

Tema 2-2

Recursión e inducción



UNIVERSIDAD
DE CHILE

Arturo Merino



Recuerdo/Plan

- **Recuerdo:**

- **Recuerdo:**
 - Demostraciones en lenguaje natural

- **Recuerdo:**

- Demostraciones en lenguaje natural
- Técnicas fundamentales basadas en deducción natural

- **Recuerdo:**

- Demostraciones en lenguaje natural
- Técnicas fundamentales basadas en deducción natural

- **Plan:**

- **Recuerdo:**

- Demostraciones en lenguaje natural
- Técnicas fundamentales basadas en deducción natural

- **Plan:**

- Definiciones recursivas

- **Recuerdo:**

- Demostraciones en lenguaje natural
- Técnicas fundamentales basadas en deducción natural

- **Plan:**

- Definiciones recursivas
- Funciones en objetos recursivos

- **Recuerdo:**

- Demostraciones en lenguaje natural
- Técnicas fundamentales basadas en deducción natural

- **Plan:**

- Definiciones recursivas
- Funciones en objetos recursivos
- Inducción estructural

Definición recursiva

Definiciones recursivas

- **Objetos base + cómo combinarlos.**

Definiciones recursivas

- **Objetos base + cómo combinarlos.**

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

Definiciones recursivas

- **Objetos base + cómo combinarlos.**

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,

Definiciones recursivas

- **Objetos base + cómo combinarlos.**

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

Definiciones recursivas

- **Objetos base + cómo combinarlos.**

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

Diremos que $u \in \mathcal{U}$ se *construye* a partir de B y $\mathcal{F}$ si existe una secuencia (finita) $u_1, \dots, u_m = u$ tal que para todo $i \in \{1, \dots, m\}$

Definiciones recursivas

- **Objetos base + cómo combinarlos.**

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

Diremos que $u \in \mathcal{U}$ se *construye* a partir de B y $\mathcal{F}$ si existe una secuencia (finita) $u_1, \dots, u_m = u$ tal que para todo $i \in \{1, \dots, m\}$

- $u_i \in B$, (caso base)

Definiciones recursivas

- **Objetos base + cómo combinarlos.**

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

Diremos que $u \in \mathcal{U}$ se *construye* a partir de B y $\mathcal{F}$ si existe una secuencia (finita) $u_1, \dots, u_m = u$ tal que para todo $i \in \{1, \dots, m\}$

- $u_i \in B$, (caso base)
- $u_i = f(u_{j_1}, \dots, u_{j_k})$, para alguna regla $f \in \mathcal{F}$ de aridad k e índices $j_1, \dots, j_k < i$. (caso recursivo)

Definiciones recursivas

- **Objetos base + cómo combinarlos.**

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

Diremos que $u \in \mathcal{U}$ se *construye* a partir de B y $\mathcal{F}$ si existe una secuencia (finita) $u_1, \dots, u_m = u$ tal que para todo $i \in \{1, \dots, m\}$

- $u_i \in B$, (caso base)
- $u_i = f(u_{j_1}, \dots, u_{j_k})$, para alguna regla $f \in \mathcal{F}$ de aridad k e índices $j_1, \dots, j_k < i$. (caso recursivo)

A la secuencia $u_1, \dots, u_m$ se le llama *producción* de u .

Definiciones recursivas

- **Para definir un objeto:** base + reglas.

Definiciones recursivas

- **Para definir un objeto:** base + reglas.

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

Definiciones recursivas

- **Para definir un objeto:** base + reglas.

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

- Ej:

Definiciones recursivas

- **Para definir un objeto:** base + reglas.

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

- **Ej:**
 - Números naturales.

Definiciones recursivas

- **Para definir un objeto:** base + reglas.

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

- **Ej:**
 - Números naturales.
 - Potencias de 2.

Definiciones recursivas

- **Para definir un objeto:** base + reglas.

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

- **Ej:**
 - Números naturales.
 - Potencias de 2.
 - Strings binarios – Strings generales.

Definiciones recursivas

- **Para definir un objeto:** base + reglas.

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

- **Ej:**
 - Números naturales.
 - Potencias de 2.
 - Strings binarios – Strings generales.
 - Paréntesis bien balanceados.

Definiciones recursivas

- **Para definir un objeto:** base + reglas.

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

- **Ej:**
 - Números naturales.
 - Potencias de 2.
 - Strings binarios – Strings generales.
 - Paréntesis bien balanceados.
 - Strings binarios palíndromos.

