducción estructural

Ejemplo

HI: Supongamos que una lista cualquiera L cumple $P(L)$, es decir, tiene exactamente la misma cantidad de flechas que de elementos.

TI: Debemos demostrar que $P(L \to k)$ es verdadero, es decir, que $L \to k$ tiene tantas flechas como elementos, con $k \in \mathbb{N}$. Es claro que $L \to k$ tiene exactamente una flecha y un elemento más que L. Por HI, sabemos que L tiene la misma cantidad de flechas y de elementos, y por lo tanto $P(L \to k)$ es verdadera.

Inducción estructural

Ejemplo

HI: Supongamos que una lista cualquiera L cumple $P(L)$, es decir, tiene exactamente la misma cantidad de flechas que de elementos.

TI: Debemos demostrar que $P(L \to k)$ es verdadero, es decir, que $L \to k$ tiene tantas flechas como elementos, con $k \in \mathbb{N}$. Es claro que $L \to k$ tiene exactamente una flecha y un elemento más que L. Por HI, sabemos que L tiene la misma cantidad de flechas y de elementos, y por lo tanto $P(L \to k)$ es verdadera.

Por inducción estructural se sigue que todas las listas en $\mathcal{L}_{\mathbb{N}}$ tienen la misma cantidad de flechas que de elementos. $\square$

Inducción estructural

Ejemplo

HI: Supongamos que una lista cualquiera L cumple $P(L)$, es decir, tiene exactamente la misma cantidad de flechas que de elementos.

TI: Debemos demostrar que $P(L \to k)$ es verdadero, es decir, que $L \to k$ tiene tantas flechas como elementos, con $k \in \mathbb{N}$. Es claro que $L \to k$ tiene exactamente una flecha y un elemento más que L. Por HI, sabemos que L tiene la misma cantidad de flechas y de elementos, y por lo tanto $P(L \to k)$ es verdadera.

Por inducción estructural se sigue que todas las listas en $\mathcal{L}_{\mathbb{N}}$ tienen la misma cantidad de flechas que de elementos. $\square$

La def. de $\mathcal{L}_{\mathbb{N}}$ nos guía en las demostraciones de propiedades dentro de $\mathcal{L}_{\mathbb{N}}$

Inducción estructural

Para demostrar propiedades más complejas en $\mathcal{L}_{\mathbb{N}}$, definamos más operadores.

Inducción estructural

Para demostrar propiedades más complejas en $\mathcal{L}_{\mathbb{N}}$, definamos más operadores.

Ejemplo

Definiremos los siguientes operadores para listas

- Largo, recibe lista y entrega número de elementos (números)

$$|\cdot|: \ \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

- Suma, recibe lista y entrega la suma de sus elementos

$$\mathsf{sum}: \ \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

- Máximo, recibe lista y entrega el máximo (o -1 si es vacía)

$$\mathsf{max}: \ \mathcal{L}_{\mathbb{N}} \to \mathbb{N} \cup \{-1\}$$

- Cabeza, recibe lista **no vacía** y entrega su primer elemento

$$\mathsf{head}: \ \mathcal{L}_{\mathbb{N}} \setminus \{\varnothing\} \to \mathbb{N}$$

Inducción estructural

Ejemplo

- Largo, recibe lista y entrega número de elementos (números)

$$|\cdot|:\ \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

Inducción estructural

Ejemplo

- Largo, recibe lista y entrega número de elementos (números)

$$|\cdot| : \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

1. $|\varnothing| = 0$

Inducción estructural

Ejemplo

- Largo, recibe lista y entrega número de elementos (números)

$$|\cdot|: \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

1. $|\varnothing| = 0$
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces $|L \to k| = |L| + 1$

Inducción estructural

Ejemplo

- Largo, recibe lista y entrega número de elementos (números)

$$|\cdot| : \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

1. $|\varnothing| = 0$
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces $|L \to k| = |L| + 1$

