

Representación del Conocimiento

Lógica de Primer Orden

Dr. Alejandro Guerra-Hernández

Universidad Veracruzana

Centro de Investigación en Inteligencia Artificial
Sebastián Camancho No. 5, Xalapa, Ver., México 91000

<mailto:aguerra@uv.mx>

<http://www.uv.mx/personal/aguerra/rc>

Maestría en Inteligencia Artificial 2019



Universidad Veracruzana

Objetivo

- ▶ El uso de la lógica de primer orden para **representar** y **resolver** problemas, nos es familiar por su uso en Prolog:

`https://www.uv.mx/personal/aguerra/pia`

- ▶ Recuerden que usamos una lógica de primer orden restringida a **cláusulas de Horn** y **resolución-SLD** [3].
- ▶ Revisaremos los conceptos básicos de la lógica de primer orden en este contexto.



Universidad Veracruzana

Representación

- ▶ Cuando describimos situaciones de nuestro interés, solemos hacer uso de **enunciados declarativos**.
- ▶ Se trata de expresiones del lenguaje natural que son o bien **verdaderas**, o bien **falsas** (a diferencia de interrogativas, imperativas, etc.).
- ▶ Las proposiciones representan **hechos** que se dan o no en la realidad.
- ▶ La lógica de primer orden tienen un compromiso ontológico más fuerte [4], donde la realidad implica además, **objetos** y **relaciones** entre ellos.



Universidad Veracruzana

Razonamiento

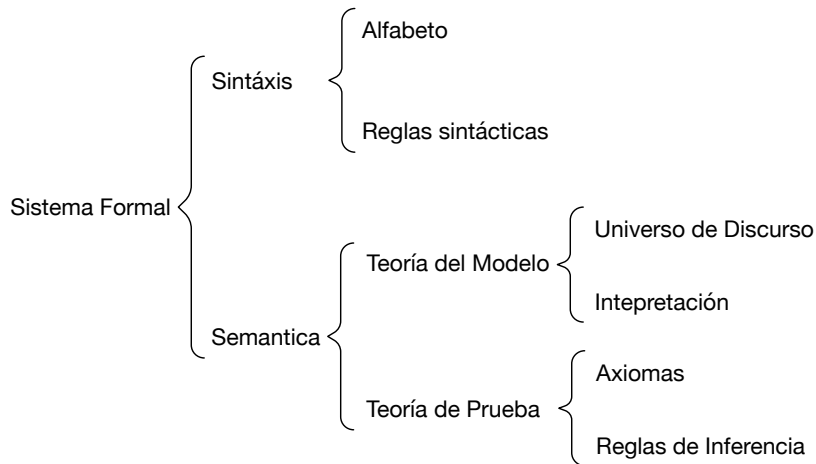
- ▶ Consideren los siguientes enunciados declarativos:
 1. Julia es madre y Luis es hijo de Julia.
 2. Toda madre ama a sus hijos.
- ▶ Conociéndolas es posible inferir:
 3. Julia ama a Luis.
- ▶ Para ello definimos un conjunto de **reglas de inferencia**, análogas a las de la deducción natural.
- ▶ Ejemplo:

$$\frac{\phi, \phi \rightarrow \psi}{\psi} (\rightarrow e)$$



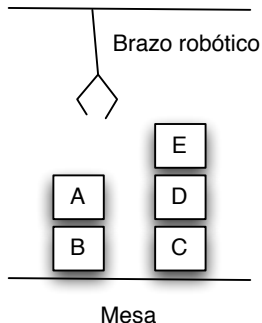
Universidad Veracruzana

Componentes de un Sistema Formal



Universidad Veracruzana

El mundo de los bloques [2]



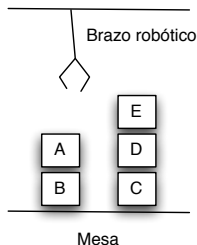
► Necesitaremos **símbolos** para:

- Objetos
- Relaciones
- Funciones
- Variables



Universidad Veracruzana

Universo de discurso



- ▶ Se denota como \mathcal{U} y es el conjunto de todos **objetos** sobre los cuales queremos expresarnos.
- ▶ Para el mundo de los bloques:

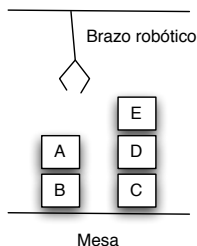
$$\mathcal{U} = \{a, b, c, d, e, \text{brazo}, \text{mesa}\}$$

- ▶ Seguiré la notación usada por Prolog, las **constantes** son cadenas que inician con minúscula.



Universidad Veracruzana

Funciones



- ▶ Son relaciones especiales entre los miembros de \mathcal{U} . Mapean un conjunto de objetos de entrada a un **objeto único** de salida.
- ▶ La función **parcial sombrero** $= \{(b, a), (c, d), (d, e)\}$:

$$\text{sombrero}(b) \mapsto a$$

- ▶ **Base funcional** es el conjunto de todas las funciones consideradas.



Universidad Veracruzana

Predicados

- ▶ Denotan **relaciones** entre los miembros de \mathcal{U} . Mapean a **falso** o **verdadero**.

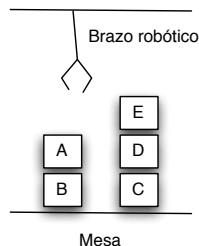
- ▶ El predicado $\text{sobre} = \{(a, b), (d, c), (e, d)\}$:

$$\text{sobre}(a, b) \mapsto \text{true}$$

- ▶ El predicado $\text{libre} = \{a, e\}$:

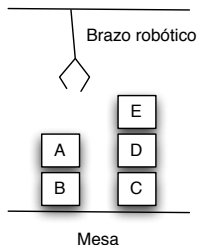
$$\text{libre}(d) \mapsto \text{false}$$

- ▶ **Base relacional** es el conjunto de todos los predicados considerados.



Universidad Veracruzana

Predicados y Funciones



- ▶ $sobre = base = \{(a, b), \dots\}$
- ▶ Los predicados pueden verse como funciones **booleanas**:

$$sobre(a, b) \mapsto true$$

- ▶ Las funciones en este contexto, tienen su codominio en algún subconjunto de \mathcal{U} :

$$base(a) \mapsto b$$



Universidad Veracruzana

Extensión de un predicado

- ▶ Para universos de discurso **finitos**, existe un límite superior en el número posible de predicados n -arios.
- ▶ Para un universo de discurso de **cardinalidad** b , existen b^n distintas n -tuplas.
- ▶ Cualquier predicado n -ario es un subconjunto de estas b^n tuplas.
- ▶ Por lo tanto, un predicado n -ario debe corresponder a uno de **máximo** $2^{(b^n)}$ conjuntos posibles.
- ▶ Caso aplicación: en programación lógica inductiva está relacionado con el tamaño de mi espacio de búsqueda!



Universidad Veracruzana

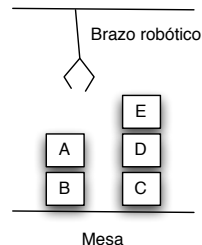
Variables y cuantificadores

- ▶ Toman valores de \mathcal{U} .
- ▶ El cuantificador “**para todo**” (\forall) expresa hechos acerca de todos los objetos en \mathcal{U} , sin necesidad de enumerarlos:

$$\forall X \text{ libre}(X) \Leftarrow \dots$$

- ▶ El cuantificador “**existe**” (\exists) expresa la existencia de un objeto en \mathcal{U} con cierta propiedad en particular:

$$\forall X \exists Y \text{ ocupado}(X) \Leftarrow \text{sobre}(Y, X)$$



Alfabeto de la Lógica de Primer Orden

Const El conjunto de símbolos de constantes;

Var El conjunto de símbolos de variables;

Pred El conjunto de símbolos de predicados;

Func El conjunto de símbolos funcionales ($Const \subset Func$);

¬ El operador monario de negación;

∨ El operador binario de disyunción;

∀ El símbolo de cuantificación universal;

() Los paréntesis.



Universidad Veracruzana

Reglas sintácticas (términos)

1. Si $\phi \in Const$ entonces $\phi \in Term$;
2. Si $\phi \in Var$, entonces $\phi \in Term$;
3. Si $\phi/n \in Func$, entonces $\phi(\phi_1, \dots, \phi_n) \in Term$ si y sólo si $\phi_{1 \leq i \leq n} \in Term$.
4. Ninguna otra expresión es un término.

