# Ingeniería de ontologías

# Lógica descriptiva

# Razonamiento

Setiembre 2023

FACULTAD DE INGENIERIA

# Agenda

- **Lógica descriptiva**
  - **Mecanismos de razonamiento**
  - **Algoritmo de tableau**
  - **Mundo abierto**

# Lógica Descriptiva

Partimos de un conjunto de **nombres de conceptos atómicos *A*, *B*,..** y **nombres de roles atómicos *R*, *S*,.., nombres de individuos *a*, *b*, …**

Cada lógica permite construir conceptos y axiomas con diferente expresividad:

$\mathcal{ALC}$**:** $\top, \bot, \sqcap, \sqcup, \exists, \forall, \neg$

$\mathcal{S}$**:** $\mathcal{ALC}$ + **roles transitivos** Trans($R$)

A $\mathcal{S}$ se agregan constructores que se representan por diferentes letras:

$\mathcal{H}$**:** inclusión de roles $\mathcal{O}$**:** nominales $\{a\}$ $\mathcal{I}$**:** roles inversos

$\mathcal{N}$**:** restricciones numéricas $\mathcal{Q}$**:** restrictions numéricas calificadas

$\mathcal{R}$: Dis($R$, $S$) roles disjuntos Irr($R$) roles irreflexivos

Aserciones de negación de roles: $(John, Mary) : \neg likes$,

Axiomas de inclusión de roles complejos: $R \circ S \sqsubseteq Q$, universal role $U$, $\exists R.Self$

Descripción de conceptos en lógica $\mathcal{ALCQ}$:

$C, D := \bot | \top | A | \neg C | C \sqcap D | C \sqcup D | \forall R.C | \exists R.C | >= nR.C$

Concepto vacío Todo el Universo

# Lógica Descriptiva

$\mathcal{ALC}$: ⊤, ⊥, ⊓, ⊔, ∃, ∀, ¬

$\mathcal{S}$: $\mathcal{ALC}$ + **roles transitivos** Trans($R$)

A $\mathcal{S}$ se agregan constructores que se representan por diferentes letras:

$\mathcal{H}$: inclusión de roles $\mathcal{O}$: nominales $\{a\}$ $\mathcal{I}$: roles inversos

$\mathcal{N}$: restricciones numéricas $\mathcal{Q}$: restrictions numéricas calificadas

$\mathcal{R}$: Dis($R$, $S$) roles disjuntos Irr($R$) roles irreflexivos

Aserciones de negación de roles: $(John, Mary) : \neg likes,$

Axiomas de inclusión de roles complejos: $R \circ S \sqsubseteq Q$, universal role $U$, $\exists R.Self$

$\mathcal{H}$: $R \sqsubseteq S$ *tieneHijo ⊑ tieneDescendiente*

$\mathcal{O}$: $\{a\}$ {Uruguay} $\mathcal{I}$: $S = R^-$ *Hijo = Padre⁻*

$\mathcal{N}$: >=$nR$.⊤ >=2*tieneHijo*.⊤ $\mathcal{Q}$: >=$nR.C$ >=2*tieneHijo.Mujer*

$\mathcal{R}$: **Dis($R$, $S$)** Dis(*esAmigoDe, esEnemigoDe*) **Irr($R$)** Irr(*tieneHijo*)

$R \circ S \sqsubseteq Q$ *esHermano ∘ esPadre ⊑ esTio*

Roles transitivos: **Trans($R$)** → $R \circ R \sqsubseteq R$

# Lógica Descriptiva

**Base de conocimiento** $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$

**TBox** $\mathcal{T}$: Conocimiento básico, descripción intensional sobre el dominio de un problema. $C \sqsubseteq D$

**ABox** $\mathcal{A}$: Conocimiento sobre individuos, descripción extensional, sobre el dominio de un problema. $C(a)$ $R(a, b)$ $a = b$ $a \neq b$

**RBox** $\mathcal{R}$: Conocimiento básico, descripción intensional sobre el conjunto de pares de elementos del dominio. $R \sqsubseteq S$ $\mathrm{Dis}(R, S)$ $R \circ S \sqsubseteq Q$

$Mujer \sqsubseteq Persona$

$tieneHijo(maria, diego)$

$mariajose = majo$

$esPadre \circ esPadre \sqsubseteq esAbuelo$

# Lógica Descriptiva

## Interpretación

Sea $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ tal que:

$\Delta^{\mathcal{I}}$: dominio de interpretación, conjunto no vacío

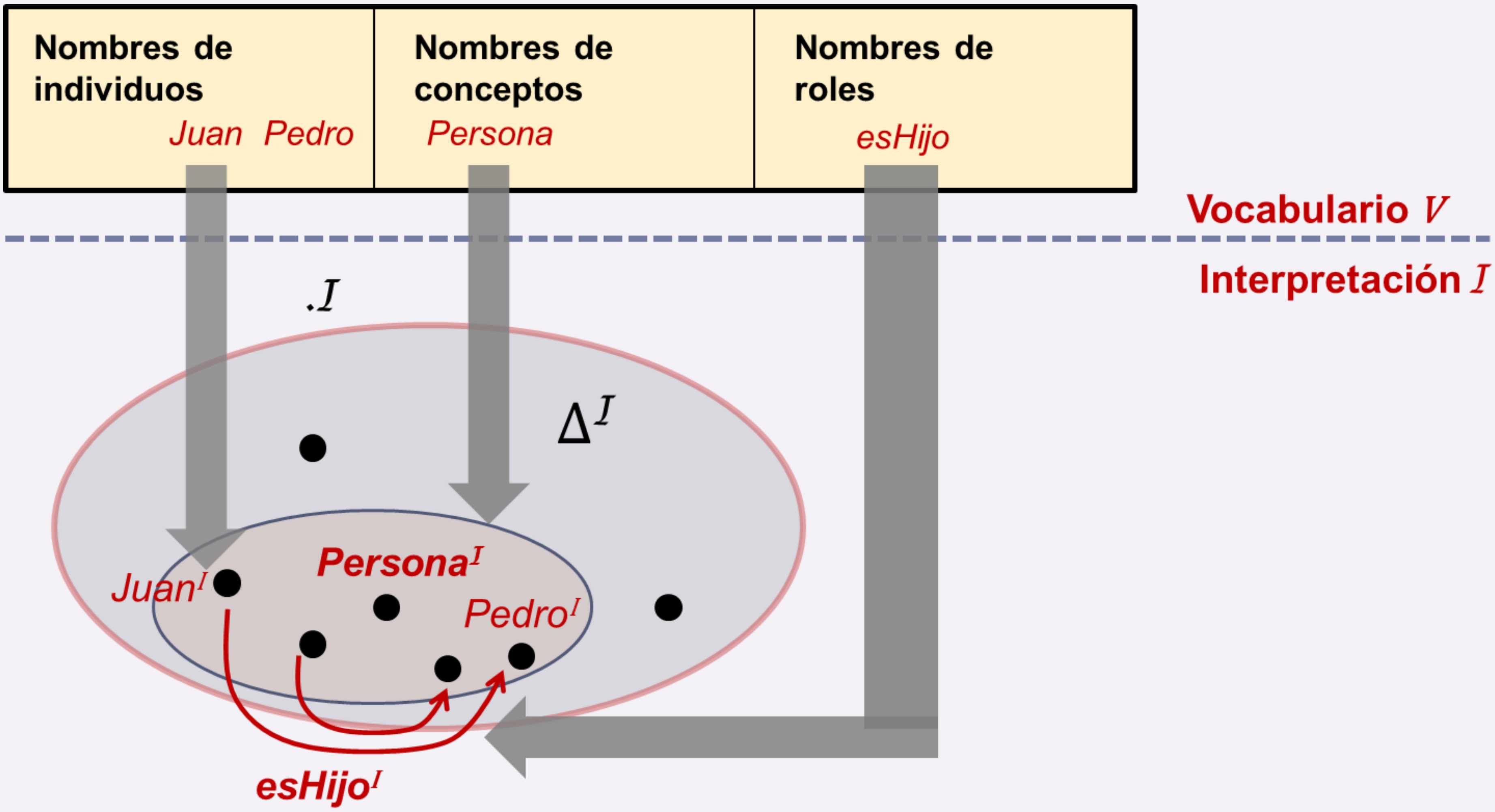
$\cdot^{\mathcal{I}}$: función de interpretación que asigna

