Programación Lógica y Funcional

Programación Lógica

Dr. Alejandro Guerra-Hernández

Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana
Campus Sur, Calle Paseo Lote II, Sección Segunda No 112,
Nuevo Xalapa, Xalapa, Ver., México 91097

mailto:aguerra@uv.mx
https://www.uv.mx/personal/aguerra/plf

Maestría en Inteligencia Artificial 2025

Universidad Veracruzana

Objetivo

- A la programación lógica le concierne el uso de la Lógica de Primer Orden para representar y resolver problemas.
- Usamos una Lógica de Primer Orden restringida a cláusulas de Horn como lenguaje de representación y resolución-SLD como algoritmo de razonamiento [4, 2].

Universidad Veracruzana

Representación

- Cuando describimos situaciones de nuestro interés, usamos enunciados declarativos.
- Ej. Los martes hay clase de programación lógica y funcional.
- Son expresiones del lenguaje natural que son o no el caso (a diferencia de las interrogativas, admirativas, imperativas, etc.).
- Si bien las proposiciones representan este tipo de expresiones, los predicados de la Lógica de Primer Orden tienen un compromiso ontológico más fuerte [5], donde la realidad implica 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 asumimos un conjunto de reglas de inferencia.
- Ejemplo: *Modus Ponens*

$$\frac{P, P \rightarrow Q}{Q} \quad (M.P.)$$

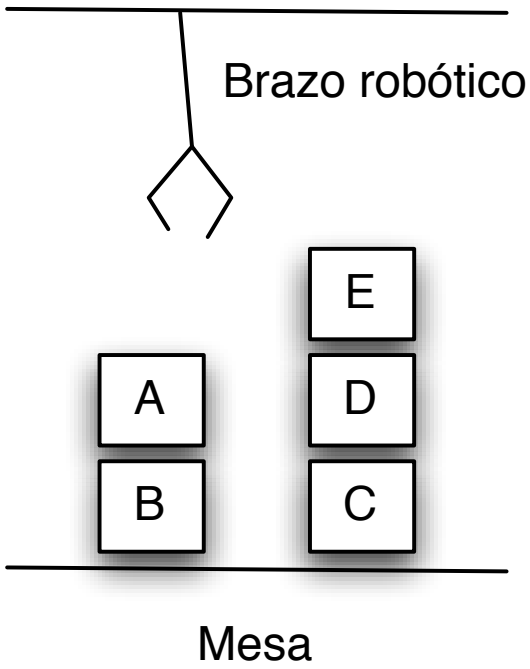
Universidad Veracruzana

Componentes de un Sistema Formal

- Sistema Formal
  - Sintáxis
    - Alfabeto
    - Reglas sintácticas
    - Teoría de Prueba
      - Axiomas
      - Reglas de Inferencia
  - Semantica
    - Teoría del Modelo
      - Universo de Discurso
      - Intepretación

Universidad Veracruzana

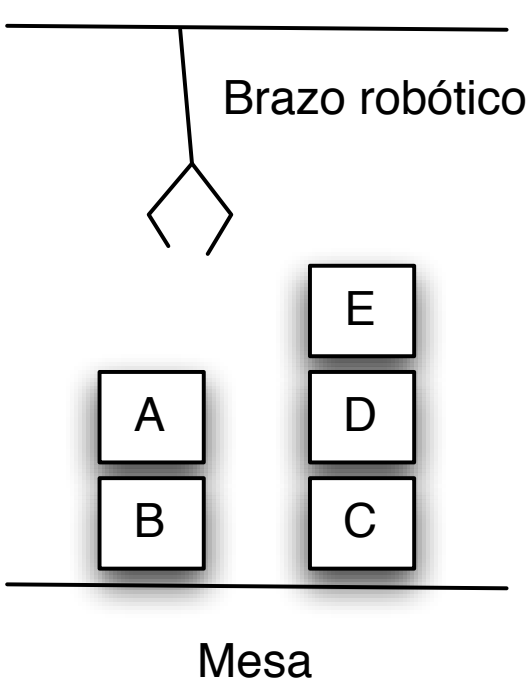
El mundo de los bloques [1]



- Necesitaremos símbolos para:
  - Objetos
  - Relaciones
  - Funciones
  - Variables