Definiciones recursivas

- **Para definir un objeto:** base + reglas.

Def.

Sea $\mathcal{U}$ un dominio de objetos. Supongamos que tenemos:

- $B \subseteq \mathcal{U}$ *objetos base*,
- $\mathcal{F}$ *reglas de producción* donde $f \in \mathcal{F}$ es una función $f: \mathcal{U}^k \rightarrow \mathcal{U}$.

- **Ej:**
 - Números naturales.
 - Potencias de 2.
 - Strings binarios – Strings generales.
 - Paréntesis bien balanceados.
 - Strings binarios palíndromos.
 - FBF.

Definiciones recursivas

- Usualmente B y $\mathcal{F}$ **implícitos**.

Definiciones recursivas

- Usualmente B y $\mathcal{F}$ **implícitos**.
- Ej:

Definiciones recursivas

- Usualmente B y $\mathcal{F}$ **implícitos**.
- **Ej:**
 - $\mathbb{N} \leftarrow$ Números naturales.

Definiciones recursivas

- Usualmente B y $\mathcal{F}$ **implícitos**.
- **Ej:**
 - $\mathbb{N} \leftarrow$ Números naturales.
 - $\text{Pot} \leftarrow$ Potencias de 2.

Definiciones recursivas

- Usualmente B y $\mathcal{F}$ **implícitos**.
- **Ej:**
 - $\mathbb{N} \leftarrow$ Números naturales.
 - $\text{Pot} \leftarrow$ Potencias de 2.
 - $\{0, 1\}^* \leftarrow$ Strings binarios – $\Sigma^* \leftarrow$ Strings sobre un alfabeto finito Σ .

Definiciones recursivas

- Usualmente B y $\mathcal{F}$ **implícitos**.
- **Ej:**
 - $\mathbb{N} \leftarrow$ Números naturales.
 - $\text{Pot} \leftarrow$ Potencias de 2.
 - $\{0, 1\}^* \leftarrow$ Strings binarios – $\Sigma^* \leftarrow$ Strings sobre un alfabeto finito Σ .
 - $\text{BB} \leftarrow$ Paréntesis bien balanceados.

Definiciones recursivas

- Usualmente B y $\mathcal{F}$ **implícitos**.
- **Ej:**
 - $\mathbb{N} \leftarrow$ Números naturales.
 - $\text{Pot} \leftarrow$ Potencias de 2.
 - $\{0, 1\}^* \leftarrow$ Strings binarios – $\Sigma^* \leftarrow$ Strings sobre un alfabeto finito Σ .
 - $\text{BB} \leftarrow$ Paréntesis bien balanceados.
 - FBF

Definiciones recursivas

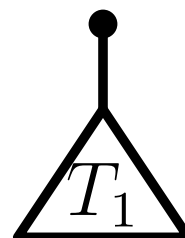
- Usalmente B y $\mathcal{F}$ **implícitos**.
- **Ej:**
 - $\mathbb{N} \leftarrow$ Números naturales.
 - $\text{Pot} \leftarrow$ Potencias de 2.
 - $\{0, 1\}^* \leftarrow$ Strings binarios – $\Sigma^* \leftarrow$ Strings sobre un alfabeto finito Σ .
 - $\text{BB} \leftarrow$ Paréntesis bien balanceados.
 - FBF
- *Fórmulas de Horn:*
 - $\perp$ y $\top$ son fórmulas de Horn.
 - Si p es una variable, p es fórmula de Horn.
 - Si H_1 y H_2 son fórmulas de Horn, $(H_1 \wedge H_2)$ también.
 - Si H es una f. de Horn y p una variable, $(p \rightarrow H)$ es fórmula de Horn.