- A cada concepto $A$ un conjunto $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
- A cada rol $R$ una relación binaria $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
- A cada individuo $a$ un elemento $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$

| Constructor | DL | Semántica |
|---|---|---|
| bottom | $\bot$ | $\emptyset$ |
| top | $\top$ | $\Delta^{\mathcal{I}}$ |
| negación | $\neg C$ | $\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$ |
| conjunción | $C \sqcap D$ | $C^{\mathcal{I}} \cap D^{\mathcal{I}}$ |
| disjunción | $C \sqcup D$ | $C^{\mathcal{I}} \cup D^{\mathcal{I}}$ |
| restricción existencial | $\exists R.C$ | $\{x \mid \exists y.(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$ |
| restricción universal | $\forall R.C$ | $\{x \mid \forall y.(x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$ |
| restricción de cardinalidad | $>=nR.C$ | $\{x \mid \#\{y.(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} >= n\}$ |

# Lógica Descriptiva

## Interpretación



# Lógica Descriptiva

## Modelo de una base de conocimiento

Sea $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ una interpretación y $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ una base de conocimiento,

$\mathcal{I}$ es un **modelo** de $\mathcal{K}$ si:

- $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ para todo $C \sqsubseteq D$ en $\mathcal{T}$
- $a^{\mathcal{I}} \in C^{\mathcal{I}}$ para todo $C(a)$ en $\mathcal{A}$
- $<a^{\mathcal{I}}, b^{\mathcal{I}}> \in R^{\mathcal{I}}$ para todo $R(a, b)$ en $\mathcal{A}$
- $a^{\mathcal{I}} = b^{\mathcal{I}}$ para todo $a = b$ en $\mathcal{A}$
- $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ para todo $a \neq b$ en $\mathcal{A}$

**$\mathcal{K}$ es consistente si existe $\mathcal{I}$ modelo de $\mathcal{K}$**

# Lógica Descriptiva –Mecanismos de razonamiento

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ Base de conocimiento

$\mathcal{K}$ es **consistente** si existe un modelo $\mathcal{I}$ de $\mathcal{K}$

$C$ es **satisfactible** con respecto a $\mathcal{K}$ si existe un modelo $\mathcal{I}$ de $\mathcal{K}$ tal que $C^{\mathcal{I}} \neq \emptyset$

$C$ está **incluido** en $D$ con respecto a $\mathcal{K}$
si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ se satisface para todo modelo $\mathcal{I}$ de $\mathcal{K} \longrightarrow \mathcal{K} \models C \sqsubseteq D$

$a$ es una **instancia** de $C$ con respecto a $\mathcal{K}$
si $a^{\mathcal{I}} \in C^{\mathcal{I}}$ para todo modelo $\mathcal{I}$ de $\mathcal{K} \longrightarrow \mathcal{K} \models C(a)$

# Mecanismos de razonamiento - Ejemplos

## Mecanismos de razonamiento

Ejemplo:

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

TBox $\mathcal{T}$:

$Mujer \sqsubseteq Persona$

$Persona \equiv Mujer \sqcup Hombre$

$Madre \equiv Mujer \sqcap \exists tieneHijo.Persona$

Abox $\mathcal{A}$:

$Mujer(maria), Mujer(juana), Hombre(diego)$
$tieneHijo(maria, diego)$

$\mathcal{K} \models Hombre \sqsubseteq Persona$ ?

$\mathcal{K} \models Madre(maria)$ ?

# Mecanismos de razonamiento - Ejemplos

## Mecanismos de razonamiento

Ejemplo:

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

TBox $\mathcal{T}$:

$Mujer \sqsubseteq Persona$

$Persona \equiv Mujer \sqcup Hombre$

$Madre \equiv Mujer \sqcap \exists tieneHijo.Persona$

Abox $\mathcal{A}$:

$Mujer(maria)$, $Mujer(juana)$, $Hombre(diego)$

$tieneHijo(maria, diego)$

$\mathcal{K} \models Hombre \sqsubseteq Persona$ ? → *Hombre* está **incluido** en *Persona*

$\mathcal{K} \models Madre(maria)$ ? → *maria* es una **instancia** de *Madre*

# Mecanismos de razonamiento - Ejemplos

## Mecanismos de razonamiento

Ejemplo:

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

TBox $\mathcal{T}$:

$Mujer \sqsubseteq Persona \quad Persona \equiv Mujer \sqcup Hombre \quad Madre \equiv Mujer \sqcap \exists tieneHijo.Persona$

$Mujer \sqcap Hombre \sqsubseteq \bot \qquad C \sqsubseteq Mujer \sqcap Hombre.$

Abox $\mathcal{A}$:

$Mujer(maria), Mujer(juana), Hombre(diego)$

$tieneHijo(maria, diego)$

*C es satisfactible?*

# Mecanismos de razonamiento - Ejemplos

## Mecanismos de razonamiento

Ejemplo:

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

TBox $\mathcal{T}$:

$Mujer \sqsubseteq Persona \quad Persona \equiv Mujer \sqcup Hombre \quad Madre \equiv Mujer \sqcap \exists tieneHijo.Persona$

$Mujer \sqcap Hombre \sqsubseteq \bot \qquad C \sqsubseteq Mujer \sqcap Hombre.$

Abox $\mathcal{A}$:

$Mujer(maria)$, $Mujer(juana)$, $Hombre(diego)$
$tieneHijo(maria, diego)$

*C es satisfactible?*

*C* no es satisfactible, ya que no existe ninguna interpretación tal que $C^{\mathrm{I}} \neq \emptyset$.

