

Tema 2-1

Técnicas de demostración



UNIVERSIDAD
DE CHILE

Arturo Merino



Recuerdo/Plan

- **Recuerdo:**

- **Recuerdo:**
 - Deducción natural en lógica de predicados

Recuerdo/Plan

- **Recuerdo:**

- Deducción natural en lógica de predicados

- **Plan:**

- **Recuerdo:**

- Deducción natural en lógica de predicados

- **Plan:**

- Demostraciones en lenguaje natural

- **Recuerdo:**

- Deducción natural en lógica de predicados

- **Plan:**

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

Demostraciones en lenguaje natural

Demostraciones en lenguaje natural

- ¿Difs. deducción natural?

Lemma 5.1. *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda: \mathcal{I} \rightarrow [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \dots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ .

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability

Demostraciones en lenguaje natural

- ¿Difs. deducción natural?

Lemma 5.1. *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda: \mathcal{I} \rightarrow [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \dots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ .

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability

- Prosa en lenguaje natural.

Demostraciones en lenguaje natural

- ¿Difs. deducción natural?

Lemma 5.1. *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda: \mathcal{I} \rightarrow [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \dots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ .

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability

- Prosa en lenguaje natural.
- Símbolos para sostener argumentos

Demostraciones en lenguaje natural

- ¿Difs. deducción natural?

Lemma 5.1. *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda: \mathcal{I} \rightarrow [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \dots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ .

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability

- **Demostraciones en lenguaje natural:**

- Prosa en lenguaje natural.
- Símbolos para sostener argumentos

Demostraciones en lenguaje natural

- ¿Difs. deducción natural?

Lemma 5.1. *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda: \mathcal{I} \rightarrow [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \dots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ .

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability

- **Demostraciones en lenguaje natural:**

- **Obj:** Comunicar a otro humano, para que pueda **entender** y **validar**.

- Prosa en lenguaje natural.
- Símbolos para sostener argumentos

Demostraciones en lenguaje natural

- ¿Difs. deducción natural?

Lemma 5.1. *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda: \mathcal{I} \rightarrow [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \dots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ .

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability

- **Demostraciones en lenguaje natural:**

- **Obj:** Comunicar a otro humano, para que pueda **entender** y **validar**.
- En prosa en español,

- Prosa en lenguaje natural.
- Símbolos para sostener argumentos

Demostraciones en lenguaje natural

- ¿Difs. deducción natural?

Lemma 5.1. *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda: \mathcal{I} \rightarrow [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \dots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ .

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability

- **Demostraciones en lenguaje natural:**

- **Obj:** Comunicar a otro humano, para que pueda **entender y validar**.
- En prosa en español,
- Usando símbolos cuando sea necesario,

- Prosa en lenguaje natural.
- Símbolos para sostener argumentos

Demostraciones en lenguaje natural

- ¿Difs. deducción natural?

Lemma 5.1. *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda: \mathcal{I} \rightarrow [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \dots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ .

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability

- **Demostraciones en lenguaje natural:**

- **Obj:** Comunicar a otro humano, para que pueda **entender** y **validar**.
- En prosa en español,
- Usando símbolos cuando sea necesario,
- Válidas, formales y entendibles.

- Prosa en lenguaje natural.
- Símbolos para sostener argumentos

Demostraciones en lenguaje natural

- ¿Difs. deducción natural?

Lemma 5.1. *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda: \mathcal{I} \rightarrow [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \dots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ .

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability

- **Demostraciones en lenguaje natural:**

- **Obj:** Comunicar a otro humano, para que pueda **entender** y **validar**.
- En prosa en español,
- Usando símbolos cuando sea necesario,
- Válidas, formales y entendibles.

- How to Prove it? Daniel J. Velleman.

- Prosa en lenguaje natural.
- Símbolos para sostener argumentos

Técnicas fundamentales

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ”:

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ”:

- Suponer P

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ” :

- Suponer P
- Usar definiciones, propiedades, teoremas

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ” :

- Suponer P
- Usar definiciones, propiedades, teoremas
- Concluir Q .