Definiciones recursivas

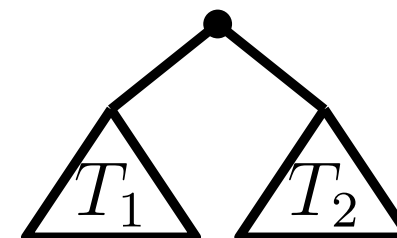
- Usalmente B y $\mathcal{F}$ **implícitos**.
- **Ej:**
 - $\mathbb{N} \leftarrow$ Números naturales.
 - $\text{Pot} \leftarrow$ Potencias de 2.
 - $\{0, 1\}^* \leftarrow$ Strings binarios – $\Sigma^* \leftarrow$ Strings sobre un alfabeto finito Σ .
 - $\text{BB} \leftarrow$ Paréntesis bien balanceados.
 - FBF
- *Fórmulas de Horn:*
 - $\perp$ y $\top$ son fórmulas de Horn.
 - Si p es una variable, p es fórmula de Horn.
 - Si H_1 y H_2 son fórmulas de Horn, $(H_1 \wedge H_2)$ también.
 - Si H es una f. de Horn y p una variable, $(p \rightarrow H)$ es fórmula de Horn.
- *Árboles 1-2:*

$$\bullet \in \mathcal{T}_{1-2}$$

Si $T_1, T_2 \in \mathcal{T}_{1-2}$, entonces



$\in \mathcal{T}_{1-2}$ y



$\in \mathcal{T}_{1-2}$

Funciones en objetos recursivos

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

(varias producciones del mismo objeto deben dar el mismo resultado)

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

(varias producciones del mismo objeto deben dar el mismo resultado)

- Ej:

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

(varias producciones del mismo objeto deben dar el mismo resultado)

- Ej:**
 - $\text{par}(n) = \text{Paridad de } n \in \mathbb{N}.$

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

(varias producciones del mismo objeto deben dar el mismo resultado)

- Ej:**
 - $\text{par}(n) = \text{Paridad de } n \in \mathbb{N}$.
 - $|w| = \text{Largo de un string } w \in \Sigma^*$.

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

(varias producciones del mismo objeto deben dar el mismo resultado)

- Ej:**
 - $\text{par}(n) = \text{Paridad de } n \in \mathbb{N}$.
 - $|w| = \text{Largo de un string } w \in \Sigma^*$.
 - $|w|_1 = \text{Número de 1's en } w \in \{0, 1\}^*$.

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

(varias producciones del mismo objeto deben dar el mismo resultado)

- Ej:**
 - $\text{par}(n) = \text{Paridad de } n \in \mathbb{N}$.
 - $|w| = \text{Largo de un string } w \in \Sigma^*$.
 - $|w|_1 = \text{Número de 1's en } w \in \{0, 1\}^*$.
 - $\text{val}(w) = \text{Valor numérico de un string binario } w \in \{0, 1\}^*$.

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

(varias producciones del mismo objeto deben dar el mismo resultado)

- Ej:**
 - $\text{par}(n) = \text{Paridad de } n \in \mathbb{N}$.
 - $|w| = \text{Largo de un string } w \in \Sigma^*$.
 - $|w|_1 = \text{Número de 1's en } w \in \{0, 1\}^*$.
 - $\text{val}(w) = \text{Valor numérico de un string binario } w \in \{0, 1\}^*$.
 - $\hat{v}(\phi) = \text{Valuación de una fórmula de Horn } \phi$

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

(varias producciones del mismo objeto deben dar el mismo resultado)

- Ej:**
 - $\text{par}(n) = \text{Paridad de } n \in \mathbb{N}$.
 - $|w| = \text{Largo de un string } w \in \Sigma^*$.
 - $|w|_1 = \text{Número de 1's en } w \in \{0, 1\}^*$.
 - $\text{val}(w) = \text{Valor numérico de un string binario } w \in \{0, 1\}^*$.
 - $\hat{v}(\phi) = \text{Valuación de una fórmula de Horn } \phi$
 - $h(T) = \text{altura de un árbol 1-2 } T$.

Definiciones sobre objetos recursivos

- Definir en los **casos bases** y las **reglas de producción**.

Def.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Una función $g: S \rightarrow X$ se *define recursivamente* especificando $g(b)$ para $b \in B$ y para cada regla de producción cómo obtener $g(f(s_1, \dots, s_k))$ a partir de $g(s_1), \dots, g(s_k)$.

(varias producciones del mismo objeto deben dar el mismo resultado)

- Ej:**
 - $\text{par}(n) = \text{Paridad de } n \in \mathbb{N}$.
 - $|w| = \text{Largo de un string } w \in \Sigma^*$.
 - $|w|_1 = \text{Número de 1's en } w \in \{0, 1\}^*$.
 - $\text{val}(w) = \text{Valor numérico de un string binario } w \in \{0, 1\}^*$.
 - $\hat{v}(\phi) = \text{Valuación de una fórmula de Horn } \phi$
 - $h(T) = \text{altura de un árbol 1-2 } T$.
 - $\ell(T) = \text{número de hojas de un árbol 1-2 } T$.