# Lógica Descriptiva –Mecanismos de razonamiento

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ Base de conocimiento

$\mathcal{K}$ es **consistente** si existe un modelo $\mathcal{I}$ de $\mathcal{K}$

$C$ es **satisfactible** con respecto a $\mathcal{K}$ si existe un modelo $\mathcal{I}$ de $\mathcal{K}$ tal que $C^{\mathcal{I}} \neq \emptyset$

$C$ está **incluido** en $D$ con respecto a $\mathcal{K}$

si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ se satisface para todo modelo $\mathcal{I}$ de $\mathcal{K}$ $\longrightarrow$ $\mathcal{K} \models C \sqsubseteq D$

$a$ es una **instancia** de $C$ con respecto a $\mathcal{K}$

si $a^{\mathcal{I}} \in C^{\mathcal{I}}$ para todo modelo $\mathcal{I}$ de $\mathcal{K}$ $\longrightarrow$ $\mathcal{K} \models C(a)$

$\Downarrow$

Algoritmo de razonamiento

# Lógica Descriptiva –Mecanismos de razonamiento

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ Base de conocimiento

$\mathcal{K}$ es **consistente** si existe un modelo $\mathcal{I}$ de $\mathcal{K}$

$C$ es **satisfactible** con respecto a $\mathcal{K}$ si existe un modelo $\mathcal{I}$ de $\mathcal{K}$ tal que $C^{\mathcal{I}} \neq \emptyset$

$C$ está **incluido** en $D$ con respecto a $\mathcal{K}$
si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ se satisface para todo modelo $\mathcal{I}$ de $\mathcal{K} \longrightarrow \mathcal{K} \models C \sqsubseteq D$

$a$ es una **instancia** de $C$ con respecto a $\mathcal{K}$
si $a^{\mathcal{I}} \in C^{\mathcal{I}}$ para todo modelo $\mathcal{I}$ de $\mathcal{K} \longrightarrow \mathcal{K} \models C(a)$

⇩

Algoritmo de razonamiento → **TABLEAU**

# Algoritmo de Tableau

Dada un base de conocimiento $\mathcal{K}$, intenta **construir un modelo de $\mathcal{K}$**

Si lo construye → $\mathcal{K}$ es consistente

Si no logra construirlo → $\mathcal{K}$ es inconsistente

Pasos:

- **Inicialización**:

  A partir de los individuos de $\mathcal{K}$ y de los axiomas de Abox, **construye un grafo (tableau) inicial**

- **Aplicación de reglas:**

  **Extiende el grafo** con la **aplicación de un conjunto de reglas**, que dependen de la lógica de $\mathcal{K}$

- **Terminación:**

  Si obtiene un grafo sobre el que **no se pueden aplicar más reglas**, y no existe **ninguna contradicción** → $\mathcal{K}$ es **consistente**

# Algoritmo de Tableau - Grafo

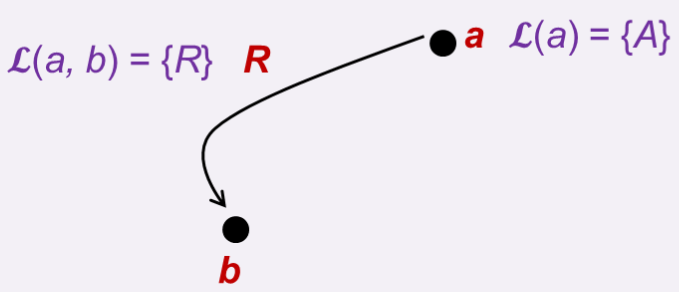
Un **Tableau** para una base de conocimiento $\mathcal{ALC}$ es un **grafo** que se compone de:

- Un conjunto de **nodos etiquetados**, cada etiqueta es un nombre de *individuo* o de *variable*

- **Arcos dirigidos entre nodos**

- Dado un **nodo** con etiqueta $x$, $\mathcal{L}(x)$ es un **conjunto de conceptos**

- Por cada **par de nodos** $x$, $y$, $\mathcal{L}(x, y)$ es un **conjunto de roles**



# Tableau $\mathcal{ALC}$ – Inicialización

Se transforman todos los axiomas de $\mathcal{K}$ a forma normal de negación: **FNN**

**FNN**:

- La negación ocurre delante de los conceptos atómicos
- Ejemplo:

  $Abuela \sqcap \neg Madre$ **está en FNN**

  $\neg(\neg Abuela \sqcup Madre)$ no está en FNN

- $A \sqsubseteq B$ no está en FNN

  $\neg A \sqcup B$ **está en FNN → Concepto de Tbox**

Todos los elementos del dominio satisfacen los conceptos de Tbox

# Tableau $\mathcal{ALC}$ – Inicialización

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ en FNN

- Por cada individuo ***a*** en $\mathcal{K}$ se crea un nodo con etiqueta $a$, $\mathcal{L}(a) = \emptyset$
- Por cada par ***a*, *b*** de individuos, se crea un arco entre $a$ y $b$, $\mathcal{L}(a, b) = \emptyset$
- Por cada axioma de ABox ***C*(*a*)** en $\mathcal{K}$, $\mathcal{L}(a) \leftarrow C$
- Por cada axioma de ABox ***R*(*a*, *b*)** en $\mathcal{K}$, $\mathcal{L}(a, b) \leftarrow R$

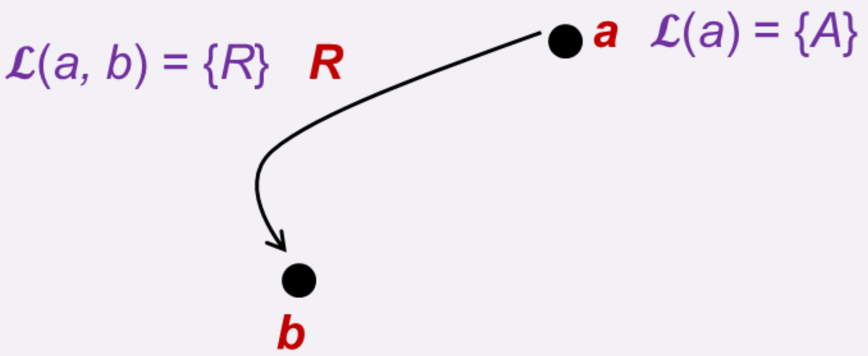
Ejemplo: $\mathcal{K} = \{\ C(a),\ \neg C \sqcap D(a)\ \}$

Un nodo: $a$

$\mathcal{L}(a) = \{\ C,\ \neg C \sqcap D\ \}$

# Tableau $\mathcal{ALC}$ – Inicialización

$A \sqsubseteq B$
$A(a)$, $R(a, b)$



# Tableau $\mathcal{ALC}$ – Inicialización

Otro ejemplo:

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Humano \sqsubseteq \exists tienePadre.Humano\} = \{\neg Humano \sqcup \exists tienePadre.Humano\}$

$\mathcal{A} = \{Ave(picaflor), Humano(picaflor)\}$

$\mathcal{L}(picaflor) = \{Ave, Humano\}$

*picaflor*

# Tableau $\mathcal{ALC}$ – Aplicación de reglas

Se aplican **Reglas de Expansión**:

- **Regla ⊓**: Si $C \sqcap D \in \mathcal{L}(x)$ y $\{C, D\} \not\subseteq \mathcal{L}(x)$ entonces $\mathcal{L}(x) \leftarrow \{C, D\}$
- **Regla ⊔**: Si $C \sqcup D \in \mathcal{L}(x)$ y $\{C, D\} \cap \mathcal{L}(x) = \emptyset$ entonces $\mathcal{L}(x) \leftarrow \{C\}$ ó

  $\mathcal{L}(x) \leftarrow \{D\}$
- **Regla ∃**: Si $\exists R.C \in \mathcal{L}(x)$ y no existe *y tal que* $R \in \mathcal{L}(x, y)$ y $C \in \mathcal{L}(y)$ entonces:
  - Se agrega un nodo con etiqueta *y*
  - $\mathcal{L}(x, y) = \{R\}$
  - $\mathcal{L}(y) = \{C\}$
- **Regla ∀**: Si $\forall R.C \in \mathcal{L}(x)$ y existe *y tal que* $R \in \mathcal{L}(x, y)$ y $C \notin \mathcal{L}(y)$ entonces