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ” :

- Suponer P
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\rightarrow$ -introducción.

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ” :

- Suponer P
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\rightarrow$ -introducción.
- **Ej:** Si $n \in \mathbb{N}$ es par, entonces n^2 es par.

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ”:

- Suponer P
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\rightarrow$ -introducción.
- **Ej:** Si $n \in \mathbb{N}$ es par, entonces n^2 es par.

Tec.

Contrarrecíproca: Para demostrar “Si P , entonces Q ”:

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ”:

- Suponer P
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\rightarrow$ -introducción.
- **Ej:** Si $n \in \mathbb{N}$ es par, entonces n^2 es par.

Tec.

Contrarrecíproca: Para demostrar “Si P , entonces Q ”:

- Suponer $\neg Q$

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ”:

- Suponer P
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\rightarrow$ -introducción.
- **Ej:** Si $n \in \mathbb{N}$ es par, entonces n^2 es par.

Tec.

Contrarrecíproca: Para demostrar “Si P , entonces Q ”:

- Suponer $\neg Q$
- Usar definiciones, propiedades, teoremas

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ”:

- Suponer P
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\rightarrow$ -introducción.
- **Ej:** Si $n \in \mathbb{N}$ es par, entonces n^2 es par.

Tec.

Contrarrecíproca: Para demostrar “Si P , entonces Q ”:

- Suponer $\neg Q$
- Usar definiciones, propiedades, teoremas
- Concluir $\neg P$.

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ”:

- Suponer P
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\rightarrow$ -introducción.
- **Ej:** Si $n \in \mathbb{N}$ es par, entonces n^2 es par.

Tec.

Contrarrecíproca: Para demostrar “Si P , entonces Q ”:

- Suponer $\neg Q$
- Usar definiciones, propiedades, teoremas
- Concluir $\neg P$.

- **Basado en:** $P \rightarrow Q \equiv \neg Q \rightarrow \neg P$

Implicancia

Tec.

Demostración directa: Para demostrar “Si P , entonces Q ”:

- Suponer P
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\rightarrow$ -introducción.
- **Ej:** Si $n \in \mathbb{N}$ es par, entonces n^2 es par.

Tec.

Contrarrecíproca: Para demostrar “Si P , entonces Q ”:

- Suponer $\neg Q$
- Usar definiciones, propiedades, teoremas
- Concluir $\neg P$.

- **Basado en:** $P \rightarrow Q \equiv \neg Q \rightarrow \neg P$
- **Ej:** Si $n \in \mathbb{N}$ es tal que n^2 par, entonces n es par.

Disjunción

Tec.

Por casos: Para demostrar “Si $P_1 \vee P_2 \vee \dots \vee P_k$, entonces Q ”:

Disjunción

Tec.

Por casos: Para demostrar “Si $P_1 \vee P_2 \vee \dots \vee P_k$, entonces Q ”:

- Suponer P_1

Disjunción

Tec.

Por casos: Para demostrar “Si $P_1 \vee P_2 \vee \dots \vee P_k$, entonces Q ”:

- Suponer P_1
- Usar definiciones, propiedades, teoremas

Disjunción

Tec.

Por casos: Para demostrar “Si $P_1 \vee P_2 \vee \dots \vee P_k$, entonces Q ”:

- Suponer P_1
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- ⋮

Disjunción

Tec.

Por casos: Para demostrar “Si $P_1 \vee P_2 \vee \dots \vee P_k$, entonces Q ”:

- Suponer P_1
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- $\vdots$
- Suponer P_k

Disjunción

Tec.

Por casos: Para demostrar “Si $P_1 \vee P_2 \vee \dots \vee P_k$, entonces Q ”:

- Suponer P_1
- Usar definiciones, propiedades, teoremas
- Concluir Q .
-
- Suponer P_k
- Usar definiciones, propiedades, teoremas

Disjunción

Tec.