- Suma, recibe lista y entrega la suma de sus elementos

$$\mathrm{sum} : \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

Inducción estructural

Ejemplo

- Largo, recibe lista y entrega número de elementos (números)

$$|\cdot|: \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

  1. $|\varnothing| = 0$
  2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces $|L \to k| = |L| + 1$

- Suma, recibe lista y entrega la suma de sus elementos

$$\mathrm{sum}: \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

  1. $\mathrm{sum}(\varnothing) = 0$

Inducción estructural

Ejemplo

- Largo, recibe lista y entrega número de elementos (números)

$$|\cdot|: \ \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

  1. $|\varnothing| = 0$
  2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces $|L \to k| = |L| + 1$

- Suma, recibe lista y entrega la suma de sus elementos

$$\text{sum}: \ \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$$

  1. $\text{sum}(\varnothing) = 0$
  2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces $\text{sum}(L \to k) = \text{sum}(L) + k$

Inducción estructural

Ejemplo

- Máximo, recibe lista y entrega el máximo (o -1 si es vacía)

$$\max : \mathcal{L}_{\mathbb{N}} \to \mathbb{N} \cup \{-1\}$$

Inducción estructural

Ejemplo

- Máximo, recibe lista y entrega el máximo (o -1 si es vacía)

$$\max : \mathcal{L}_{\mathbb{N}} \to \mathbb{N} \cup \{-1\}$$

1. $\max(\varnothing) = -1$

Inducción estructural

Ejemplo

- Máximo, recibe lista y entrega el máximo (o -1 si es vacía)

$$\max : \mathcal{L}_{\mathbb{N}} \to \mathbb{N} \cup \{-1\}$$

1. $\max(\varnothing) = -1$
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces

$$\max(L \to k) = \begin{cases} \max(L) & \text{si } \max(L) \geq k \\ k & \text{en otro caso} \end{cases}$$

Inducción estructural

Ejemplo

- Máximo, recibe lista y entrega el máximo (o -1 si es vacía)

$$\max : \mathcal{L}_{\mathbb{N}} \to \mathbb{N} \cup \{-1\}$$

1. $\max(\varnothing) = -1$
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces

$$\max(L \to k) = \begin{cases} \max(L) & \text{si } \max(L) \geq k \\ k & \text{en otro caso} \end{cases}$$

- Cabeza, recibe lista **no vacía** y entrega su primer elemento

$$\text{head} : \mathcal{L}_{\mathbb{N}} \setminus \{\varnothing\} \to \mathbb{N}$$

Inducción estructural

Ejemplo

- Máximo, recibe lista y entrega el máximo (o -1 si es vacía)

$$\max : \mathcal{L}_{\mathbb{N}} \to \mathbb{N} \cup \{-1\}$$

1. $\max(\varnothing) = -1$
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces

$$\max(L \to k) = \begin{cases} \max(L) & \text{si } \max(L) \geq k \\ k & \text{en otro caso} \end{cases}$$

- Cabeza, recibe lista **no vacía** y entrega su primer elemento

$$\text{head} : \mathcal{L}_{\mathbb{N}} \setminus \{\varnothing\} \to \mathbb{N}$$

1. Si $k \in \mathbb{N}$, entonces $\text{head}(\to k) = k$

Inducción estructural

Ejemplo

- Máximo, recibe lista y entrega el máximo (o -1 si es vacía)

$$\max : \mathcal{L}_{\mathbb{N}} \to \mathbb{N} \cup \{-1\}$$

1. $\max(\varnothing) = -1$
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ y $k \in \mathbb{N}$, entonces

$$\max(L \to k) = \begin{cases} \max(L) & \text{si } \max(L) \geq k \\ k & \text{en otro caso} \end{cases}$$

- Cabeza, recibe lista **no vacía** y entrega su primer elemento

$$\mathrm{head} : \mathcal{L}_{\mathbb{N}} \setminus \{\varnothing\} \to \mathbb{N}$$

1. Si $k \in \mathbb{N}$, entonces $\mathrm{head}(\to k) = k$
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ no vacía y $k \in \mathbb{N}$, entonces $\mathrm{head}(L \to k) = \mathrm{head}(L)$

Inducción estructural

Además, podemos definir operadores que retornan listas.