  $\mathcal{L}(y) \leftarrow \{C\}$
- **Regla $\mathcal{T}$**: Si $C \in \mathcal{T}$ y $C \notin \mathcal{L}(x)$ entonces $\mathcal{L}(x) \leftarrow \{C\}$

# Tableau $\mathcal{ALC}$ – Inicialización y aplicación de reglas

Ejemplo:

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Humano \sqsubseteq \exists tienePadre.Humano\} = \{\neg Humano \sqcup \exists tienePadre.Humano\}$

$\mathcal{A} = \{ Ave(picaflor), Humano(picaflor)\}$

$\mathcal{L}(picaflor) = \{ Ave, Humano, \neg Humano \sqcup \exists tienePadre.Humano \}$

*picaflor*

# Tableau $\mathcal{ALC}$ – Inicialización y aplicación de reglas

Otro ejemplo:

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Humano \sqsubseteq \exists tienePadre.Humano\} = \{\neg Humano \sqcup \exists tienePadre.Humano\}$

$\mathcal{A} = \{ Ave(picaflor), Humano(picaflor)\}$

$\mathcal{L}(picaflor) = \{ Ave, Humano, \cancel{\neg Humano} \sqcup \exists tienePadre.Humano \}$

$\mathcal{L}(x) = \{Humano, \neg Humano \sqcup \exists tienePadre.Humano \}$

$$
\begin{tikzcd}
picaflor \arrow[d, "tienePadre"] \\
x
\end{tikzcd}
$$

# Tableau $\mathcal{ALC}$ – Inicialización y aplicación de reglas

Otro ejemplo:

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Humano \sqsubseteq \exists tienePadre.Humano\} = \{\neg Humano \sqcup \exists tienePadre.Humano\}$

$\mathcal{A} = \{ Ave(picaflor), Humano(picaflor)\}$

$\mathcal{L}(picaflor) = \{ Ave, Humano, \neg \cancel{Humano} \sqcup \exists tienePadre.Humano \}$

$\mathcal{L}(x) = \{Humano, \neg \cancel{Humano} \sqcup \exists tienePadre.Humano \}$

*picaflor*

$x$

$\mathcal{L}(picaflor, x) = \{tienePadre\}$

# Tableau $\mathcal{ALC}$ – Terminación

Cuándo **paramos** de aplicar reglas?

- El grafo tiene una contradicción si existen $a$ y $C$ tal que

  $\{C, \neg C\} \subseteq \mathcal{L}(a)$

Cuando no se puede aplicar más reglas:

- Si por **todos los caminos** se llega a una **contradicción**: **base inconsistente**
- Si por algún camino no hay contradicción, **encontré un modelo**:

$$\begin{tikzcd}
 & \mathcal{L}_0 \arrow[d] & \\
 & \mathcal{L}_1 \arrow[dl, "\sqcup"] \arrow[dr, "\sqcup"] & \\
\mathcal{L}_2 & & \mathcal{L}_3 \arrow[d] \\
 & & \mathcal{L}_4
\end{tikzcd}$$

**base consistente**

# Tableau – Estrategia de aplicación de las reglas

Estrategia de aplicación de las reglas: **NO DETERMINISTA**

- Qué regla aplicar en cada paso

- Regla $\sqcup$: Si $C \sqcup D \in \mathcal{L}(x)$ y $\{C, D\} \cap \mathcal{L}(x) = \emptyset$ entonces $\mathcal{L}(x) \leftarrow \{C\}$ ó $\mathcal{L}(x) \leftarrow \{D\}$

# Mundo abierto

Información en la **Web** puede estar **incompleta**

**≠**

Información en una **base de datos** se asume **completa**

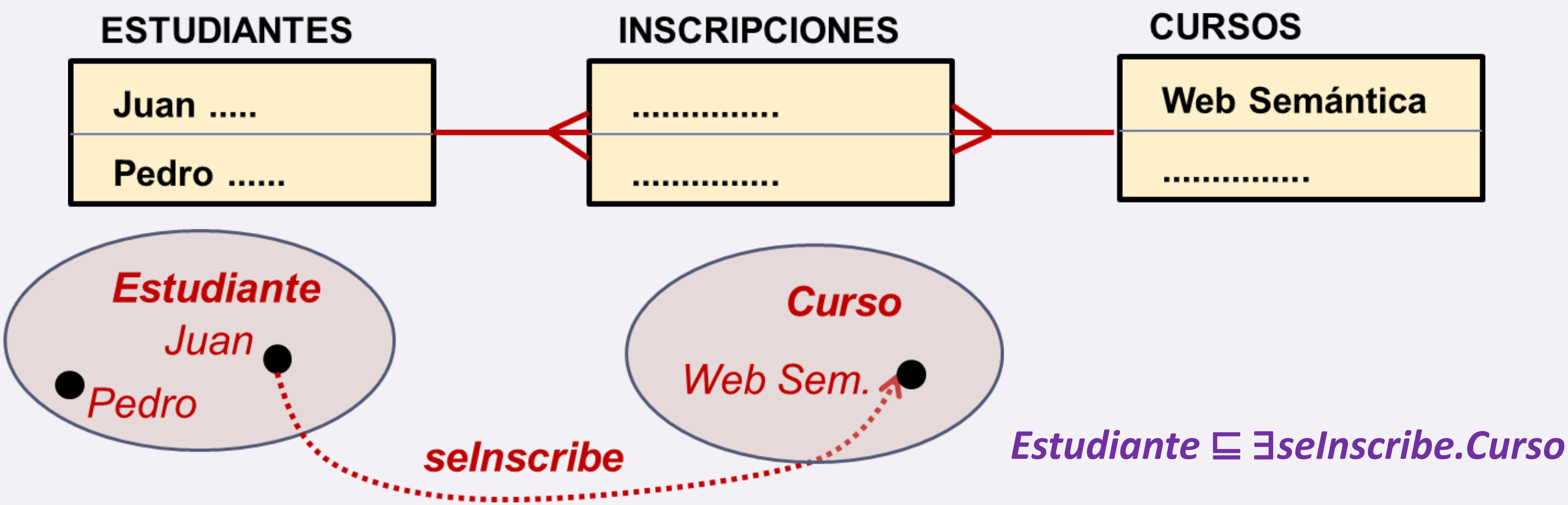
- No se puede derivar que una afirmación es falsa, porque no pueda demostrarse que es verdadera.
- **No se puede derivar falsedad de la ausencia de verdad**