Por casos: Para demostrar “Si $P_1 \vee P_2 \vee \dots \vee P_k$, entonces Q ”:

- Suponer P_1
- Usar definiciones, propiedades, teoremas
- Concluir Q .
-
- Suponer P_k
- Usar definiciones, propiedades, teoremas
- Concluir Q .

Disjunción

Tec.

Por casos: Para demostrar “Si $P_1 \vee P_2 \vee \dots \vee P_k$, entonces Q ”:

- Suponer P_1
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- ⋮
- Suponer P_k
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\vee$ -eliminación.

Disjunción

Tec.

Por casos: Para demostrar “Si $P_1 \vee P_2 \vee \dots \vee P_k$, entonces Q ”:

- Suponer P_1
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- ⋮
- Suponer P_k
- Usar definiciones, propiedades, teoremas
- Concluir Q .

- **Basado en:** $\vee$ -eliminación.
- **Ej:** Si $n \in \mathbb{N}$ no es múltiplo de 3, entonces 3 divide a $n^2 - 1$.

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.

Negación

Tec.

Rescritura: Para demostrar “ $\neg P$ ”:

- Reescribir $\neg P$ “en positivo” como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas

Negación

Tec.

Rescritura: Para demostrar “ $\neg P$ ”:

- Reescribir $\neg P$ “en positivo” como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- Concluir $\neg P$.

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- Concluir $\neg P$.

- **Ej:** A, B, C tres cjtos. con $A \cap C \subseteq B$. Si $a \in C$, muestre $a \notin A \setminus B$.

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- Concluir $\neg P$.

- **Ej:** A, B, C tres cjtos. con $A \cap C \subseteq B$. Si $a \in C$, muestre $a \notin A \setminus B$.

Tec.

Contradicción: Para demostrar " $\neg P$:"

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- Concluir $\neg P$.

- **Ej:** A, B, C tres cjtos. con $A \cap C \subseteq B$. Si $a \in C$, muestre $a \notin A \setminus B$.

Tec.

Contradicción: Para demostrar " $\neg P$:"

- Asumir P .

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- Concluir $\neg P$.

- **Ej:** A, B, C tres cjtos. con $A \cap C \subseteq B$. Si $a \in C$, muestre $a \notin A \setminus B$.

Tec.

Contradicción: Para demostrar " $\neg P$:"

- Asumir P .
- Usar definiciones, propiedades, teoremas

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- Concluir $\neg P$.

- **Ej:** A, B, C tres cjtos. con $A \cap C \subseteq B$. Si $a \in C$, muestre $a \notin A \setminus B$.

Tec.

Contradicción: Para demostrar " $\neg P$:"

- Asumir P .
- Usar definiciones, propiedades, teoremas
- Concluir una contradicción.

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- Concluir $\neg P$.

- **Ej:** A, B, C tres cjtos. con $A \cap C \subseteq B$. Si $a \in C$, muestre $a \notin A \setminus B$.

Tec.

Contradicción: Para demostrar " $\neg P$:"

- Asumir P .
- Usar definiciones, propiedades, teoremas
- Concluir una contradicción.
- Concluir $\neg P$.

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- Concluir $\neg P$.

- **Ej:** A, B, C tres cjtos. con $A \cap C \subseteq B$. Si $a \in C$, muestre $a \notin A \setminus B$.

Tec.

Contradicción: Para demostrar " $\neg P$:"

- Asumir P .
- Usar definiciones, propiedades, teoremas
- Concluir una contradicción.
- Concluir $\neg P$.

- **Basado en:** reducción al absurdo.

Negación

Tec.

Rescritura: Para demostrar " $\neg P$:"

- Reescribir $\neg P$ "en positivo" como $Q \equiv \neg P$.
- Usar definiciones, propiedades, teoremas
- Concluir Q .
- Concluir $\neg P$.

- **Ej:** A, B, C tres cjtos. con $A \cap C \subseteq B$. Si $a \in C$, muestre $a \notin A \setminus B$.

Tec.