Inducción estructural

Además, podemos definir operadores que retornan listas.

Ejemplo

El operador sufijo recibe una lista no vacía y entrega la lista resultante de sacarle el primer elemento

$$\text{suf}: \mathcal{L}_{\mathbb{N}} \setminus \{\varnothing\} \to \mathcal{L}_{\mathbb{N}}$$

Inducción estructural

Además, podemos definir operadores que retornan listas.

Ejemplo

El operador sufijo recibe una lista no vacía y entrega la lista resultante de sacarle el primer elemento

$$\mathrm{suf}: \mathcal{L}_{\mathbb{N}} \setminus \{\varnothing\} \to \mathcal{L}_{\mathbb{N}}$$

1. Si $k \in \mathbb{N}$, entonces $\mathrm{suf}(\to k) = \varnothing$

Inducción estructural

Además, podemos definir operadores que retornan listas.

Ejemplo

El operador sufijo recibe una lista no vacía y entrega la lista resultante de sacarle el primer elemento

$$\mathrm{suf}: \mathcal{L}_{\mathbb{N}} \setminus \{\varnothing\} \to \mathcal{L}_{\mathbb{N}}$$

1. Si $k \in \mathbb{N}$, entonces $\mathrm{suf}(\to k) = \varnothing$
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ no vacía y $k \in \mathbb{N}$, entonces $\mathrm{suf}(L \to k) = \mathrm{suf}(L) \to k$

Inducción estructural

Además, podemos definir operadores que retornan listas.

Ejemplo

El operador sufijo recibe una lista no vacía y entrega la lista resultante de sacarle el primer elemento

$$\mathrm{suf}: \mathcal{L}_{\mathbb{N}} \setminus \{\varnothing\} \to \mathcal{L}_{\mathbb{N}}$$

1. Si $k \in \mathbb{N}$, entonces $\mathrm{suf}(\to k) = \varnothing$
2. Si $L \in \mathcal{L}_{\mathbb{N}}$ no vacía y $k \in \mathbb{N}$, entonces $\mathrm{suf}(L \to k) = \mathrm{suf}(L) \to k$

Con estos operadores podemos demostrar propiedades más complejas en $\mathcal{L}_{\mathbb{N}}$

Inducción estructural

Teorema (props. listas)

Si $L, L_1, L_2 \in \mathcal{L}_{\mathbb{N}}$, entonces

1. $\text{sum}(L) \geq 0$
2. $\max(L) \leq \text{sum}(L)$
3. $\text{sum}(L) = \text{head}(L) + \text{sum}(\text{suf}(L))$
4. Si $L_1, L_2 \neq \varnothing$, entonces

$$L_1 = L_2 \quad \text{si y solo si} \quad \text{suf}(L_1) = \text{suf}(L_2) \text{ y } \text{sum}(L_1) = \text{sum}(L_2)$$

Inducción estructural

Teorema (props. listas)

Si $L, L_1, L_2 \in \mathcal{L}_{\mathbb{N}}$, entonces

1. $\text{sum}(L) \geq 0$
2. $\max(L) \leq \text{sum}(L)$
3. $\text{sum}(L) = \text{head}(L) + \text{sum}(\text{suf}(L))$
4. Si $L_1, L_2 \neq \varnothing$, entonces

$$L_1 = L_2 \quad \text{si y solo si} \quad \text{suf}(L_1) = \text{suf}(L_2) \text{ y } \text{sum}(L_1) = \text{sum}(L_2)$$

Demostraremos 4.
El resto queda propuesto (★)

Inducción estructural

Teorema (prop. 4. de listas)

Sean $L_1, L_2 \in \mathcal{L}_{\mathbb{N}}$. Si $L_1, L_2 \neq \varnothing$, entonces

$$L_1 = L_2 \quad \text{si y solo si} \quad \text{suf}(L_1) = \text{suf}(L_2) \text{ y } \text{sum}(L_1) = \text{sum}(L_2)$$

Inducción estructural

Teorema (prop. 4. de listas)

Sean $L_1, L_2 \in \mathcal{L}_{\mathbb{N}}$. Si $L_1, L_2 \neq \varnothing$, entonces

$$L_1 = L_2 \quad \text{si y solo si} \quad \mathrm{suf}(L_1) = \mathrm{suf}(L_2) \text{ y } \mathrm{sum}(L_1) = \mathrm{sum}(L_2)$$

Demostración

La dirección $(\Rightarrow)$ es trivial.