# Mundo abierto

Información en la **Web** puede estar **incompleta**

≠

Información en una **base de datos** se asume **complete**

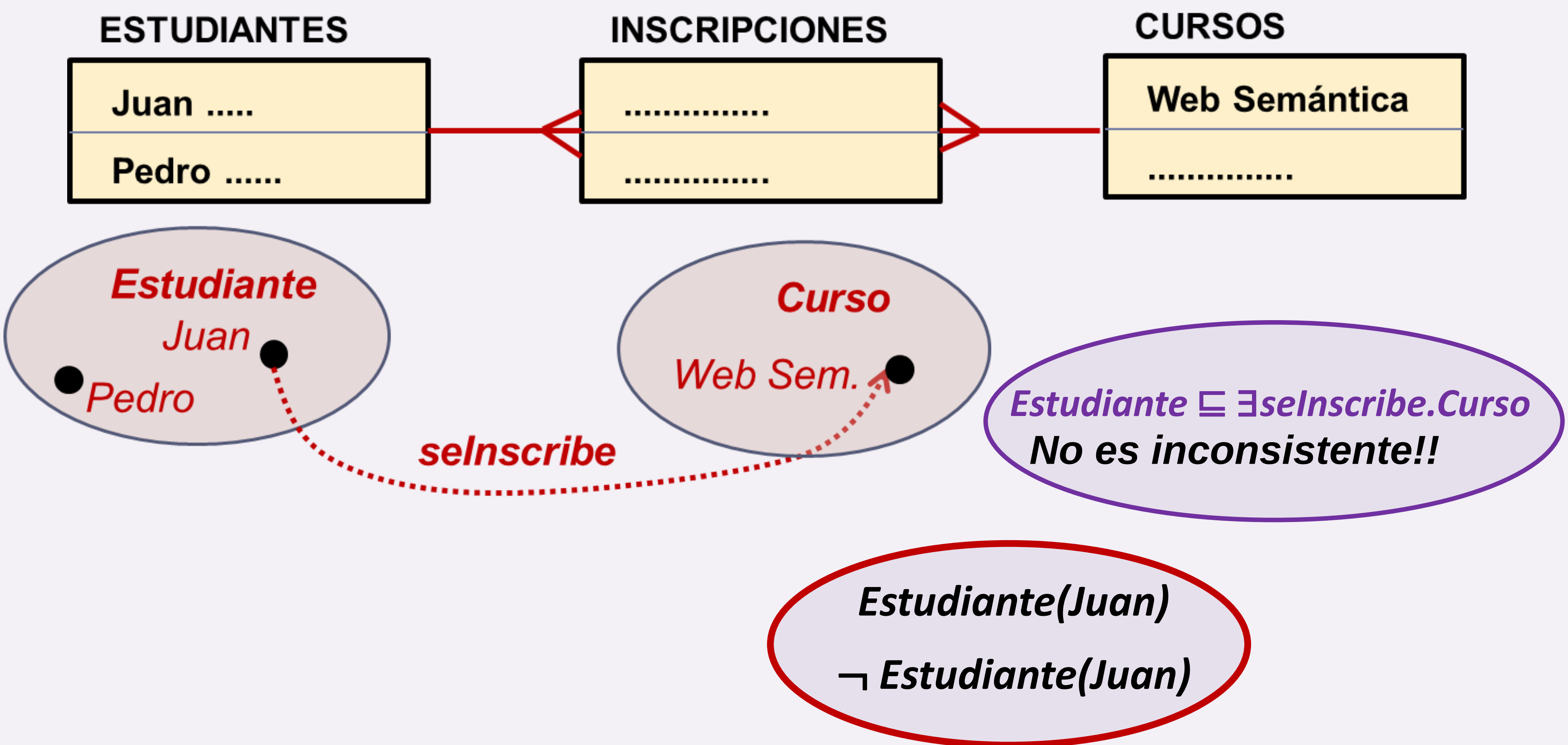


# Mundo abierto

Información en la **Web** puede estar **incompleta**

$\neq$

Información en una **base de datos** se asume **complete**

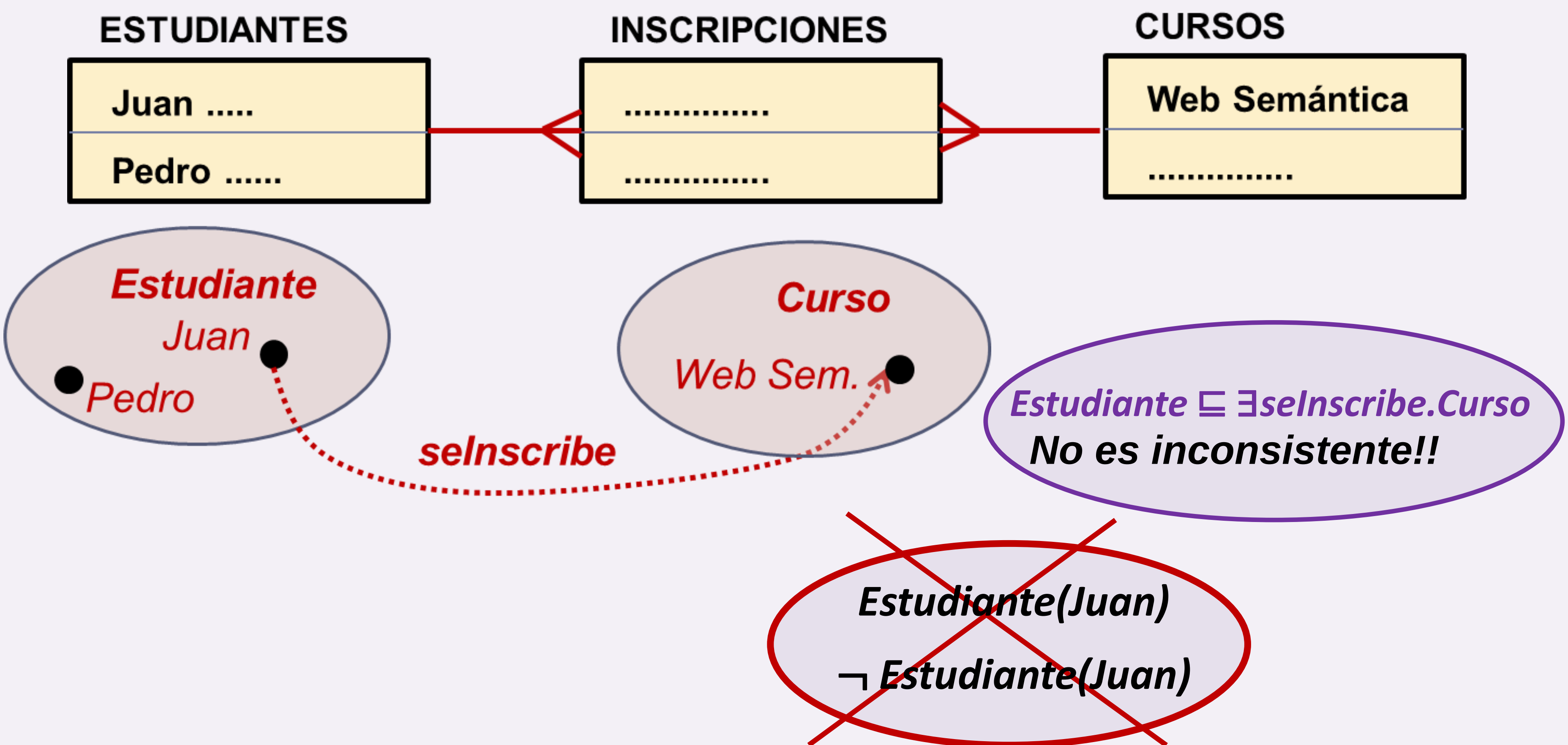


# Mundo abierto

Información en la **Web** puede estar **incompleta**

$\neq$

Información en una **base de datos** se asume **complete**



# Mundo abierto

Información en la **Web** puede estar **incompleta**

$\neq$

Información en una **base de datos** se asume **completa**

**DL** asume **SUN**: *Supuesto de Unicidad de Nombres*

Pero **OWL no asume SUN**:

- En el mundo de la Web no es realista este supuesto, ya que el mismo recurso puede existir con dos URIS diferentes
- Se debe declarar explícitamente que **dos individuos son diferentes**.