► Ejemplos:

1. $a, b, c, mesa, \dots$
2. $X, Y, Z, Cubo1, Cubo2, \dots$
3. $base(b), sombrero(X), \dots$



Universidad Veracruzana

Reglas sintácticas (fórmulas bien formadas)

1. Si $\phi/n \in Pred$, entonces $\phi(\phi_0, \dots, \phi_n) \in \mathcal{L}_{FOL}$ si y sólo si $\phi_i \in Term, i = 0 \dots n$;
2. Si $\phi \in \mathcal{L}_{FOL}$, entonces $\neg\phi \in \mathcal{L}_{FOL}$;
3. Si $\phi \in \mathcal{L}_{FOL}$ y $\psi \in \mathcal{L}_{FOL}$, entonces $(\phi \vee \psi) \in \mathcal{L}_{FOL}$
4. Si $\phi \in \mathcal{L}_{FOL}$ y $X \in Vars$ es una variable que ocurre en ϕ , entonces $\forall X \phi \in \mathcal{L}_{FOL}$
5. Nada más es una fórmula bien formada (fbf).

► Ejemplos:

1. $sobre(a, b), libre(X), ama(X, hijo(X)), \dots$
2. $\neg libre(a), \neg libre(Y), \dots$
3. $sobre(X, Y) \wedge \neg libre(X), \dots$
4. $\forall X ama(X, hijo(X)), \dots$



Universidad Veracruzana

Definiciones auxiliares

Conjunción. $(\phi \wedge \psi) =_{def} \neg(\neg\phi \vee \neg\psi)$;

Implicación material. $(\phi \rightarrow \psi) =_{def} (\neg\phi \vee \psi)$;

Equivalencia material. $(\phi \equiv \psi) =_{def} ((\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi))$;

Falso. $\mathbf{f} =_{def} \neg\phi \wedge \phi$;

Verdadero. $\mathbf{t} =_{def} \neg\mathbf{f}$

Cuantificador existencial. $\exists X \phi =_{def} \neg(\forall X \neg\phi)$



Universidad Veracruzana

Términos BNF

- Un término se define como:

$$t ::= x \mid c \mid f(t, \dots, t)$$

donde $x \in Var$; $c \in Func$ tal que $|c| = 0$; y $f \in Func$ tal que $|f| > 0$.



Universidad Veracruzana

Fórmulas bien formadas BNF

- ▶ Las fbf del lenguaje de la Lógica de Primer Orden se construyen como sigue:

$$\phi ::= P(t_1, \dots, t_n) \mid \neg(\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid (\forall x \phi) \mid (\exists x \phi)$$

donde $P \in Pred$ es un símbolo de predicado de aridad $n \geq 1$; t_i denota términos; y $x \in Var$.



Universidad Veracruzana

Notación extra

- ▶ En una fbf de la forma $\forall X \phi$, se dice que la fbf ϕ está bajo el **alcance** del cuantificador $\forall X$.
- ▶ En tal caso, se dice que la ocurrencia de X en ϕ está **acotada**, en caso contrario se dice que la ocurrencia de la variable es **libre**.
- ▶ **Ejemplo.** En $\exists X \text{ sobre}(X, Y)$ la variable X está acotada, mientras que Y está libre.
- ▶ Un término sin variables se conoce como **término de base**.
- ▶ **Ejemplo.** $\text{sobre}(a, b)$.



Universidad Veracruzana

Interpretación

- ▶ Para expresar que al menos hay un bloque que no tiene nada encima, escribimos: $\exists X \text{ bloque}(X) \wedge \text{libre}(X)$.
- ▶ Cuando usamos cuantificadores siempre tenemos en mente al \mathcal{U} , en este caso $\{a, b, c, d, e, \text{brazo}, \text{mesa}\}$.
- ▶ Una **interpretación** de esta expresión es un subconjunto de \mathcal{U} tal que los miembros de ese subconjunto satisfacen el **significado esperado** de la expresión.
- ▶ **Ejemplo.** En este caso $\{a, e\}$.



Universidad Veracruzana

Teoría del modelo

- ▶ Para obtener un **modelo** para el lenguaje \mathcal{L}_{FOL} formamos el par $M = \langle D, V \rangle$, donde D es el universo de discurso y la interpretación V es una función, tal que:
 - ▶ Para cualquier predicado ϕ de aridad $n \geq 1$, $V(\phi)$ regresa las n -tuplas que corresponden a la interpretación esperada del predicado.
 - ▶ Para una constante, la función V regresa la misma constante, ej. $V(c) = c$.
- ▶ Algunas veces la expresión $V(\phi)$ se abrevia ϕ^V .



Universidad Veracruzana

Interpretación para el mundo de bloques

$$a^V = a$$

$$b^V = b$$

$$c^V = c$$

$$d^V = d$$

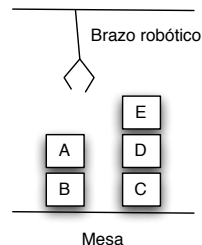
$$e^V = e$$

$$\text{sobre}^V = \{(a, b), (e, d), (d, c)\}$$

$$\text{enLaMesa}^V = \{b, c\}$$

$$\text{libre}^V = \{a, e\}$$

$$\text{porEncima}^V = \{(a, b), (e, d), (e, c), (d, c)\}$$



Interpretación y asignación de variables

- ▶ Una **interpretación** V , con respecto a un dominio de discurso D , es una función que satisface las siguientes propiedades:
 - ▶ Si $\phi \in Const$, entonces $V(\phi) = \phi$;
 - ▶ Si $\phi/n \in Pred$, tal que $n \geq 1$, entonces $V(\phi) \subseteq D^n$.
- ▶ **Ejemplo.** $libre^V \subset \{a, b, c, d, e, mesa, mano\}$.
- ▶ **Ejemplo.** $sobre^V \subset D \times D$.
- ▶ Decimos que U es una **asignación de variables** basada en el modelo $M = \langle D, V \rangle$ si para todo $\phi \in Var$, $U(\phi) \in D$.



Universidad Veracruzana

Asignación de términos

- La **asignación de términos** T , dadas la interpretación V y la asignación de variables U , es un mapeo de términos a objetos del universo de discurso que se define como sigue:
1. Si $\phi \in Const$, entonces $T_{VU}(\phi) = V(\phi)$.
 2. Si $\phi \in Var$, entonces $T_{VU}(\phi) = U(\phi)$.
 3. Si $\phi \in Term$ es de la forma $\psi(\phi_1, \dots, \phi_n)$; y $V(\psi) = g$; y $T_{VU}(\phi_i) = x_i$, entonces $T_{VU}(\psi(\phi_1, \dots, \phi_n)) = g(x_1, \dots, x_n)$.



Universidad Veracruzana

Satisfacción

- El hecho de que el enunciado ϕ se **satisfaga** bajo una interpretación V y una asignación U , se escribe:

$$\models_v \phi[U]$$

- Entonces podemos escribir $M \models V_U(\phi)$ para expresar que ϕ es verdadera en el modelo $M = \langle D, V \rangle$ cuando las variables en ϕ toman valores de acuerdo a la asignación U .
- **Ejemplo.** $M \models V_U(\text{sobre}(X, b))$ si $X \setminus a \in U$.



Universidad Veracruzana

Satisfacción:

- ▶ Dado un modelo $M = \langle D, V \rangle$ y una asignación de términos T_{VU} :
 1. $\models_V (\phi = \psi)[U]$ ssi $T_{VU}(\phi) = T_{VU}(\psi)$.
 2. $\models_V \phi(\tau_1, \dots, \tau_n)[U]$ ssi $(T_{VU}(\tau_1), \dots, T_{VU}(\tau_n)) \in \phi^V$.
 3. $\models_V (\neg\phi)[U]$ ssi $\not\models_V \phi[U]$.
 4. $\models_V (\phi \wedge \psi)[U]$ ssi $\models_V \phi[U]$ y $\models_V \psi[U]$.
 5. $\models_V (\phi \vee \psi)[U]$ ssi $\models_V \phi[U]$ o $\models_V \psi[U]$.
 6. $\models_V (\phi \rightarrow \psi)[U]$ ssi $\not\models_V \phi[U]$ o $\models_V \psi[U]$.
 7. $\models_V (\forall v \phi)[U]$ ssi para todo $d \in D$ es el caso que $\models_V \phi[W]$, donde $v^W = d$ y $\mu^W = \mu^U$ para $\mu \neq v$.
 8. $\models_V (\exists v \phi)[U]$ ssi para algún $d \in D$ es el caso que $\models_V \phi[W]$, donde $v^W = d$ y $\mu^W = \mu^U$ para $\mu \neq v$.
- ▶ A las asignaciones de variables como W , se les conoce como v -alternativas.