Inducción estructural

Teorema (prop. 4. de listas)

Sean $L_1, L_2 \in \mathcal{L}_{\mathbb{N}}$. Si $L_1, L_2 \neq \varnothing$, entonces

$$L_1 = L_2 \quad \text{si y solo si} \quad \mathrm{suf}(L_1) = \mathrm{suf}(L_2) \text{ y } \mathrm{sum}(L_1) = \mathrm{sum}(L_2)$$

Demostración

La dirección $(\Rightarrow)$ es trivial.

Para la dirección $(\Leftarrow)$, supondremos que L_1, L_2 son listas tales que

$$\mathrm{suf}(L_1) = \mathrm{suf}(L_2) \text{ y } \mathrm{sum}(L_1) = \mathrm{sum}(L_2)$$

Inducción estructural

Teorema (prop. 4. de listas)

Sean $L_1, L_2 \in \mathcal{L}_{\mathbb{N}}$. Si $L_1, L_2 \neq \varnothing$, entonces

$$L_1 = L_2 \quad \text{si y solo si} \quad \mathrm{suf}(L_1) = \mathrm{suf}(L_2) \text{ y } \mathrm{sum}(L_1) = \mathrm{sum}(L_2)$$

Demostración

La dirección $(\Rightarrow)$ es trivial.

Para la dirección $(\Leftarrow)$, supondremos que L_1, L_2 son listas tales que

$$\mathrm{suf}(L_1) = \mathrm{suf}(L_2) \text{ y } \mathrm{sum}(L_1) = \mathrm{sum}(L_2)$$

¿Cuál(es) es(son) **CB**?

Inducción estructural

Demostración

Para la dirección $(\Leftarrow)$, supondremos que L_1, L_2 son listas tales que

$$\mathrm{suf}(L_1) = \mathrm{suf}(L_2) \text{ y } \mathrm{sum}(L_1) = \mathrm{sum}(L_2)$$

- **BI:** Sean $L_1 = \rightarrow k$ y $L_2 = \rightarrow j$ dos listas tales que $\mathrm{suf}(L_1) = \mathrm{suf}(L_2)$ y $\mathrm{sum}(L_1) = \mathrm{sum}(L_2)$. Por definición de sum, tenemos que

  $$k = \mathrm{sum}(\rightarrow k) = \mathrm{sum}(\rightarrow j) = j$$

  y luego $k = j$. Concluimos que $L_1 = L_2$.

Inducción estructural

Demostración

Para la dirección $(\Leftarrow)$, supondremos que L_1, L_2 son listas tales que

$$\mathrm{suf}(L_1) = \mathrm{suf}(L_2) \text{ y } \mathrm{sum}(L_1) = \mathrm{sum}(L_2)$$

- **BI:** Sean $L_1 = \rightarrow k$ y $L_2 = \rightarrow j$ dos listas tales que $\mathrm{suf}(L_1) = \mathrm{suf}(L_2)$ y $\mathrm{sum}(L_1) = \mathrm{sum}(L_2)$. Por definición de sum, tenemos que

  $$k = \mathrm{sum}(\rightarrow k) = \mathrm{sum}(\rightarrow j) = j$$

  y luego $k = j$. Concluimos que $L_1 = L_2$.

- **HI:** Dadas dos listas L_1 y L_2 cualquiera, supongamos que si $\mathrm{suf}(L_1) = \mathrm{suf}(L_2)$ y $\mathrm{sum}(L_1) = \mathrm{sum}(L_2)$, entonces $L_1 = L_2$.