*Juan* ● = ● *Pedro*

*Juan* ● $\neq$ ● *Pedro*

# Mundo abierto

Información en la **Web** puede estar **incompleta**

≠

Información en una **base de datos** se asume **completa**

- No se puede derivar que una afirmación es falsa, porque no pueda demostrarse que es verdadera.
- No se puede derivar falsedad de la ausencia de verdad

**El razonador sigue este paradigma!!**

# Tableau – Mundo abierto

## Clases disjuntas

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Hombre \sqsubseteq Humano,\ \ Pájaro \sqsubseteq Ave\}$

$\mathcal{A} = \{\boldsymbol{Pájaro(miPicaflor),\ \ Hombre(Juan)}\}$

***miPicaflor*** **es *Humano*?**

***Juan*** **es *Ave*?**

# Tableau – Mundo abierto

## Clases disjuntas

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Hombre \sqsubseteq Humano,\ \ Pájaro \sqsubseteq Ave\} = \boldsymbol{\{\neg Hombre \sqcup Humano, \neg Pájaro \sqcup Ave\}}$

$\mathcal{A} = \boldsymbol{\{Pájaro(miPicaflor),\ \ Hombre(Juan)\}}$

# Tableau – Mundo abierto

## Clases disjuntas

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Hombre \sqsubseteq Humano,\ \ Pájaro \sqsubseteq Ave\} = \boldsymbol{\{\neg Hombre \sqcup Humano, \neg Pájaro \sqcup Ave\}}$

$\mathcal{A} = \boldsymbol{\{Pájaro(miPicaflor),\ \ Hombre(Juan)\}}$

$\mathcal{L}(miPicaflor) = \{\ Pájaro\ \}$

$\mathcal{L}(Juan) = \{\ Hombre\ \}$

# Tableau – Mundo abierto

## Clases disjuntas

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Hombre \sqsubseteq Humano,\ \ Pájaro \sqsubseteq Ave\} = \boldsymbol{\{\neg Hombre \sqcup Humano, \neg Pájaro \sqcup Ave\}}$

$\mathcal{A} = \boldsymbol{\{Pájaro(miPicaflor),\ \ Hombre(Juan)\}}$

$\mathcal{L}(miPicaflor) = \{\ Pájaro, \neg Hombre \sqcup Humano, \neg Pájaro \sqcup Ave\}$

$\mathcal{L}(Juan) = \{\ Hombre, \neg Hombre \sqcup Humano, \neg Pájaro \sqcup Ave\}$

# Tableau – Mundo abierto

## Clases disjuntas

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Hombre \sqsubseteq Humano,\ \ Pájaro \sqsubseteq Ave\} = \{\neg \boldsymbol{Hombre} \sqcup \boldsymbol{Humano}, \neg \boldsymbol{Pájaro} \sqcup \boldsymbol{Ave}\}$

$\mathcal{A} = \{\boldsymbol{Pájaro(miPicaflor)},\ \ \boldsymbol{Hombre(Juan)}\}$

$\mathcal{L}(miPicaflor) = \{Pájaro, \neg Hombre \sqcup \boldsymbol{Humano}, \cancel{\neg Pájaro} \sqcup Ave\}$

$\mathcal{L}(Juan) = \{\ Hombre, \cancel{\neg Hombre} \sqcup Humano, \neg Pájaro \sqcup \boldsymbol{Ave}\}$

# Tableau – Mundo abierto

## Clases disjuntas

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Hombre \sqsubseteq Humano,\ \ Pájaro \sqsubseteq Ave\} = \{\neg \mathbf{Hombre} \sqcup \mathbf{Humano},\ \neg \mathbf{Pájaro} \sqcup \mathbf{Ave}\}$

$\mathcal{A} = \{\mathbf{Pájaro(miPicaflor)},\ \ \mathbf{Hombre(Juan)}\}$

$\mathcal{L}(miPicaflor) = \{Pájaro,\ \neg Hombre \sqcup \mathbf{Humano},\ \neg \cancel{Pájaro} \sqcup Ave\}$

$\mathcal{L}(Juan) = \{\ Hombre,\ \neg \cancel{Hombre} \sqcup Humano,\ \neg Pájaro \sqcup \mathbf{Ave}\}$

No hay inconsistencia, pero existe **un modelo** donde *Ave* y *Humano* tienen instancias comunes

**Explícitamente se debe expresar que *Ave* y *Humano* son clases disjuntas**

⇩

$\{\mathbf{Ave} \sqcap \mathbf{Humano} \sqsubseteq \bot\} = \{\neg(Ave \sqcap Humano) \sqcup \bot\} = \{\neg \mathbf{Ave} \sqcup \neg \mathbf{Humano} \sqcup \bot\}$

# Tableau – Mundo abierto

## Clases disjuntas

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$\mathcal{T} = \{Hombre \sqsubseteq Humano,\ \ Pájaro \sqsubseteq Ave\} = \{\neg \textbf{Hombre} \sqcup \textbf{Humano}, \neg \textbf{Pájaro} \sqcup \textbf{Ave}\}$

$\mathcal{A} = \{\textbf{Pájaro}(\textbf{miPicaflor}),\ \ \textbf{Hombre}(\textbf{Juan})\}$

$\mathcal{L}(miPicaflor) = \{Pájaro, \neg Hombre \sqcup \textbf{Humano}, \cancel{\neg Pájaro} \sqcup Ave\}$

$\mathcal{L}(Juan) = \{ Hombre, \cancel{\neg Hombre} \sqcup Humano, \neg Pájaro \sqcup \textbf{Ave}\}$

$\{\textbf{Ave} \sqcap \textbf{Humano} \sqsubseteq \bot\} = \{\neg(Ave \sqcap Humano) \sqcup \bot\} = \{\neg \textbf{Ave} \sqcup \neg \textbf{Humano} \sqcup \bot\}$

# Tableau – Mundo abierto

## Restricciones: ∀, ∃

$Padre \sqsubseteq \forall tieneHijo.Humano$ → ∀ se satisface trivialmente: ∀ **no implica** ∃

# Tableau – Mundo abierto

## Restricciones: ∀, ∃

$Padre \sqsubseteq \forall tieneHijo.Humano$ → ∀ se satisface trivialmente: ∀ **no implica** ∃

Para expresar que todos los padres **deben tener al menos** un hijo:

$Padre \sqsubseteq \exists tieneHijo.Humano$

# Tableau – Mundo abierto

## Restricciones: ∀, ∃

$Padre \sqsubseteq \exists tieneHijo.Humano$

Mundo abierto: los padres pueden tener otros hijos que no sean humanos…

# Tableau – Mundo abierto

## Restricciones: ∀, ∃

*Padre* $\sqsubseteq$ ∃*tieneHijo.Humano*

Mundo abierto: los padres pueden tener otros hijos que no sean humanos…

Para **“cerrar”** este mundo:

***Padre*** $\sqsubseteq$ **∃*tieneHijo.Humano*** $\sqcap$ **∀*tieneHijo.Humano***

Axioma de Clausura

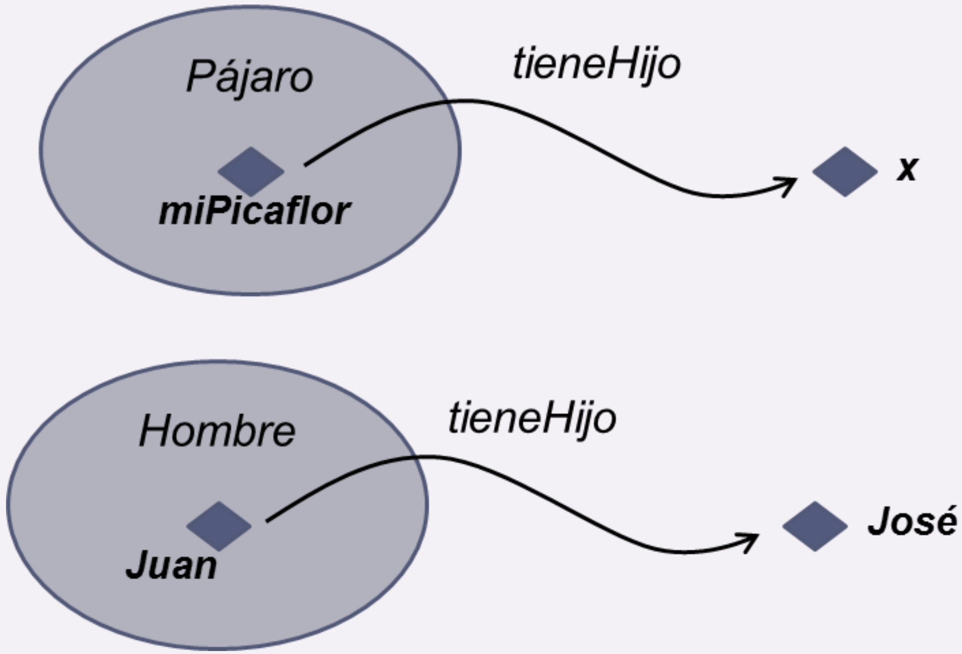
# Tableau – Mundo abierto

## Dominio y rango

**Axiomas de dominio y rango son usados por el razonador para hacer inferencias, NO para chequear inconsistencia**

$\mathcal{T} = \{\exists \mathit{tieneHijo}.\top \sqsubseteq \mathit{Hombre}\}$

$\mathcal{A} = \{\mathit{Pájaro}(\mathit{miPicaflor}),\ \mathit{Hombre}(\mathit{Juan}), \mathit{tieneHijo}(\mathit{Juan}, \mathit{José}), \mathit{tieneHijo}(\mathit{miPicaflor}, x)\}$




**¿Qué hace el razonador?**

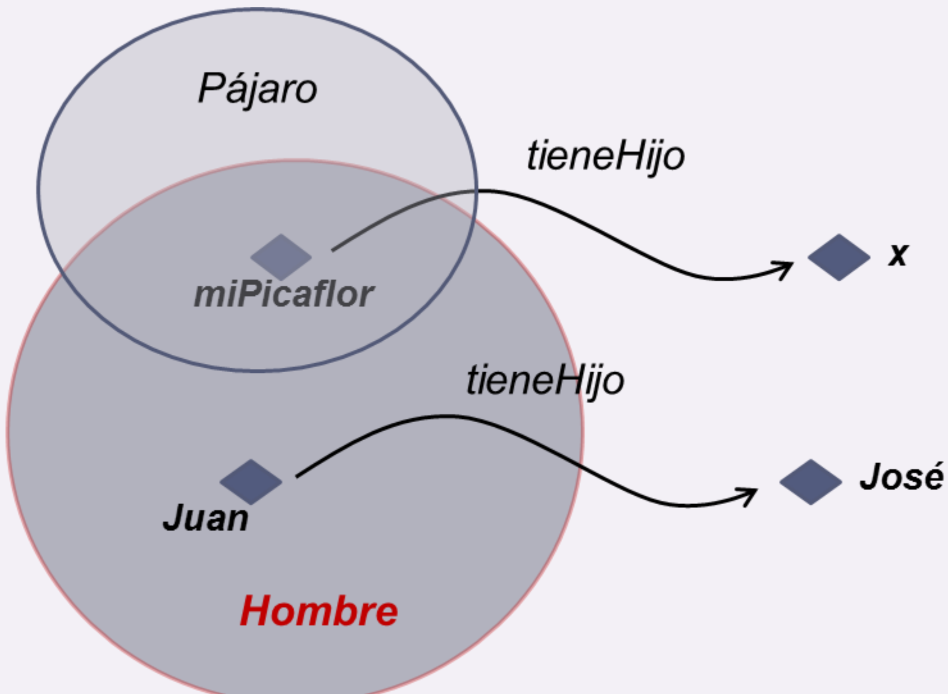
# Tableau – Mundo abierto

## Dominio y rango

**Axiomas de dominio y rango son usados por el razonador para hacer inferencias, NO para chequear inconsistencia**

$\mathcal{T} = \{\exists \textit{tieneHijo}.\top \sqsubseteq \textit{Hombre}\} = \{\neg \exists \textit{tieneHijo}.\top \sqcup \textit{Hombre}\}$

$\mathcal{A} = \{\textit{Pájaro}(\textit{miPicaflor}),\ \textit{Hombre}(\textit{Juan}),\ \textit{tieneHijo}(\textit{Juan}, \textit{José}),\ \textit{tieneHijo}(\textit{miPicaflor}, x\}$



# Ejercicio

$\mathcal{T} = \{$***Hombre* $\sqsubseteq$ *Humano, Mujer* $\sqsubseteq$ *Humano,***

***Humano* $\sqsubseteq \forall$*tieneAmigo.Mujer* $\sqcap \exists$*tienePadre.Hombre***$\}$

$\mathcal{A} = \{$*Hombre*(*Pedro*), *Hombre*(*Juan*), *Mujer*(*María*), *tieneAmigo*(*Pedro*, *María*),

*tieneAmigo*(*Pedro*, *Juan*), *tienePadre*(*María*, *Juan*)$\}$

# Ejercicio

$\mathcal{T}$ = {***Hombre* $\sqsubseteq$ *Humano, Mujer* $\sqsubseteq$ *Humano, Humano* $\sqsubseteq$ $\forall$*tieneAmigo.Mujer* $\sqcap$ $\exists$*tienePadre.Hombre***} =

{$\neg$ *Hombre* $\sqcup$ *Humano,* $\neg$ *Mujer* $\sqcup$ *Humano,* $\neg$ *Humano* $\sqcup$ ($\forall$*tieneAmigo.Mujer* $\sqcap$ $\exists$*tienePadre.Hombre*)}

$\mathcal{A}$ = {*Hombre*(*Pedro*), *Hombre*(*Juan*), *Mujer*(*María*), *tieneAmigo*(*Pedro, María*),

*tieneAmigo*(*Pedro, Juan*), *tienePadre*(*María, Juan*)}


$\mathcal{L}$(*Pedro, Juan*) = {*tieneAmigo*}

$\mathcal{L}$(*Pedro, Maria*) = {*tieneAmigo*}

$\mathcal{L}$(*Maria, Juan*) = {*tienePadre*}

# Ejercicio

$\mathcal{T}$ = {**Hombre ⊑ Humano, Mujer ⊑ Humano, Humano ⊑ ∀tieneAmigo.Mujer ⊓ ∃tienePadre.Hombre**} =

{¬ *Hombre* ⊔ *Humano*, ¬ *Mujer* ⊔ *Humano*, ¬ *Humano* ⊔ (∀*tieneAmigo.Mujer* ⊓ ∃*tienePadre.Hombre*)}

$\mathcal{A}$ = {*Hombre*(*Pedro*), *Hombre*(*Juan*), *Mujer*(*María*), *tieneAmigo*(*Pedro, María*)},