Definiciones complementarias

- ▶ Si una interpretación V satisface a un enunciado ϕ para toda asignación de variables, se dice que V es un **modelo** de ϕ .
- ▶ Un enunciado se dice **satisfacible** si existe alguna interpretación y asignación de variables que lo satisfaga.
- ▶ Se dice que una fbf ϕ es **válida**, si y sólo si se satisface en toda interpretación y asignación de variables.
- ▶ Las fbf válidas lo son en virtud de su **estructura lógica**, por lo que no proveen información acerca del dominio descrito.
- ▶ **Ejemplo.** Por ejemplo $p(X) \vee \neg p(X)$ es una fbf válida.



Universidad Veracruzana

Mi mamá me ama

► Retomemos el ejemplo de la introducción:

1. Toda madre ama a sus hijos.
2. Julia es madre y Luis es hijo de Julia.
3. Julia ama a Luis.

► Se puede formalizar como:

1. $\forall X \forall Y \text{ madre}(X) \wedge \text{hijo_de}(Y, X) \rightarrow \text{ama}(X, Y)$
2. $\text{madre}(\text{julia}) \wedge \text{hijo_de}(\text{luis}, \text{julia})$
3. $\text{ama}(\text{julia}, \text{luis})$



Universidad Veracruzana

Reglas de inferencia

- ▶ La inferencia puede verse como un proceso de manipulación de fbf, donde a partir las **premisas**, se producen las **conclusiones**.

- ▶ **Modus Ponens**. O eliminación de la implicación:

$$\frac{\phi \quad \phi \rightarrow \psi}{\psi} \quad (\rightarrow e)$$

- ▶ **Eliminación de cuantificador universal**..:

$$\frac{\forall X \phi(X)}{\phi(t)} \quad (\forall e)$$

- ▶ **Introducción de conjunción**..:

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \quad (\wedge i)$$



Universidad Veracruzana

Ejemplo de derivación

► Inicio:

1. $\forall X \forall Y \text{madre}(X) \wedge \text{hijo_de}(Y, X) \rightarrow \text{ama}(X, Y)$
2. $\text{madre}(\text{julia}) \wedge \text{hijo_de}(\text{luis}, \text{julia})$

► Al aplicar la eliminación de cuantificador universal ($\forall E$) a (1) obtenemos:

3. $\forall Y (\text{madre}(\text{julia}) \wedge \text{hijo_de}(Y, \text{julia}) \rightarrow \text{ama}(\text{julia}, Y))$

► Al aplicar nuevamente ($\forall E$) a (3) obtenemos:

4. $\text{madre}(\text{julia}) \wedge \text{hijo_de}(\text{luis}, \text{julia}) \rightarrow \text{ama}(\text{julia}, \text{luis})$

► Finalmente, al aplicar Modus Ponens a (2) y (4):

5. $\text{ama}(\text{julia}, \text{luis})$



Universidad Veracruzana

Solidez y robustez

- ▶ Un conjunto de reglas de inferencia se dice **sólido**, si para todo conjunto de fbf cerradas (sin ocurrencia de variables libres) Δ y cada fbf cerrada ϕ , siempre que $\Delta \vdash \phi$ se tiene que $\Delta \models \phi$.
- ▶ Las reglas de inferencia se dicen **completas** si $\Delta \vdash \phi$ siempre que $\Delta \models \phi$.



Universidad Veracruzana

Enunciados declarativos

- Describen relaciones **positivas** entre elementos de \mathcal{U} : Incondicionadas (**hechos**) y Condicionadas (**reglas**).

1. Antonio es hijo de Juan.
 2. Ana es hija de Antonio.
 3. Juan es hijo de Marcos.
 4. Alicia es hija de Juan.
 5. El nieto de una persona es el hijo del hijo de esa persona.
1. $hijo_de(antonio, juan)$
 2. $hijo_de(ana, antonio)$
 3. $hijo_de(juan, marcos)$
 4. $hijo_de(alicia, juan)$
 5. $\forall X \forall Y (nieto_de(X, Y) \leftarrow \exists Z (hijo_de(Z, Y) \wedge hijo_de(X, Z)))$



Universidad Veracruzana

Formas alternativas para una regla

- ▶ La fórmula:

$$\forall X \forall Y (nieto_de(X, Y) \leftarrow \exists Z (hijo_de(Z, Y) \wedge hijo_de(X, Z)))$$

- ▶ Se puede escribir como:

- ▶ $\forall X \forall Y (nieto_de(X, Y) \vee \neg \exists Z (hijo_de(Z, Y) \wedge hijo_de(X, Z)))$
- ▶ $\forall X \forall Y (nieto_de(X, Y) \vee \forall Z \neg (hijo_de(Z, Y) \wedge hijo_de(X, Z)))$
- ▶ $\forall X \forall Y \forall Z (nieto_de(X, Y) \vee \neg (hijo_de(Z, Y) \wedge hijo_de(X, Z)))$
- ▶ $\forall X \forall Y \forall Z (nieto_de(X, Y) \leftarrow (hijo_de(Z, Y) \wedge hijo_de(X, Z)))$

- ▶ Con **equivalencias** como:

- ▶ $\phi \rightarrow \psi \equiv \neg \phi \vee \psi$ ó
- ▶ $\forall X \phi \equiv \neg \exists X \neg \phi$.



Universidad Veracruzana

Literales

- ▶ Una **literal** es un átomo o la negación de un átomo.
- ▶ Una **literal positiva** es un átomo.
- ▶ Una **literal negativa** es la negación de un átomo.
- ▶ Ejemplos:
 - ▶ $hijo_de(juan, marcos)$.
 - ▶ $\neg hijo_de(juan, alicia)$.
- ▶ Son los bloques de construcción ϕ_i en:

$$\phi_0 \leftarrow \phi_1 \wedge \dots \wedge \phi_n \quad (n \geq 0)$$



Universidad Veracruzana

Cláusulas

- ▶ Una **cláusula** es una disyunción finita de cero o más literales.
- ▶ Una cláusula se dice **definitiva**, si tiene exactamente una literal positiva.

$$\phi_0 \vee \neg\phi_1 \vee \dots \vee \neg\phi_n \quad (n \geq 0)$$

- ▶ Lo cual es equivalente a la forma general de fbf que nos interesaba:

$$\phi_0 \leftarrow \phi_1 \wedge \dots \wedge \phi_n \quad (n \geq 0)$$

- ▶ La cláusula vacía (sin literales) se denota como \square y es equivalente a $\square \leftarrow \blacksquare$ (en nuestra notación previa $\perp \leftarrow \top$).



Universidad Veracruzana

Hechos y reglas revisitados

- ▶ En $\phi_0 \leftarrow \phi_1 \wedge \dots \wedge \phi_n$ ($n \geq 0$), tenemos que:
 - ▶ Si $n = 0$, la literal ϕ_0 será positiva, por lo que la cláusula definitiva será un **hecho**.
 - ▶ Si $n > 0$ la cláusula definitiva toma la forma de una **regla**, donde ϕ_0 es la **cabeza** de la regla; y la conjunción $\phi_1 \wedge \dots \wedge \phi_n$ su **cuerpo**.
- ▶ Una forma restringida de **cuantificación**:

$$\forall X \forall Y (nieto_de(X, Y) \vee \neg \exists Z (hijo_de(Z, Y) \wedge hijo_de(X, Z))).$$



Universidad Veracruzana

Programa y Meta Definitivos

- ▶ Un **programa definitivo** es un conjunto finito de cláusulas definitivas.
- ▶ Una cláusula sin literales positivas es una **meta definitiva**.

$$\leftarrow \phi_1 \wedge \dots \wedge \phi_n \quad (n \geq 1)$$



Universidad Veracruzana

Ejemplos

- Considere las siguientes consultas:

Consulta	Meta definitiva
¿Es Ana hija de Antonio?	$\leftarrow \text{hijo}(\text{ana}, \text{antonio})$
¿Quién es nieto de Ana?	$\leftarrow \text{nieto}(X, \text{ana})$
¿De quién es nieto Antonio?	$\leftarrow \text{nieto}(\text{antonio}, X)$
¿Quién es nieto de quién?	$\leftarrow \text{nieto}(X, Y)$

- ¿Cuales serían las respuestas computadas?