Inducción estructural

Inducción estructural

Prin.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Inducción estructural

Prin.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Sea $P(s)$ una propiedad que depende de un objeto $s \in S$ tal que se cumple lo siguiente:

Inducción estructural

Prin.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Sea $P(s)$ una propiedad que depende de un objeto $s \in S$ tal que se cumple lo siguiente:

- $P(b)$ es verdadera para todo $b \in B$ (caso base)

Inducción estructural

Prin.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Sea $P(s)$ una propiedad que depende de un objeto $s \in S$ tal que se cumple lo siguiente:

- $P(b)$ es verdadera para todo $b \in B$ (caso base)
- Para toda $f \in \mathcal{F}$ de aridad k y objetos $s_1, \dots, s_k \in S$, se cumple $P(s_1) \wedge \dots \wedge P(s_k) \rightarrow P(f(s_1, \dots, s_k))$ (paso inductivo)

Inducción estructural

Prin.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Sea $P(s)$ una propiedad que depende de un objeto $s \in S$ tal que se cumple lo siguiente:

- $P(b)$ es verdadera para todo $b \in B$ (caso base)
- Para toda $f \in \mathcal{F}$ de aridad k y objetos $s_1, \dots, s_k \in S$, se cumple
 $P(s_1) \wedge \dots \wedge P(s_k) \rightarrow P(f(s_1, \dots, s_k))$ (paso inductivo)

Entonces, $P(s)$ es verdadera para todo $s \in S$.

Inducción estructural

Prin.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Sea $P(s)$ una propiedad que depende de un objeto $s \in S$ tal que se cumple lo siguiente:

- $P(b)$ es verdadera para todo $b \in B$ (caso base)
- Para toda $f \in \mathcal{F}$ de aridad k y objetos $s_1, \dots, s_k \in S$, se cumple $P(s_1) \wedge \dots \wedge P(s_k) \rightarrow P(f(s_1, \dots, s_k))$ (paso inductivo)

Entonces, $P(s)$ es verdadera para todo $s \in S$.

- P es verdadera en los casos bases.

Inducción estructural

Prin.

Sea S un conjunto de objetos contruidos a partir de objetos básicos B y reglas de producción $\mathcal{F}$.

Sea $P(s)$ una propiedad que depende de un objeto $s \in S$ tal que se cumple lo siguiente:

- $P(b)$ es verdadera para todo $b \in B$ (caso base)
- Para toda $f \in \mathcal{F}$ de aridad k y objetos $s_1, \dots, s_k \in S$, se cumple $P(s_1) \wedge \dots \wedge P(s_k) \rightarrow P(f(s_1, \dots, s_k))$ (paso inductivo)

Entonces, $P(s)$ es verdadera para todo $s \in S$.

- P es verdadera en los casos bases.
- P se transmite a través de las reglas de producción.

■ Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

■ Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

Casos bases:

Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

Casos bases:

- Caso base 1 satisface $P(b_1)$ ✓

Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

Casos bases:

- Caso base 1 satisface $P(b_1)$ ✓
- ⋮
- Caso base t satisface $P(b_t)$ ✓

Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

Casos bases:

- Caso base 1 satisface $P(b_1)$ ✓
- ⋮
- Caso base t satisface $P(b_t)$ ✓

Paso inductivo:

Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

Casos bases:

- Caso base 1 satisface $P(b_1)$ ✓
- ⋮
- Caso base t satisface $P(b_t)$ ✓

Paso inductivo:

- Regla de producción $f_1(s_1, \dots, s_{k_1}) = s$

Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

Casos bases:

- Caso base 1 satisface $P(b_1)$ ✓
- $\vdots$
- Caso base t satisface $P(b_t)$ ✓

Paso inductivo:

- Regla de producción $f_1(s_1, \dots, s_{k_1}) = s$
Hip: $P(s_1) \wedge P(s_2) \wedge \dots \wedge P(s_k)$

Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

Casos bases:

- Caso base 1 satisface $P(b_1)$ ✓
- ⋮
- Caso base t satisface $P(b_t)$ ✓

Paso inductivo:

- Regla de producción $f_1(s_1, \dots, s_{k_1}) = s$
Hip: $P(s_1) \wedge P(s_2) \wedge \dots \wedge P(s_{k_1})$
PDQ: $P(s)$

Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

Casos bases:

- Caso base 1 satisface $P(b_1)$ ✓
- ⋮
- Caso base t satisface $P(b_t)$ ✓

Paso inductivo:

- Regla de producción $f_1(s_1, \dots, s_{k_1}) = s$
 - Hip:** $P(s_1) \wedge P(s_2) \wedge \dots \wedge P(s_k)$
 - PDQ:** $P(s)$
 - Dem:** $\dots$ ✓

Demostraciones por inducción estructural

- ¿Cómo se ve una demostración de inducción estructural?