*tieneAmigo*(*Pedro, Juan*), *tienePadre*(*María, Juan*)}

*ℒ*(*x*) = {*Hombre*, ……..}


*ℒ*(*Pedro*) = {**Hombre**, ~~¬ Hombre~~ ⊔ **Humano**, ~~¬ Mujer~~ ⊔ **Humano**,
~~¬ Humano~~ ⊔ (∀tieneAmigo.Mujer ⊓ ∃tienePadre.Hombre),
∀**tieneAmigo.Mujer**, ∃**tienePadre.Hombre**}

*ℒ*(*Maria*) = {**Mujer**, ~~¬ Hombre~~ ⊔ **Humano**, ~~¬ Mujer~~ ⊔ **Humano**,
~~¬ Humano~~ ⊔ (∀tieneAmigo.Mujer ⊓ ∃tienePadre.Hombre),
∀**tieneAmigo.Mujer**, ∃**tienePadre.Hombre**}

*ℒ*(*y*) = {*Hombre*, ……..}

*ℒ*(*Juan*) = {*Hombre*, ~~¬ Hombre~~ ⊔ Humano, ~~¬ Mujer~~ ⊔ Humano,
~~¬ Humano~~ ⊔ (∀tieneAmigo.Mujer ⊓ ∃tienePadre.Hombre),
∀**tieneAmigo.Mujer**, ∃**tienePadre.Hombre**, ***Mujer***}

*ℒ*(*Pedro, Juan*) = {*tieneAmigo*}
*ℒ*(*Pedro, Maria*) = {*tieneAmigo*}
*ℒ*(*Maria, Juan*) = {*tienePadre*}

*ℒ*(*Pedro, x*) = {*tienePadre*}
*ℒ*(*Juan, y*) = {*tienePadre*}

# Ejercicio

$\mathcal{T}$ = {$\textit{\textbf{Hombre}} \sqsubseteq \textit{\textbf{Humano}}$, $\textit{\textbf{Mujer}} \sqsubseteq \textit{\textbf{Humano}}$,

$\textit{\textbf{Humano}} \sqsubseteq \forall \textit{\textbf{tieneAmigo.Mujer}} \sqcap \exists \textit{\textbf{tienePadre.Hombre}}$,

$\textit{\textbf{Hombre}} \sqcap \textit{\textbf{Mujer}} \sqsubseteq \bot$}

$\mathcal{A}$ = {*Hombre*(*Pedro*), *Hombre*(*Juan*), *Mujer*(*María*), *tieneAmigo*(*Pedro*, *María*)},

*tieneAmigo*(*Pedro*, *Juan*), *tienePadre*(*María*, *Juan*)}

# Ejercicio

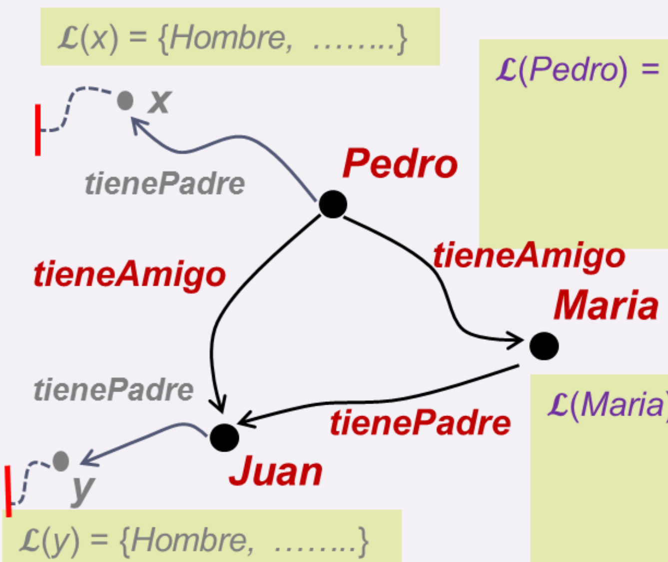
$\mathcal{T} = \{\textit{Hombre} \sqsubseteq \textit{Humano}, \textit{Mujer} \sqsubseteq \textit{Humano}, \textit{Humano} \sqsubseteq \forall \textit{tieneAmigo.Mujer} \sqcap \exists \textit{tienePadre.Hombre},$

$\textit{Hombre} \sqcap \textit{Mujer} \sqsubseteq \bot\} = \{\neg \textit{Hombre} \sqcup \textit{Humano}, \neg \textit{Mujer} \sqcup \textit{Humano},$

$\neg \textit{Humano} \sqcup (\forall \textit{tieneAmigo.Mujer} \sqcap \exists \textit{tienePadre.Hombre}), \neg \textit{Hombre} \sqcup \neg \textit{Mujer} \sqcup \bot\}$

$\mathcal{A} = \{\textit{Hombre}(\textit{Pedro}), \textit{Hombre}(\textit{Juan}), \textit{Mujer}(\textit{María}), \textit{tieneAmigo}(\textit{Pedro}, \textit{María})\},$

$\textit{tieneAmigo}(\textit{Pedro}, \textit{Juan}), \textit{tienePadre}(\textit{María}, \textit{Juan})\}$




$\mathcal{L}(\textit{Pedro}) = \{\textit{Hombre}, \cancel{\neg \textit{Hombre}} \sqcup \textit{Humano}, \cancel{\neg \textit{Mujer}} \sqcup \textit{Humano},$
$\cancel{\neg \textit{Humano}} \sqcup (\forall \textit{tieneAmigo.Mujer} \sqcap \exists \textit{tienePadre.Hombre}),$
$\forall \textit{tieneAmigo.Mujer}, \exists \textit{tienePadre.Hombre},$
$\cancel{\neg \textit{Hombre}} \sqcup \neg \textit{Mujer} \sqcup \bot\}$

$\mathcal{L}(\textit{Maria}) = \{\textit{Mujer}, \cancel{\neg \textit{Hombre}} \sqcup \textit{Humano}, \cancel{\neg \textit{Mujer}} \sqcup \textit{Humano},$
$\cancel{\neg \textit{Humano}} \sqcup (\forall \textit{tieneAmigo.Mujer} \sqcap \exists \textit{tienePadre.Hombre}),$
$\forall \textit{tieneAmigo.Mujer}, \exists \textit{tienePadre.Hombre},$
$\neg \textit{Hombre} \sqcup \cancel{\neg \textit{Mujer}} \sqcup \bot\}$

$\mathcal{L}(\textit{Juan}) = \{\textit{Hombre}, \cancel{\neg \textit{Hombre}} \sqcup \textit{Humano}, \cancel{\neg \textit{Mujer}} \sqcup \textit{Humano},$
$\cancel{\neg \textit{Humano}} \sqcup (\forall \textit{tieneAmigo.Mujer} \sqcap \exists \textit{tienePadre.Hombre}),$
$\forall \textit{tieneAmigo.Mujer}, \exists \textit{tienePadre.Hombre}, \textit{Mujer},$
$\neg \textit{Hombre} \sqcup \neg \textit{Mujer} \sqcup \bot\}$

**Contradicción!!**

# Bibliografía

Description Logics Handbook: Theory, Implementation, and Applications. Editores Franz Baader and Diego Calvanese and Deborah L. McGuinness and Daniele Nardi and Peter F. Patel-Schneider. 2003.

Capítulo 2.

Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph: Foundations of Semantic Web Technologies, Chapman & Hall/CRC, 2009.

Capítulo 5.