Universidad Veracruzana

Respuestas obtenidas

- ▶ Puesto que la meta no contiene variables, la respuesta es *true*.
- ▶ Puesto que el programa no contiene información sobre los nietos de Ana, la respuesta es *false*.
- ▶ Puesto que Antonio es nieto de Marcos, la respuesta obtenida sería $X = \text{marcos}$.
- ▶ La consulta final obtiene tres respuestas: $X = \text{antonio}$ $Y = \text{alicia}$, $X = \text{alicia}$ $Y = \text{marcos}$, $X = \text{ana}$ $Y = \text{juan}$.



Universidad Veracruzana

Cláusulas de Horn

- ▶ Una **cláusula de Horn** es una cláusula ó una meta definitivas.
- ▶ La cláusula vacía \square es una meta definitiva y, por lo tanto, una cláusula de Horn.
- ▶ Las cláusulas de Horn implican restricciones expresivas. Por ejemplo, no podemos expresar $p(a) \vee p(b)$
- ▶ Debido a su estructura restringida, las cláusulas de Horn son más fáciles de manipular que las cláusulas generales.



Universidad Veracruzana

Significado lógico de las metas

- El significado lógico de las metas puede explicarse haciendo referencia a la fbf equivalente cuantificada universalmente:

$$\forall X_1 \dots X_m \neg(\phi_1 \wedge \dots \wedge \phi_n)$$

donde todas las X_i ocurren en la meta.

- Equivale a:

$$\neg \exists X_1 \dots X_m (\phi_1 \wedge \dots \wedge \phi_n)$$



Universidad Veracruzana

Conocimiento positivo

- ▶ Los programas definitivos solo pueden expresar conocimiento positivo, tanto los hechos, como las reglas, nos dicen que elementos de una estructura están en una relación, pero **no nos dicen cuales no**.
- ▶ Por lo tanto, al usar el lenguaje de los programas definitivos, no es posible construir **descripciones contradictorias**, i.e., conjuntos de fbf no satisfacibles.
- ▶ **Todo programa definitivo tiene un modelo.**



Universidad Veracruzana

Modelos e interpretaciones (recordatorio)

- ▶ Sea ϕ una fbf y V una interpretación. V es un **modelo** de ϕ si $\models_V \phi$.
- ▶ Sea Δ un conjunto finito de fbf y V una interpretación. V es un modelo de Δ si $\models_V \phi$ para toda $\phi \in \Delta$.
- ▶ Una clase interesante de modelos para los programas definitivos se conoce como **interpretaciones de Herbrand**.



Universidad Veracruzana

Universo y Base de Herbrand

- ▶ Sea L un alfabeto extraído de un programa definitivo Δ , donde $|Const| \geq 1$.
- ▶ El **Universo de Herbrand** (U_Δ) es el conjunto de todos los términos formados con las constantes y funtores de L .
- ▶ La **Base de Herbrand** (B_Δ) es el conjunto de todos los átomos que pueden formarse con los predicados y los términos en U_L .



Universidad Veracruzana

Ejemplo: impar

- ▶ Sea $\Delta = \{ \text{impar}(s(0)), \text{impar}(s(s(X))) \leftarrow \text{impar}(X) \}$.
- ▶ $U_{\Delta} = \{0, s(0), s(s(0)), s(s(s(0))), \dots\}$
- ▶ $B_{\Delta} = \{ \text{impar}(0), \text{impar}(s(0)), \text{impar}(s(s(0))), \dots \}$



Universidad Veracruzana

Otro ejemplo

- ▶ Sea $\Delta = \{p(a), q(a, f(b)), q(X, X) \leftarrow p(X)\}$
- ▶ $U_{\Delta} = \{a, b, f(a), f(b), f(f(a)), f(f(b)), \dots\}$
- ▶ $B_{\Delta} = \{p(a), p(b), p(f(a)), p(f(b)), \dots\}$



Universidad Veracruzana

Interpretación de Herbrand

- ▶ Sea Δ un programa definitivo y L el alfabeto compuesto por los símbolos de Δ .
- ▶ V es una interpretación de Herbrand de Δ , si y sólo si:
 - ▶ El dominio de V es U_Δ .
 - ▶ Para cada constante $c \in L$, $c^V = c$.
 - ▶ Para cada functor $f/n \in L$, $f^V : U_\Delta^n \mapsto U_\Delta$.
 - ▶ Para cada predicado $p/n \in L$, $p^V \subseteq U_\Delta^n$.



Universidad Veracruzana

Modelo de Herbrand

- ▶ Sea Δ un programa definitivo;
- ▶ L el alfabeto compuesto por los símbolos en Δ ;
- ▶ y V una interpretación de Herbrand de Δ .
- ▶ Si V es un modelo de toda fbf en Δ , se dice que es un **modelo de Herbrand** de Δ .



Universidad Veracruzana

Ejemplo: impar

- ▶ Consideren el programa Δ en el ejemplo *impar*/1.
- ▶ Una posible interpretación de este programa es

$$\textit{impar}^V = \{s(0), s(s(s(0)))\}$$

- ▶ Una interpretación de Herbrand se puede especificar mediante una familia de tales relaciones (una por cada símbolo de predicado).



Universidad Veracruzana

Otro ejemplo

- Consideren ahora algunas interpretaciones de Herbrand de $\Delta = \{p(a), q(a, f(b)), q(X, X) \leftarrow p(X)\}$:

$$V_1 = \{p(a), p(b), q(a, b), q(b, b)\}$$

$$V_2 = \{p(a), q(a, a), q(a, f(b))\}$$

$$V_3 = \{p(f(f(a))), p(b), q(a, a), q(a, f(b))\}$$

$$V_4 = \{p(a), p(b), q(a, a), q(b, b), q(a, f(b))\}$$

- V_2 y V_4 son modelos de Herbrand de Δ
- V_1 y V_3 no lo son.



Universidad Veracruzana

Resultados de interés

- ▶ Para poder determinar si una interpretación de Herbrand V es un **modelo de una fbf** cuantificada universalmente $\forall\phi$, es suficiente verificar si ϕ es verdadera en V , para todas las asignaciones posibles de las variables de ϕ .
- ▶ Para el lenguaje restringido de cláusulas definitivas, si queremos verificar que una fbf atómica ϕ es **consecuencia de un programa definitivo** Δ basta con verificar que todo modelo de Herbrand de Δ es también un modelo de Herbrand de ϕ .



Universidad Veracruzana

Programa definitivo extendido con meta definitiva

- ▶ Sea Δ un programa definitivo y γ una meta definitiva.
- ▶ Sea L un alfabeto compuesto por los símbolos en el programa y la meta definitivos.
- ▶ Si V' es un modelo de $\Delta \cup \{\gamma\}$, entonces $V = \{\phi \in B_\Delta \mid \models_{V'} \phi\}$ es un modelo de Herbrand de $\Delta \cup \{\gamma\}$.



Universidad Veracruzana

Consistencia

- ▶ Sea Δ un programa definitivo y ϕ una cláusula definitiva.
- ▶ Sea $\Delta' = \Delta \cup \{\neg\phi\}$.
- ▶ Entonces $\Delta \models \phi$ si y sólo si Δ' no tiene modelo de Herbrand.
- ▶ Esto es, si Δ' es **no satisfacible**, lo cual es cierto sólo si Δ' no tiene modelos y por lo tanto, no tiene modelo de Herbrand.



Universidad Veracruzana

Intersección de modelos de Herbrand

- ▶ Sea M una familia **no vacía** de modelos de Herbrand de un programa definitivo Δ .
- ▶ Entonces la intersección $V = \bigcap M$ es un modelo de Herbrand de Δ .
- ▶ Al tomar la intersección de los modelos de Herbrand de un programa definitivo (todos tienen al menos un modelo, e.g., B_Δ), obtenemos el **modelo mínimo de Herbrand**.



Universidad Veracruzana

Ejemplo

- ▶ Sea Δ el programa definitivo $\{masculino(adan), femenino(eva)\}$ con su interpretación obvia.
- ▶ Δ tiene los siguientes modelos de Herbrand:
 1. $\{masculino(adan), femenino(eva)\}$
 2. $\{masculino(adan), masculino(eva), femenino(eva)\}$
 3. $\{masculino(adan), masculino(eva), femenino(adan)\}$
 4. $\{masculino(adan), masculino(eva), femenino(eva), femenino(adan)\}$
- ▶ La intersección de los modelos nos lleva a un modelo de Herbrand.
- ▶ El modelo mínimo es el único que corresponde con el **modelo pretendido** del programa.