Inducción estructural

Demostración

Para la dirección ($\Leftarrow$), supondremos que L_1, L_2 son listas tales que

$$\mathrm{suf}(L_1) = \mathrm{suf}(L_2) \text{ y } \mathrm{sum}(L_1) = \mathrm{sum}(L_2)$$

- **BI:** Sean $L_1 = \rightarrow k$ y $L_2 = \rightarrow j$ dos listas tales que $\mathrm{suf}(L_1) = \mathrm{suf}(L_2)$ y $\mathrm{sum}(L_1) = \mathrm{sum}(L_2)$. Por definición de sum, tenemos que

  $$k = \mathrm{sum}(\rightarrow k) = \mathrm{sum}(\rightarrow j) = j$$

  y luego $k = j$. Concluimos que $L_1 = L_2$.

- **HI:** Dadas dos listas L_1 y L_2 cualquiera, supongamos que si $\mathrm{suf}(L_1) = \mathrm{suf}(L_2)$ y $\mathrm{sum}(L_1) = \mathrm{sum}(L_2)$, entonces $L_1 = L_2$.

Ojo: el antecedente de la **HI** no necesariamente se cumple.
Cuando se cumple, entonces podemos concluir que $L_1 = L_2$

Inducción estructural

- **HI:** Dadas dos listas L_1 y L_2 cualquiera, supongamos que si $\text{suf}(L_1) = \text{suf}(L_2)$ y $\text{sum}(L_1) = \text{sum}(L_2)$, entonces $L_1 = L_2$.
- **TI:** Sean ahora dos listas $L_1 \to k$ y $L_2 \to j$. Queremos demostrar que si $\text{suf}(L_1 \to k) = \text{suf}(L_2 \to j)$ y $\text{sum}(L_1 \to k) = \text{sum}(L_2 \to j)$, entonces $L_1 \to k = L_2 \to j$.

Inducción estructural

- **HI:** Dadas dos listas L_1 y L_2 cualquiera, supongamos que si $\text{suf}(L_1) = \text{suf}(L_2)$ y $\text{sum}(L_1) = \text{sum}(L_2)$, entonces $L_1 = L_2$.

- **TI:** Sean ahora dos listas $L_1 \to k$ y $L_2 \to j$. Queremos demostrar que si $\text{suf}(L_1 \to k) = \text{suf}(L_2 \to j)$ y $\text{sum}(L_1 \to k) = \text{sum}(L_2 \to j)$, entonces $L_1 \to k = L_2 \to j$.

  Supongamos entonces que $\text{suf}(L_1 \to k) = \text{suf}(L_2 \to j)$ y $\text{sum}(L_1 \to k) = \text{sum}(L_2 \to j)$.

Inducción estructural

- **HI:** Dadas dos listas L_1 y L_2 cualquiera, supongamos que si $\mathrm{suf}(L_1) = \mathrm{suf}(L_2)$ y $\mathrm{sum}(L_1) = \mathrm{sum}(L_2)$, entonces $L_1 = L_2$.

- **TI:** Sean ahora dos listas $L_1 \to k$ y $L_2 \to j$. Queremos demostrar que si $\mathrm{suf}(L_1 \to k) = \mathrm{suf}(L_2 \to j)$ y $\mathrm{sum}(L_1 \to k) = \mathrm{sum}(L_2 \to j)$, entonces $L_1 \to k = L_2 \to j$.

  Supongamos entonces que $\mathrm{suf}(L_1 \to k) = \mathrm{suf}(L_2 \to j)$ y $\mathrm{sum}(L_1 \to k) = \mathrm{sum}(L_2 \to j)$. Por definición de ambas funciones, obtenemos que

$$\mathrm{suf}(L_1) \to k = \mathrm{suf}(L_2) \to j$$
$$\mathrm{sum}(L_1) + k = \mathrm{sum}(L_2) + j$$

Inducción estructural

- **HI:** Dadas dos listas L_1 y L_2 cualquiera, supongamos que si $\text{suf}(L_1) = \text{suf}(L_2)$ y $\text{sum}(L_1) = \text{sum}(L_2)$, entonces $L_1 = L_2$.