Casos bases:

- Caso base 1 satisface $P(b_1)$ ✓
- ⋮
- Caso base t satisface $P(b_t)$ ✓

Paso inductivo:

- Regla de producción $f_1(s_1, \dots, s_{k_1}) = s$
Hip: $P(s_1) \wedge P(s_2) \wedge \dots P(s_k)$
PDQ: $P(s)$
Dem: ... ✓
- ⋮
- Regla de producción $f_\ell(r_1, \dots, r_{k_\ell}) = r$
Hip: $P(r_1) \wedge P(r_2) \wedge \dots P(r_k)$
PDQ: $P(r)$
Dem: ... ✓

Ejemplos

- Demuestre que para todo bitstring $w \in \{0, 1\}^*$ tenemos que $|w|_0 + |w|_1 = |w|$.

Ejemplos

- Demuestre que para todo bitstring $w \in \{0, 1\}^*$ tenemos que $|w|_0 + |w|_1 = |w|$.
- Demuestre que para todo bitstring $w \in \{0, 1\}^*$ tenemos que $0 \leq \text{val}(w) < 2^{|w|}$

Ejemplos

- Demuestre que para todo bitstring $w \in \{0, 1\}^*$ tenemos que $|w|_0 + |w|_1 = |w|$.
- Demuestre que para todo bitstring $w \in \{0, 1\}^*$ tenemos que $0 \leq \text{val}(w) < 2^{|w|}$
- Demuestre que para todo par de bitstrings $x, y \in \{0, 1\}^*$ tenemos que $\text{val}(xy) = 2^{|y|}\text{val}(x) + \text{val}(y)$

Ejemplos

- Demuestre que para todo bitstring $w \in \{0, 1\}^*$ tenemos que $|w|_0 + |w|_1 = |w|$.
- Demuestre que para todo bitstring $w \in \{0, 1\}^*$ tenemos que $0 \leq \text{val}(w) < 2^{|w|}$
- Demuestre que para todo par de bitstrings $x, y \in \{0, 1\}^*$ tenemos que $\text{val}(xy) = 2^{|y|}\text{val}(x) + \text{val}(y)$
- Demuestre que para todo 1-2 árbol T se tiene que $\ell(T) \leq 2^{h(T)}$.

Ejemplos

- Demuestre que para todo bitstring $w \in \{0, 1\}^*$ tenemos que $|w|_0 + |w|_1 = |w|$.
- Demuestre que para todo bitstring $w \in \{0, 1\}^*$ tenemos que $0 \leq \text{val}(w) < 2^{|w|}$
- Demuestre que para todo par de bitstrings $x, y \in \{0, 1\}^*$ tenemos que $\text{val}(xy) = 2^{|y|}\text{val}(x) + \text{val}(y)$
- Demuestre que para todo 1-2 árbol T se tiene que $\ell(T) \leq 2^{h(T)}$.
- Sea v_X una valuación que es verdadera en un conjunto de variables X . Sean A, B conjuntos. Muestre que si ϕ es una fórmula de Horn verdadera con v_A y v_B , entonces es verdadera con $v_{A \cap B}$.

Dudas/Preguntas

Wrap-Up

- **Hoy:**

Wrap-Up

- **Hoy:**
 - Objetos recursivos

Wrap-Up

- **Hoy:**
 - Objetos recursivos
 - Funciones sobre objetos recursivos

Wrap-Up

- **Hoy:**
 - Objetos recursivos
 - Funciones sobre objetos recursivos
 - Inducción estructural

Wrap-Up

- **Hoy:**
 - Objetos recursivos
 - Funciones sobre objetos recursivos
 - Inducción estructural
- **Próx. clase:**

Wrap-Up

- **Hoy:**

- Objetos recursivos
- Funciones sobre objetos recursivos
- Inducción estructural

- **Próx. clase:**

- Inducción sobre los naturales