Universidad Veracruzana

Consecuencia Lógica

- ▶ El modelo mínimo de Herbrand de un programa definitivo Δ , M_Δ , es el conjunto de todas las **consecuencias lógicas atómicas de base** del programa:

$$M_\Delta = \{\phi \in B_\Delta \mid \Delta \models \phi\}$$

- ▶ La prueba de este teorema pasa por demostrar que $M_\Delta \supseteq \{\phi \in B_\Delta \mid \Delta \models \phi\}$ y que $M_\Delta \subseteq \{\phi \in B_\Delta \mid \Delta \models \phi\}$.



Universidad Veracruzana

Operador de consecuencia inmediata

- ▶ Sea $base(\Delta)$ el conjunto de todas las cláusulas de base en Δ .
- ▶ T_Δ es una función sobre las interpretaciones de Herbrand de Δ definida como sigue:

$$T_\Delta(V) = \{\phi_0 \mid \phi_0 \leftarrow \phi_1, \dots, \phi_n \in base(\Delta) \wedge \{\phi_1, \dots, \phi_n\} \subseteq V\}$$

- ▶ Se puede demostrar que existe una interpretación mínima V , t.q. $T_\Delta(V) = V$ y que V es idéntica al modelo mínimo de Herbrand de Δ , M_Δ .



Universidad Veracruzana

Notación

- ▶ Existe una notación estándar para denotar a los miembros de esta secuencia de interpretaciones construídas a partir de Δ :

$$\begin{aligned}T_{\Delta} \uparrow 0 &= \emptyset \\T_{\Delta} \uparrow (i + 1) &= T_{\Delta}(T_{\Delta} \uparrow i) \\T_{\Delta} \uparrow n &= \bigcup_{i=0}^{\infty} T_{\Delta} \uparrow i\end{aligned}$$

- ▶ El conjunto construído de esta manera es idéntico al modelo mínimo de Herbrand de Δ .



Universidad Veracruzana

Ejemplo

- Tomando Δ como el programa de *impar*/1 tenemos:

$$T_{\Delta} \uparrow 0 = \emptyset$$

$$T_{\Delta} \uparrow 1 = \{\textit{impar}(s(0))\}$$

$$T_{\Delta} \uparrow 2 = \{\textit{impar}(s(0)), \textit{impar}(s(s(s(0))))\}$$

$$\vdots$$

$$T_{\Delta} \uparrow \omega = \{\textit{impar}(s^n(0)) \mid n \in \{1, 3, 5, \dots\}\}$$



Universidad Veracruzana

Fin

- ▶ Sea Δ un programa definitivo y M_Δ su modelo mínimo de Herbrand.
- ▶ Entonces:
 - ▶ M_Δ es la interpretación mínima de Herbrand, tal que $T_\Delta(M_\Delta) = M_\Delta$.
 - ▶ $M_\Delta = T_\Delta \uparrow \omega$.



Universidad Veracruzana

Programas y metas definitivos

- Consideren el siguiente programa definitivo Δ :

papa(juan, marta).
recien_nacido(marta).
orgullosa(X) ← padre(X, Y), recien_nacido(Y).
padre(X, Y) ← papa(X, Y).
padre(X, Y) ← mama(X, Y).

- ¿Cómo interpretamos la meta $\leftarrow \text{orgullosa}(Z)$. ?



Universidad Veracruzana

Metas

- ▶ $\leftarrow orgullosa(Z) \equiv \forall Z \neg orgullosa(Z)$
- ▶ $\forall Z \neg orgullosa(Z) \equiv \neg \exists Z orgullosa(Z)$.
- ▶ Si demostramos que ese enunciado es contradictorio en Δ , entonces sabremos que $\Delta \models \exists Z orgullosa(Z)$.
- ▶ Pero eso solo respondería *true* a la pregunta original.
- ▶ El objetivo en realidad es encontrar una **substitución** θ tal que el conjunto $\Delta \cup \{\neg orgullosa(Z)\theta\}$ sea insatisfacible, lo que equivale a que $\Delta \models orgullosa(Z)\theta$.
- ▶ **Ejemplo.** $\Delta \cup \neg orgullosa(Z)\{Z/juan\}$



Universidad Veracruzana

Razonamiento

- ▶ Asumimos la meta γ_0 – Para todo Z , Z no está orgulloso.
- ▶ Una regla en Δ describe una condición para que alguien esté orgulloso:

$$\text{orgulloso}(X) \leftarrow \text{padre}(X, Y), \text{recien_nacido}(Y)$$

- ▶ Lo cual es lógicamente equivalente a:

$$\forall (\neg \text{orgulloso}(X) \rightarrow \neg(\text{padre}(X, Y) \wedge \text{recien_nacido}(Y)))$$



Universidad Veracruzana

Estrategia

- ▶ Partiendo de:

$$\forall (\neg \text{orgullosa}(X) \rightarrow \neg(\text{padre}(X, Y) \wedge \text{recien_nacido}(Y)))$$

- ▶ Al renombrar X por Z , eliminar el cuantificador universal y usar *modus ponens* con respecto a γ_0 , obtenemos γ_1 :

$$\neg(\text{padre}(Z, Y) \wedge \text{recien_nacido}(Y))$$

- ▶ o su equivalente:

$$\leftarrow \text{padre}(Z, Y), \text{recien_nacido}(Y).$$

- ▶ γ_1 que es verdadera en todo modelo $\Delta \cup \{\gamma_0\}$.



Universidad Veracruzana

Resolución

- ▶ Ahora solo queda probar que $\Delta \cup \{\gamma_1\}$ es no satisfacible. Observen que γ_1 es equivalente a la fbf:

$$\forall Z \forall Y (\neg \text{padre}(Z, Y) \vee \neg \text{recien_nacido}(Y))$$

- ▶ γ_1 no es satisfacible para Δ , si en todo modelo de Δ hay una persona que es padre de un recién nacido:

$$\text{padre}(X, Y) \leftarrow \text{papa}(X, Y).$$

- ▶ Por lo que γ_1 se reduce a γ_2 :

$$\leftarrow \text{papa}(Z, Y), \text{recien_nacido}(Y).$$



Universidad Veracruzana

Estrategia recursiva

- ▶ El programa declara que *juan* es padre de *marta*:

papa(juan, marta).

- ▶ Así que sólo resta probar que “*marta* no es una recién nacida” no se puede satisfacer junto con Δ :

$\leftarrow \text{recien_nacido}(marta).$

- ▶ pero el programa contiene el hecho:

recien_nacido(marta).

- ▶ equivalente a $\neg \text{recien_nacido}(marta) \rightarrow \square$
- ▶ lo que conduce a una refutación \square .



Universidad Veracruzana

Resumiendo

- ▶ Para probar la existencia de algo:
 - ▶ Suponer lo **opuesto**
 - ▶ y usar **modus ponens** y la regla de **eliminación del cuantificador universal**,
 - ▶ para encontrar un **contra ejemplo** al supuesto.



Universidad Veracruzana

Unificador

- ▶ Una meta es un conjunto de átomos a ser probados.
- ▶ Seleccionamos un átomo de la meta $p(s_1, \dots, s_n)$ y una cláusula del programa con la forma $p(t_1, \dots, t_n) \leftarrow A_1, \dots, A_n$
- ▶ Si encontramos una substitución θ que hace que $p(s_1, \dots, s_n)\theta$ y $p(t_1, \dots, t_n)\theta$ sean idénticos. Tal substitución se conoce como **unificador**.
- ▶ La nueva meta se construye remplazando el átomo seleccionado en la meta original, por los átomos de la cláusula seleccionada, aplicando θ a todos los átomos obtenidos de esta manera.



Universidad Veracruzana

Resolución-SLD

- ▶ La regla de inferencia **principio de resolución SLD** combina *modus ponens*, *eliminación del cuantificador universal* y en el paso final un *reductio ad absurdum*.
- ▶ Si se prueba en k pasos que la meta definitiva en cuestión no puede satisfacerse, probamos que:

$$\leftarrow (\phi_1, \dots, \phi_m)\theta_1 \dots \theta_k$$

- ▶ es una instancia que no puede satisfacerse. Por tanto:

$$\Delta \models (\phi_1 \wedge \dots \wedge \phi_m)\theta_1 \dots \theta_k$$



Universidad Veracruzana

Observaciones

- ▶ Esta computación **no es determinista**: Cualquier átomo de la meta puede ser seleccionado y pueden haber varias cláusulas del programa que unifiquen con el átomo seleccionado.
- ▶ Pueden existir **unificadores alternativos** para dos átomos.
- ▶ Es posible que el átomo seleccionado **no unifique** con ninguna cláusula en el programa, i.e., que no sea posible construir un contra ejemplo para la meta.
- ▶ La computación puede caer en un **ciclo**, sin producir solución alguna.