Contradicción: Para demostrar " $\neg P$:"

- Asumir P .
- Usar definiciones, propiedades, teoremas
- Concluir una contradicción.
- Concluir $\neg P$.

- **Basado en:** reducción al absurdo.
- **Ej:** $\sqrt{2}$ es irracional.

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,
- Concluir $P \leftrightarrow Q$.

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,
- Concluir $P \leftrightarrow Q$.

- **Basado en:** $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$.

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,
- Concluir $P \leftrightarrow Q$.

- **Basado en:** $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$.
- **Ej:** Sean $x, y \in \mathbb{Z}$. $x + y$ es par si y sólo si x e y tienen la misma paridad.

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,
- Concluir $P \leftrightarrow Q$.

- **Basado en:** $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$.
- **Ej:** Sean $x, y \in \mathbb{Z}$. $x + y$ es par si y sólo si x e y tienen la misma paridad.

Tec.

Cadena: para demostrar " $P \leftrightarrow Q$ ":

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,
- Concluir $P \leftrightarrow Q$.

- **Basado en:** $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$.
- **Ej:** Sean $x, y \in \mathbb{Z}$. $x + y$ es par si y sólo si x e y tienen la misma paridad.

Tec.

Cadena: para demostrar " $P \leftrightarrow Q$ ":

- Encontrar $P_1, \dots, P_k$

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,
- Concluir $P \leftrightarrow Q$.

- **Basado en:** $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$.
- **Ej:** Sean $x, y \in \mathbb{Z}$. $x + y$ es par si y sólo si x e y tienen la misma paridad.

Tec.

Cadena: para demostrar " $P \leftrightarrow Q$ ":

- Encontrar $P_1, \dots, P_k$
- Demostrar $P \leftrightarrow P_1, P_1 \leftrightarrow P_2, \dots, P_k \leftrightarrow Q$.

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,
- Concluir $P \leftrightarrow Q$.

- **Basado en:** $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$.
- **Ej:** Sean $x, y \in \mathbb{Z}$. $x + y$ es par si y sólo si x e y tienen la misma paridad.

Tec.

Cadena: para demostrar " $P \leftrightarrow Q$ ":

- Encontrar $P_1, \dots, P_k$
- Demostrar $P \leftrightarrow P_1, P_1 \leftrightarrow P_2, \dots, P_k \leftrightarrow Q$.
- Concluir $P \leftrightarrow Q$.

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,
- Concluir $P \leftrightarrow Q$.

- **Basado en:** $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$.
- **Ej:** Sean $x, y \in \mathbb{Z}$. $x + y$ es par si y sólo si x e y tienen la misma paridad.

Tec.

Cadena: para demostrar " $P \leftrightarrow Q$ ":

- Encontrar $P_1, \dots, P_k$
- Demostrar $P \leftrightarrow P_1, P_1 \leftrightarrow P_2, \dots, P_k \leftrightarrow Q$.
- Concluir $P \leftrightarrow Q$.

- **Basado en:** transitividad de $\leftrightarrow$.

Equivalencia

Tec.

Separación: para demostrar " $P \leftrightarrow Q$ ":

- Demostrar $P \rightarrow Q$,
- Demostrar $Q \rightarrow P$,
- Concluir $P \leftrightarrow Q$.

- **Basado en:** $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$.
- **Ej:** Sean $x, y \in \mathbb{Z}$. $x + y$ es par si y sólo si x e y tienen la misma paridad.

Tec.

Cadena: para demostrar " $P \leftrightarrow Q$ ":

- Encontrar $P_1, \dots, P_k$
- Demostrar $P \leftrightarrow P_1, P_1 \leftrightarrow P_2, \dots, P_k \leftrightarrow Q$.
- Concluir $P \leftrightarrow Q$.

- **Basado en:** transitividad de $\leftrightarrow$.
- **Ej:** Sean A, B, C conjuntos. Entonces $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$.

Cuantificadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

Cuantificadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$

Cuantificadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$

Cuantificadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$
- Concluir $\forall x \in A P(x)$.

Cuantificadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$
- Concluir $\forall x \in A P(x)$.

- **Basado en:** $\forall$ -intro.

Cuantificadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$
- Concluir $\forall x \in A P(x)$.

- **Basado en:** $\forall$ -intro.
- **Ej:** Para todo $x, y \in \mathbb{Q}$, tenemos que $x + y \in \mathbb{Q}$.

Cuanticadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$
- Concluir $\forall x \in A P(x)$.

- **Basado en:** $\forall$ -intro.
- **Ej:** Para todo $x, y \in \mathbb{Q}$, tenemos que $x + y \in \mathbb{Q}$.

Tec.

Existencial: para demostrar " $\exists x \in A P(x)$ ":

Cuanticadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$
- Concluir $\forall x \in A P(x)$.

- **Basado en:** $\forall$ -intro.
- **Ej:** Para todo $x, y \in \mathbb{Q}$, tenemos que $x + y \in \mathbb{Q}$.

Tec.

Existencial: para demostrar " $\exists x \in A P(x)$ ":

- Encontrar $a \in A$

Cuanticadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$
- Concluir $\forall x \in A P(x)$.

- **Basado en:** $\forall$ -intro.
- **Ej:** Para todo $x, y \in \mathbb{Q}$, tenemos que $x + y \in \mathbb{Q}$.

Tec.

Existencial: para demostrar " $\exists x \in A P(x)$ ":

- Encontrar $a \in A$
- Probar $P(a)$

Cuanticadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$
- Concluir $\forall x \in A P(x)$.

- **Basado en:** $\forall$ -intro.
- **Ej:** Para todo $x, y \in \mathbb{Q}$, tenemos que $x + y \in \mathbb{Q}$.

Tec.

Existencial: para demostrar " $\exists x \in A P(x)$ ":

- Encontrar $a \in A$
- Probar $P(a)$
- Concluir $\exists x \in A P(x)$.

Cuanticadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$
- Concluir $\forall x \in A P(x)$.

- **Basado en:** $\forall$ -intro.
- **Ej:** Para todo $x, y \in \mathbb{Q}$, tenemos que $x + y \in \mathbb{Q}$.

Tec.

Existencial: para demostrar " $\exists x \in A P(x)$ ":

- Encontrar $a \in A$
- Probar $P(a)$
- Concluir $\exists x \in A P(x)$.

- **Basado en:** $\forall$ -intro.

Cuanticadores

Tec.

Universal: para demostrar " $\forall x \in A P(x)$ ":

- Declarar un elemento arbitrario $a \in A$
- Probar $P(a)$
- Concluir $\forall x \in A P(x)$.

- **Basado en:** $\forall$ -intro.
- **Ej:** Para todo $x, y \in \mathbb{Q}$, tenemos que $x + y \in \mathbb{Q}$.

Tec.

Existencial: para demostrar " $\exists x \in A P(x)$ ":

- Encontrar $a \in A$
- Probar $P(a)$
- Concluir $\exists x \in A P(x)$.

- **Basado en:** $\exists$ -intro.
- **Ej:** Existe $m \in \mathbb{Z}$ tal que $\frac{m-7}{2m+4} = 5$.

Dudas/Preguntas

Wrap-Up

- **Hoy:**

Wrap-Up

- **Hoy:**
 - Demostraciones (en lenguaje natural)

Wrap-Up

- **Hoy:**
 - Demostraciones (en lenguaje natural)
 - Repaso técnicas de demostración

Wrap-Up

- **Hoy:**

- Demostraciones (en lenguaje natural)
- Repaso técnicas de demostración

- **Próx. clase:**

Wrap-Up

- **Hoy:**

- Demostraciones (en lenguaje natural)
- Repaso técnicas de demostración

- **Próx. clase:**

- Objetos recursivos

Wrap-Up

- **Hoy:**

- Demostraciones (en lenguaje natural)
- Repaso técnicas de demostración

- **Próx. clase:**

- Objetos recursivos
- Inducción estructural