- **TI:** Sean ahora dos listas $L_1 \to k$ y $L_2 \to j$. Queremos demostrar que si $\text{suf}(L_1 \to k) = \text{suf}(L_2 \to j)$ y $\text{sum}(L_1 \to k) = \text{sum}(L_2 \to j)$, entonces $L_1 \to k = L_2 \to j$.

  Supongamos entonces que $\text{suf}(L_1 \to k) = \text{suf}(L_2 \to j)$ y $\text{sum}(L_1 \to k) = \text{sum}(L_2 \to j)$. Por definición de ambas funciones, obtenemos que

$$\text{suf}(L_1) \to k = \text{suf}(L_2) \to j$$
$$\text{sum}(L_1) + k = \text{sum}(L_2) + j$$

  Por igualdad de listas, sabemos que necesariamente $\text{suf}(L_1) = \text{suf}(L_2)$ y $k = j$.

Inducción estructural

- **HI:** Dadas dos listas L_1 y L_2 cualquiera, supongamos que si $\text{suf}(L_1) = \text{suf}(L_2)$ y $\text{sum}(L_1) = \text{sum}(L_2)$, entonces $L_1 = L_2$.

- **TI:** Sean ahora dos listas $L_1 \to k$ y $L_2 \to j$. Queremos demostrar que si $\text{suf}(L_1 \to k) = \text{suf}(L_2 \to j)$ y $\text{sum}(L_1 \to k) = \text{sum}(L_2 \to j)$, entonces $L_1 \to k = L_2 \to j$.

  Supongamos entonces que $\text{suf}(L_1 \to k) = \text{suf}(L_2 \to j)$ y $\text{sum}(L_1 \to k) = \text{sum}(L_2 \to j)$. Por definición de ambas funciones, obtenemos que

$$\text{suf}(L_1) \to k = \text{suf}(L_2) \to j$$
$$\text{sum}(L_1) + k = \text{sum}(L_2) + j$$

  Por igualdad de listas, sabemos que necesariamente $\text{suf}(L_1) = \text{suf}(L_2)$ y $k = j$. Usando este último resultado, obtenemos también que $\text{sum}(L_1) = \text{sum}(L_2)$.

Inducción estructural

- **HI:** Dadas dos listas L_1 y L_2 cualquiera, supongamos que si $\mathrm{suf}(L_1) = \mathrm{suf}(L_2)$ y $\mathrm{sum}(L_1) = \mathrm{sum}(L_2)$, entonces $L_1 = L_2$.

- **TI:** Sean ahora dos listas $L_1 \to k$ y $L_2 \to j$. Queremos demostrar que si $\mathrm{suf}(L_1 \to k) = \mathrm{suf}(L_2 \to j)$ y $\mathrm{sum}(L_1 \to k) = \mathrm{sum}(L_2 \to j)$, entonces $L_1 \to k = L_2 \to j$.

  Supongamos entonces que $\mathrm{suf}(L_1 \to k) = \mathrm{suf}(L_2 \to j)$ y $\mathrm{sum}(L_1 \to k) = \mathrm{sum}(L_2 \to j)$. Por definición de ambas funciones, obtenemos que

$$\mathrm{suf}(L_1) \to k = \mathrm{suf}(L_2) \to j$$
$$\mathrm{sum}(L_1) + k = \mathrm{sum}(L_2) + j$$

  Por igualdad de listas, sabemos que necesariamente $\mathrm{suf}(L_1) = \mathrm{suf}(L_2)$ y $k = j$. Usando este último resultado, obtenemos también que $\mathrm{sum}(L_1) = \mathrm{sum}(L_2)$. Luego, por **HI** tenemos que $L_1 = L_2$, y como $k = j$ concluimos que $L_1 \to k = L_2 \to j$. $\square$

Outline

Objetivos de la clase

- Comprender definiciones inductivas
- Definir operadores inductivamente
- Demostrar propiedades mediante inducción estructural