Universidad Veracruzana

Substitución

- ▶ Una **substitución** θ es un conjunto finito de la forma:

$$\{X_1/t_1, \dots, X_n/t_n\}, \quad (n \geq 0)$$

donde las X_i son variables, **distintas** entre si, y los t_i son términos.

- ▶ Decimos que t_i substituye a X_i . La forma X_i/t_i se conoce como **ligadura** de X_i .
- ▶ La substitución θ se dice se dice **de base** si cada término t_i es un término base.
- ▶ La **substitución vacía** se conoce como **substitución de identidad** y se denota por ϵ .



Universidad Veracruzana

Caso (expresiones)

- ▶ Sea $\theta = \{X_1/t_1, \dots, X_n/t_n\}$ una substitución y ϕ una expresión. Entonces $\phi\theta$, el **caso** (*instance*) de ϕ bajo θ , es la expresión obtenida al substituir simultáneamente X_i por t_i para $1 \leq i \leq n$.
- ▶ Si $\phi\theta$ es una expresión de base, se dice que es un **caso base** y se dice que θ es una substitución de base para ϕ .
- ▶ Si $\Phi = \{\phi_1, \dots, \phi_n\}$ es un conjunto finito de expresiones, entonces $\Phi\theta$ denota $\{\phi_1\theta, \dots, \phi_n\theta\}$.



Universidad Veracruzana

Ejemplo

- ▶ Sea ϕ la expresión $p(Y, f(X))$ y
- ▶ Sea θ la substitución $\{X/a, Y/g(g(X))\}$.
- ▶ El caso de ϕ bajo θ es $\phi\theta = p(g(g(X)), f(a))$.
- ▶ Observen que X e Y son simultáneamente remplazados por sus respectivos términos, lo que implica que X en $g(g(X))$ no es afectada por X/a .



Universidad Veracruzana

Composición de substituciones

- ▶ Sean $\theta = \{X_1/s_1, \dots, X_m/s_m\}$ y $\sigma = \{Y_1/t_1, \dots, Y_n/t_n\}$ dos substituciones.
- ▶ Consideren la secuencia:

$$X_1/(s_1\sigma), \dots, X_m/(s_m\sigma), Y_1/t_1, \dots, Y_n/t_n$$

- ▶ Si se borran de esta secuencia las ligaduras $X_i/s_i\sigma$ cuando $X_i = s_i\sigma$ y cualquier ligadura Y_j/t_j donde $Y_j \in \{X_1, \dots, X_m\}$, obtenemos la **composición** de θ y σ denotada por $\theta\sigma$.



Universidad Veracruzana

Ejemplo

- ▶ Sea $\theta = \{X/f(Y), Z/U\}$ y $\sigma = \{Y/b, U/Z\}$.
- ▶ Construimos la secuencia de ligaduras
 $X/(f(Y)\sigma), Z/(U)\sigma, Y/b, U/Z$ lo cual es $X/f(b), Z/Z, Y/b, U/Z$.
- ▶ Al borrar la ligadura Z/Z obtenemos la secuencia
 $X/f(b), Y/b, U/Z = \theta\sigma$.



Universidad Veracruzana

Caso (substituciones)

- ▶ Sean θ y σ dos substituciones. Se dice que θ es un **caso** de σ , si existe una substitución γ , tal que $\sigma\gamma = \theta$.
- ▶ La substitución $\theta = \{X/f(b), Y/a\}$ es un caso de $\sigma = \{X/f(X), Y/a\}$, puesto que $\sigma\{X/b\} = \theta$.



Universidad Veracruzana

Propiedades de las substituciones

- Sea ϕ una expresión, y sea θ , σ y γ substituciones. Las siguientes relaciones se cumplen:
1. $\theta = \theta\epsilon = \epsilon\theta$
 2. $\phi\theta = \theta\phi$
 3. $(\phi\theta)\sigma = \phi(\theta\sigma)$



Universidad Veracruzana

Unificación

- ▶ Uno de los pasos principales en el cómputo de una meta, es que dos fbf atómicas se vuelvan sintácticamente equivalentes.
- ▶ Este proceso se conoce como **unificación** y posee una solución algorítmica.
- ▶ Sean ϕ y ψ términos. Una substitución θ tal que ϕ y ψ sean idénticos ($\phi\theta = \psi\theta$) es llamada **unificador** de ϕ y ψ .
- ▶ $unifica(conoce(juan, X), conoce(Y, Z)) = \{Y/juan, X/Z\}$



Universidad Veracruzana

Unificador más general (MGU)

- ▶ Una sustitución θ se dice **más general** que una sustitución σ , si y sólo si existe una sustitución γ tal que $\sigma = \theta\gamma$.
- ▶ Un unificador θ se dice el **unificador más general** (MGU: *Most General Unifier*) de dos términos, si y sólo si θ es más general que cualquier otro unificador entre esos términos.



Universidad Veracruzana

Forma resuelta y MGU

- ▶ Un conjunto de ecuaciones $\{X_1 = t_1, \dots, X_n = t_n\}$ está en **forma resuelta**, si y sólo si X_1, \dots, X_n son variables distintas que no ocurren en t_1, \dots, t_n .
- ▶ Sea $\{X_1 = t_1, \dots, X_n = t_n\}$ un conjunto de ecuaciones en forma resuelta. Entonces $\{X_1/t_1, \dots, X_n/t_n\}$ es un MGU (idempotente) de la forma resuelta.



Universidad Veracruzana

Equivalencia

- ▶ Dos conjuntos de ecuaciones E_1 y E_2 se dicen **equivalentes**, si tienen el mismo conjunto de unificadores.
- ▶ Para computar el MGU de dos términos ϕ y ψ , primero intente transformar la ecuación $\{\phi = \psi\}$ en una forma resuelta equivalente. Si esto falla, entonces $mgu(\phi, \psi) = \text{fallo}$. Sin embargo, si una forma resuelta $\{X_1 = t_1, \dots, X_n = t_n\}$ existe, entonces $mgu(\phi, \psi) = \{X_1/t_1, \dots, X_n/t_n\}$.



Universidad Veracruzana

Algoritmo de unificación

```

1: function UNIFICA( $E$ )
2:   repeat
3:      $(s = t) \leftarrow \text{seleccionar}(E)$ 
4:     if  $f(s_1, \dots, s_n) = f(t_1, \dots, t_n)$  ( $n \geq 0$ ) then
5:       reemplazar  $(s = t)$  por  $s_1 = t_1, \dots, s_n = t_n$ 
6:     else if  $f(s_1, \dots, s_m) = g(t_1, \dots, t_n)$  ( $f/m \neq g/n$ ) then
7:       return(fallo)
8:     else if  $X = X$  then
9:       remover la  $X = X$ 
10:    else if  $t = X$  then
11:      reemplazar  $t = X$  por  $X = t$ 
12:    else if  $X = t$  then
13:      if subtermino( $X, t$ ) then
14:        return(fallo)
15:      else reemplazar todo  $X$  por  $t$ 
16:      end if
17:    end if
18:  until No hay accion posible para  $E$ 
19: end function

```

▷ E es un conjunto de ecuaciones



Universidad Veracruzana

Ejemplo 1

- El conjunto $\{f(X, g(Y)) = f(g(Z), Z)\}$ tiene una forma resuelta, pues:

$$\begin{aligned}\{f(X, g(Y)) = f(g(Z), Z)\} &\rightarrow \{X = g(Z), g(Y) = Z\} \\ &\rightarrow \{X = g(Z), Z = g(Y)\} \\ &\rightarrow \{X = g(g(Y)), Z = g(Y)\}\end{aligned}$$



Universidad Veracruzana

Ejemplo 2

- El conjunto $\{f(X, g(X), b) = f(a, g(Z), Z)\}$ no tiene forma resuelta, puesto que:

$$\rightarrow \{X = a, g(X) = g(Z), b = Z\}$$

$$\rightarrow \{X = a, g(a) = g(Z), b = Z\}$$

$$\rightarrow \{X = a, a = Z, b = Z\}$$

$$\rightarrow \{X = a, Z = a, b = Z\}$$

$$\rightarrow \{X = a, Z = a, b = a\}$$

$$\rightarrow \text{fallo}$$



Universidad Veracruzana

Ejemplo 3

- El conjunto $\{f(X, g(X)) = f(Z, Z)\}$ no tiene forma resuelta, puesto que:

$$\rightarrow \{X = Z, g(X) = Z\}$$

$$\rightarrow \{X = Z, g(Z) = Z\}$$

$$\rightarrow \{X = Z, Z = g(Z)\}$$

\rightarrow *fallo*



Universidad Veracruzana

Consideraciones

- ▶ El algoritmo termina y regresa una forma resuelta equivalente al conjunto de ecuaciones de su entrada, o falla si la forma resuelta no existe.
- ▶ Computar $subtermino(X, t)$ (verificación de ocurrencia) hace que el algoritmo sea altamente ineficiente.
- ▶ El standard ISO Prolog (1995) declara que la unificación es **no decidable**.
- ▶ Al eliminar la verificación de ocurrencia es posible que al intentar resolver $X = f(X)$ obtengamos $X = f(f(X)) \dots = f(f(f \dots))$.



Universidad Veracruzana

Formalizando

- El método de razonamiento descrito informalmente al inicio de esta sesión, puede resumirse con la siguiente regla de inferencia:

$$\frac{\forall \neg(\phi_1 \wedge \dots \wedge \phi_{i-1} \wedge \phi_i \wedge \phi_{i+1} \wedge \dots \wedge \phi_m) \quad \forall (\psi_0 \leftarrow \psi_1 \wedge \dots \wedge \psi_n)}{\forall \neg(\phi_1 \wedge \dots \wedge \phi_{i-1} \wedge \psi_1 \wedge \dots \wedge \psi_n \wedge \phi_{i+1} \wedge \dots \wedge \phi_m)\theta}$$

- donde:

1. ϕ_1, \dots, ϕ_m son fbf atómicas.
2. $\psi_0 \leftarrow \psi_1, \dots, \psi_n$ es una cláusula definitiva en el programa Δ ($n \geq 0$).
3. $MGU(\phi_i, \psi_0) = \theta$.



Universidad Veracruzana

Observaciones

- ▶ La regla tiene dos premisas: una **meta** y una **cláusula definitiva**.
- ▶ El alcance de los cuantificadores es **disjunto**.
- ▶ Solo hay un cuantificador universal para la conclusión. Se requiere que el conjunto de variables en las premisas sea **disjunto**.
- ▶ Puesto que todas las variables en las premisas están cuantificadas, es siempre posible **renombrar** las variables de la cláusula definitiva para cumplir con esta condición.



Universidad Veracruzana

S de selección

- ▶ La meta definida puede incluir muchas fbf atómicas que unifican con la cabeza de alguna cláusula en el programa.
- ▶ Es deseable contar con un mecanismo determinista para seleccionar un átomo ϕ_i a unificar.
- ▶ Se asume una función que selecciona una submeta de la meta definida (**función de selección**).



Universidad Veracruzana

Usando la resolución-SLD

- ▶ El punto de partida es una meta definida γ_0 :

$$\leftarrow \phi_1, \dots, \phi_m \quad (m \geq 0)$$

- ▶ Una submeta ϕ_i será seleccionada. Una nueva meta γ_1 se construye al seleccionar una cláusula del programa $\psi_0 \leftarrow \psi_1, \dots, \psi_n$ ($n \geq 0$) cuya cabeza ψ_0 unifica con ϕ_i , resultando en θ_1 . γ_1 tiene la forma:

$$\leftarrow (\phi_1, \dots, \phi_{i-1}, \psi_1, \dots, \psi_n, \dots, \phi_m) \theta_1$$

- ▶ Ahora es posible aplicar el principio de resolución a γ_1 para obtener γ_2 , y así sucesivamente.



Universidad Veracruzana

Terminación

- ▶ El proceso puede terminar o no. Hay dos situaciones donde no es posible obtener γ_{i+1} a partir de γ_i :
 1. cuando la submeta seleccionada no puede ser resuelta (no es unificable con la cabeza de una cláusula del programa).
 2. cuando $\gamma_i = \square$ (meta vacía = **f**).



Universidad Veracruzana

Derivación-SLD

- Sea γ_0 una meta definitiva, Δ un programa definitivo y \mathcal{R} una función de selección. Una **derivación SLD** de γ_0 (usando Δ y \mathcal{R}) es una secuencia finita o infinita de metas:

$$\gamma_0 \xrightarrow{\phi_0} \gamma_1 \dots \gamma_{n-1} \xrightarrow{\phi_{n-1}} \gamma_n$$

- $\phi_i \in \Delta$. Las variables en ϕ_i se renombran con subíndices i .



Universidad Veracruzana

Substitución computada

- ▶ Cada derivación SLD nos lleva a una secuencia de MGUs $\theta_1, \dots, \theta_n$.
La composición

$$\theta = \begin{cases} \theta_1 \theta_2 \dots \theta_n & \text{si } n > 0 \\ \epsilon & \text{si } n = 0 \end{cases}$$

de MGUs se conoce como la **substitución computada** de la derivación.



Universidad Veracruzana

Ejemplo

- Consideren la meta definida $\leftarrow \text{orgullosa}(Z)$ y el programa del inicio de esta clase. Entonces

$$\gamma_0 = \leftarrow \text{orgullosa}(Z).$$

$$\phi_0 = \text{orgullosa}(X_0) \leftarrow \text{padre}(X_0, Y_0), \text{recien_nacido}(Y_0).$$

- La unificación de $\text{orgullosa}(Z)$ y $\text{orgullosa}(X_0)$ nos da el MGU $\theta_1 = \{X_0/Z\}$. Asumamos que nuestra función de selección es tomar la **submeta más a la izquierda**:

$$\gamma_1 = \leftarrow \text{padre}(Z, Y_0), \text{recien_nacido}(Y_0).$$

$$\phi_1 = \text{padre}(X_1, Y_1) \leftarrow \text{papa}(X_1, Y_1).$$

$$\text{con } \theta_2 = \{X_1/Z, Y_1/Y_0\}.$$



Universidad Veracruzana

Ejemplo

- ▶ La derivación continua como sigue:

$$\gamma_2 = \leftarrow \text{papa}(Z, Y_0), \text{recien_nacido}(Y_0).$$

$$\phi_2 = \text{papa}(\text{juan}, \text{marta}).$$

$$\gamma_3 = \leftarrow \text{recien_nacido}(\text{marta}).$$

$$\phi_3 = \text{recien_nacido}(\text{marta}).$$

$$\gamma_4 = \square$$

- ▶ la substitución computada para esta derivación es:

$$\begin{aligned} \theta_1\theta_2\theta_3\theta_4 &= \{X_0/Z\}\{X_1/Z, Y_1/Y_0\}\{Z/\text{juan}, Y_0/\text{marta}\} \epsilon \\ &= \{X_0/\text{juan}, X_1/\text{juan}, Y_1/\text{marta}, \\ &\quad Z/\text{juan}, Y_0/\text{marta}\} \end{aligned}$$



Universidad Veracruzana

Refutación-SLD y derivación fallida

- ▶ Una derivación SLD finita:

$$\gamma_0 \xrightarrow{\phi_0} \gamma_1 \dots \gamma_{n-1} \xrightarrow{\phi_{n-1}} \gamma_n$$

donde $\gamma_n = \square$, se llama **refutación SLD** de γ_0 .

- ▶ Una derivación de la meta γ_0 cuyo último elemento no es la meta vacía y no puede resolverse con ninguna cláusula del programa, es llamada **derivación fallida**.



Universidad Veracruzana

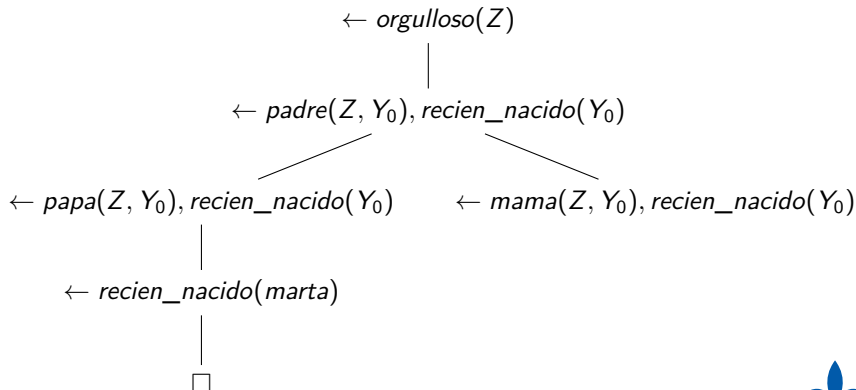
Arbol-SLD

- ▶ Sea Δ un prog. definitivo, γ_0 una meta definitiva, y \mathcal{R} una función de selección. El **árbol-SLD** de γ_0 (usando Δ y \mathcal{R}) es un árbol etiquetado, posiblemente infinito, que cumple con:
 - ▶ La raíz del árbol está etiquetada por γ_0 .
 - ▶ Si el árbol contiene un nodo etiquetado como γ_i y existe una cláusula renombrada $\phi_i \in \Delta$ tal que γ_{i+1} es derivada de γ_i y ϕ_i via \mathcal{R} , entonces el nodo etiquetado como γ_i tiene un hijo etiquetado γ_{i+1} . El arco entre ambos es ϕ_i .



Universidad Veracruzana

Ejemplo



Universidad Veracruzana

Propiedades de la resolución-SLD

Correctez. Sea Δ un programa definitivo, \mathcal{R} una función de selección, y θ una sustitución de respuesta computada a partir de Δ y \mathcal{R} para una meta $\leftarrow \phi_1, \dots, \phi_m$. Entonces $\forall ((\phi_1 \wedge \dots \wedge \phi_m)\theta)$ es una consecuencia lógica del programa Δ .

Completez. Sea Δ un programa definitivo, \mathcal{R} una función de selección y $\leftarrow \phi_1, \dots, \phi_m$ una meta definitiva. Si $\Delta \models \forall ((\phi_1 \wedge \dots \wedge \phi_m)\sigma)$, entonces existe una refutación de $\leftarrow \phi_1, \dots, \phi_m$ vía \mathcal{R} con una sustitución de respuesta computada θ , tal que $(\phi_1 \wedge \dots \wedge \phi_m)\sigma$ es un caso de $(\phi_1 \wedge \dots \wedge \phi_m)\theta$.



Universidad Veracruzana

Base de conocimientos

- ▶ Ahora podemos explorar en detalle el proceso de creación de una **base de conocimientos** (Δ).
- ▶ Recuerden que el conocimiento implica asumir que el mundo satisface alguna propiedad, expresada como un **enunciado declarativo**.
- ▶ Δ está formada por una colección de tales enunciados y nosotros asumimos que las proposiciones expresadas por ellos son las **creencias** de un agente putativo.



Universidad Veracruzana

Consideraciones

- ▶ ¿Qué es lo que queremos (o nuestro agente quiere) **computar**?
- ▶ Establecer las **razones** por las que la inferencia es necesaria en nuestros sistemas y el **número de veces** que debe llevarse a cabo.
- ▶ Establecer una **ontología**:
 - ▶ Las clases de **objeto** que nos interesan;
 - ▶ Las **propiedades** que esos objetos pueden tener;
 - ▶ Las **relaciones** que puedan darse entre ellos.
- ▶ A este proceso que se orienta a Δ desde el nivel del conocimiento, se le conoce como **Ingeniería del Conocimiento** [1].



Universidad Veracruzana

Vocabulario

- ▶ Comenzar por el conjunto de **predicados** y **funciones** dependientes del dominio.
- ▶ ¿Qué clases de **objetos** habrá en nuestro sistema?
- ▶ Las constantes serán usadas para representar **individuos con nombre**:
Ej. alejandroGuerra, gabrielAcosta, etc.
- ▶ Es posible que necesitemos **múltiples identificadores**:
Ej. 6183 puede denotar al mismo individuo que alejandroGuerra.
- ▶ **Otros** individuos con nombre: Entidades legales, lugares, objetos.
Ej. iphone2, macBookPro1, ciia, etc.



Universidad Veracruzana

Tipos de objetos

- ▶ Ahora será necesario establecer los **tipos** de objetos que emergen de los individuos con nombre adoptados.
- ▶ Para ello solemos usar **predicados** de aridad uno:
Ej. `personal(6183)`, `prof(gabrielAcosta)`, `centro(ciia)`, etc.



Universidad Veracruzana

Atributos

- ▶ Los predicados unarios también pueden representar **propiedades** de nuestros objetos:
Ej. `sni1(6183)`, `sni2(efrenMezura)`, `pnpc(ciia)`, etc.
- ▶ Observen que no podemos distinguir entre atributos y tipos de objeto. Si esto es necesario, el lenguaje FOL podría extenderse.



Universidad Veracruzana

Relaciones y funciones

- ▶ Las **relaciones** están representadas como predicados n -arios:
Ej. `trabajaEn(6183,ciia)`, `profDe(6183,rc)`, etc.
- ▶ No olviden que hay relaciones que no son binarias:
Ej. `horario(rc,8,10,[martes,jueves])`, etc.
- ▶ Las **funciones** pueden tomar varios argumentos, pero suelen ser unarias: Ej. `jefeDe(6183) \mapsto rubenDeLaMora`.
- ▶ Todas las funciones son totales, si hay alguien sin jefe en el dominio, deberíamos condierar definir `jefeDe` como una relación binaria:
Ej. `jefeDe(6183,rubenDeLaMora)`, etc.



Universidad Veracruzana

Hechos y reglas

- ▶ Con este vocabulario nuestros hechos simples pueden representarse como **literales**, i.e., predicados atómicos o su negación:
Ej. $\text{prof}(6183)$, $\neg \text{jubilado}(6183)$, etc.
- ▶ Estos hechos constituyen una **base de datos** que podría almacenarse como una **tabla relacional**.
- ▶ Otros hechos básicos son los que tiene que ver con **igualdad**:
Ej. $\text{rubenDeLaMora} = \text{coordinador}(\text{ciia})$, etc.
- ▶ También podemos definir reglas:
Ej. $\text{investigadorEn}(\text{Invest}, \text{Centro}) \text{ :- sni1}(\text{Invest}), \text{centro}(\text{Centro})$.



Universidad Veracruzana

Reglas terminológicas I

Disjunto. Dos predicados son disjuntos si la neg. de uno implica al otro:

Ej. `hombre(X) :- not mujer(X).`

Subtipo. Un predicado subsume al otro:

Ej. `medico(X) :- cirujano(X).`

Exhaustivo. Dos o más tipos completan el concepto:

Ej. `sni(X) :- candidato(X); sni1(X); sni2(X);
sni3(X); emerito(X).`

Simétrico. Definen una relación simétrica:

Ej. `colega(X,Y) :- colega(Y,X).`

Inverso. Definen una relación inversa:

Ej. `padre(X,Y) :- hijo(Y,X).`

Restricción. Establecen una restricción de tipo de objeto:

Ej. `profEn(Prof,dia) :- sni(Prof).`



Universidad Veracruzana

Reglas terminológicas II

Definición. Definen un término:

```
Ej. matutino(Curso) :- horario(Curso,_,Fin,_),  
    Fin<14.
```



Universidad Veracruzana

Individuos abstractos

- ▶ Consideren las posibles representaciones de que nic compró una bici:
 - ▶ `compra(nic, bici)`
 - ▶ `compra(nic, bici, costco)`
 - ▶ `compra(nic, bici, costco, 12000)`
 - ▶ `compra(nic, bici, costco, 12000, feb14)`
 - ▶ etc.
- ▶ Solución: Definir un individuo abstracto y tantas relaciones binarias o funciones unarias como sean necesarias:
Ej. `compra(f9872). precio(f9872,12000).`, etc.
- ▶ Permite: `precioEnDolares(C,PU$) :- precio(C,PMx),`
`PU$ = PMx/18.`



Universidad Veracruzana

Otros hechos

Estadísticos. Incluyen información sobre la probabilidad o la proporción de individuos que satisfacen el predicado.

Ej. La mayoría de los empleados está de vacaciones.

Defaults. Incluye características que normalmente son razonables de asumir, al menos que uno sea advertido de lo contrario.

Ej. Las aves vuelan.

Intencionales. Actitudes proposicionales.

Ej. Este SNI cree que es hora de irse de vacaciones y lo desea.



Universidad Veracruzana

Referencias I



R Brachman y H Levesque. *Knowledge representation and reasoning*. San Francisco, CA., USA: Morgan Kaufmann Publishers, 2004.



MR Genesereth y NJ Nilsson. *Logical Foundations for Artificial Intelligence*. Palo Alto, CA., USA: Morgan Kauffman Publishers, Inc., 1987.



SH Nienhuys-Cheng y R de Wolf. *Foundations of Inductive Logic Programming*. Ed. por JG Carbonell y J Sickmann. Vol. 1228. *Lecture Notes in Artificial Intelligence*. Berlin Heidelberg: Springer-Verlag, 1997.



S Russell y P Norvig. *Artificial Intelligence: A Modern Approach*. Tercera. Prentice Hall Series in Artificial Intelligence. USA: Prentice Hall, 2009.



Universidad Veracruzana