Universidad Veracruzana

Universo de discurso



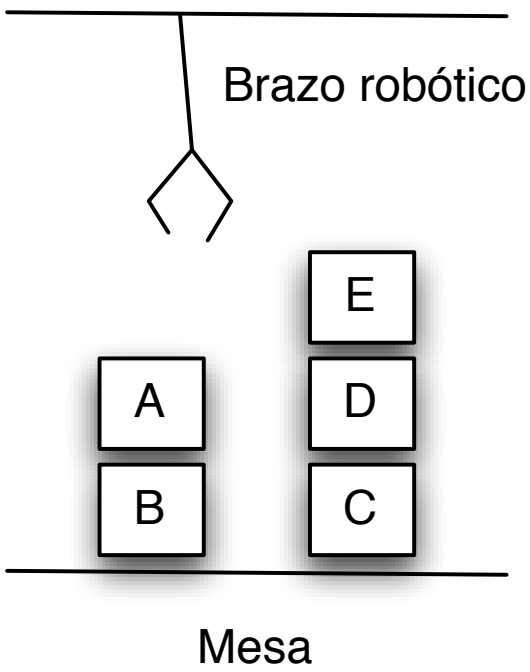
- Se denota como $\mathcal{U}$ y es el conjunto de todos objetos sobre los cuales queremos expresarnos.
- Para el mundo de los bloques:

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

- 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 $sombrero = \{(b, a), (c, d), (d, e)\}$:

$$sombrero(b) \mapsto a$$

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

Universidad Veracruzana

Predicados

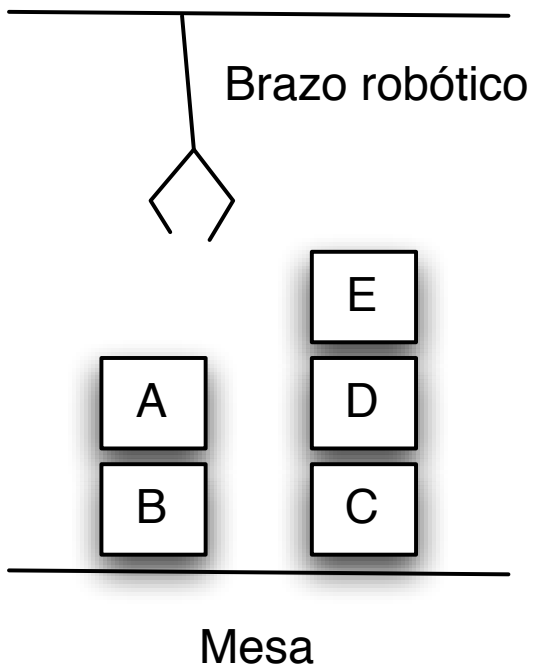
- Denotan relaciones entre los miembros de $\mathcal{U}$. Mapean a falso o verdadero.
- El predicado $sobre = \{(a, b), (d, c), (e, d)\}$:

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

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

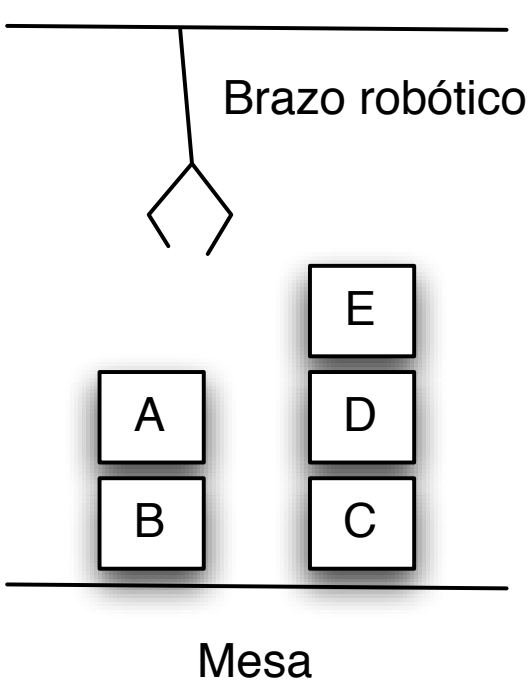
$$libre(d) \mapsto false$$

- 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 codomino en algún subconjunto de $\mathcal{U}$:

$$base(a) \mapsto b$$

Universidad Veracruzana

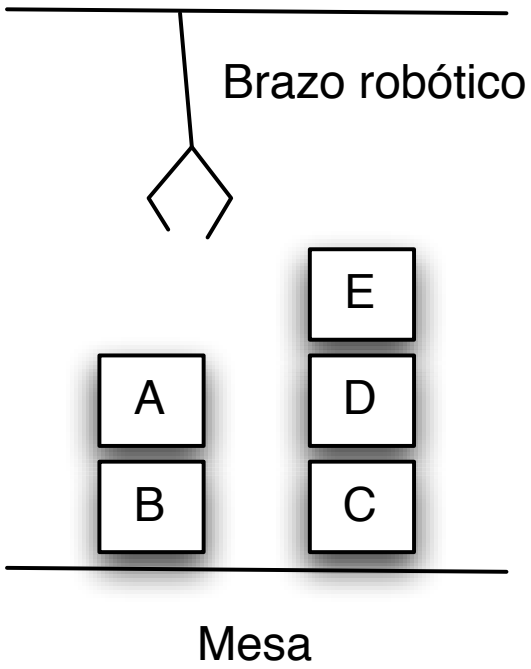
Variables y cuantificadores

- Toman valores de $\mathcal{U}$.
- El cuantificador universal ($\forall$) expresa hechos acerca de todos los objetos en $\mathcal{U}$, sin necesidad de enumerarlos:

$$\forall X \; \mathit{libre}(X)$$

- El cuantificador existencial ($\exists$) expresa la existencia de al menos un objeto en $\mathcal{U}$ con cierta propiedad en particular:

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



Universidad Veracruzana

Alfabeto de la Lógica de Primer Orden

Const	El conjunto de símbolos que denotan constantes;
Var	El conjunto de símbolos que denotan variables;
Pred	El conjunto de símbolos que denotan predicados;
Func	El conjunto de símbolos funcionales ($Const \subset Func$);
$\neg$	El operador unario de negación;
$\vee$	El operador binario de disyunción;
$\forall$	El símbolo de cuantificación universal;
()	Los paréntesis.

Universidad Veracruzana

Reglas sintácticas (términos)

1. Si $\alpha \in \mathit{Const}$ entonces $\alpha \in \mathit{Term}$;
   - Ejs. $a, b, c, \mathit{mesa}, \ldots$
2. Si $\alpha \in \mathit{Var}$, entonces $\alpha \in \mathit{Term}$;
   - Ejs. $X, Y, Z, \mathit{Cubo1}, \mathit{Cubo2}, \ldots$
3. Si $\alpha/n \in \mathit{Func}$, entonces $\alpha(\phi_1, \ldots, \phi_n) \in \mathit{Term}$ si y sólo si $\phi_{1 \leq i \leq n} \in \mathit{Term}$.
   - Ejs. $\mathit{base}(b), \mathit{sombrero}(X), \mathit{sombrero}(\mathit{base}(b)), \ldots$
4. Ninguna otra expresión es un término.

Universidad Veracruzana

Reglas sintácticas (fórmulas bien formadas) I

1. Si $\alpha \in \mathit{Pred}$ es de arirad n, entonces $\alpha(\phi_0, \ldots, \phi_n) \in \mathcal{L}_{FOL}$ si y sólo si $\phi_i \in \mathit{Term} \mid i = 0 \ldots n$;
   - Ejs. $\mathit{sobre}(a, b), \mathit{libre}(X), \mathit{libre}(\mathit{sombrero}(X)), \ldots$
2. Si $\alpha \in \mathcal{L}_{FOL}$, entonces $\neg\alpha \in \mathcal{L}_{FOL}$;
   - Ejs. $\neg\mathit{libre}(a), \neg\mathit{libre}(Y), \ldots$
3. Si $\alpha \in \mathcal{L}_{FOL}$ y $\beta \in \mathcal{L}_{FOL}$, entonces $(\alpha \vee \beta) \in \mathcal{L}_{FOL}$
   - Ejs. $\mathit{sobre}(X, Y) \vee \neg\mathit{libre}(X), \ldots$

Universidad Veracruzana

Reglas sintácticas (fórmulas bien formadas) II

4. Si $\alpha \in \mathcal{L}_{FOL}$ y $X \in \mathit{Vars}$ es una variable que ocurre en α, entonces $\forall X\ \alpha \in \mathcal{L}_{FOL}$
   - Ejs. $\forall X\ \mathit{libre}(X), \ldots$

5. Nada más es una fórmula bien formada (fbf).

Universidad Veracruzana

Definiciones auxiliares

Conjunción. $(\alpha \wedge \beta) =_{def} \neg(\neg\alpha \vee \neg\beta)$;

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

Equivalencia material. $(\alpha \leftrightarrow \beta) =_{def} ((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$;

Falso. $\mathsf{f} =_{def} \neg\alpha \wedge \alpha$;

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

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

Universidad Veracruzana

Términos BNF

- Un término se define como:

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

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

Universidad Veracruzana

Fórmulas bien formadas BNF

- Las fbf del lenguaje de la Lógica de Primer Orden se construye a partir de las variables *Var*, los functores *Func* y los predicados *Pred* como sigue:

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

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 $\forall X\ \alpha$, se dice que la fbf α está bajo el alcance del cuantificador $\forall X$.
- En tal caso, se dice que la ocurrencia de X en α está acotada, en caso contrario se dice que la ocurrencia de la variable es libre.
- Ejemplo: En $\exists X\ sobre(X, Y)$ la variable X está acotada, mientras que Y está libre.
- Un término o fbf sin variables se conoce como básico/a (*ground*).
- Ejemplo: $sobre(a, b)$.

Universidad Veracruzana

Interpretación

- Para expresar que al menos hay un bloque que no tiene nada encima, escribimos:

$$\exists X \; \mathit{bloque}(X) \wedge \mathit{libre}(X)$$

- Cuando usamos cuantificadores siempre tenemos en mente al $\mathcal{U}$, en este caso $\{a, b, c, d, e, \mathit{brazo}, \mathit{mesa}\}$.
- 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. En la escena usada $\{a, e\}$.

Universidad Veracruzana

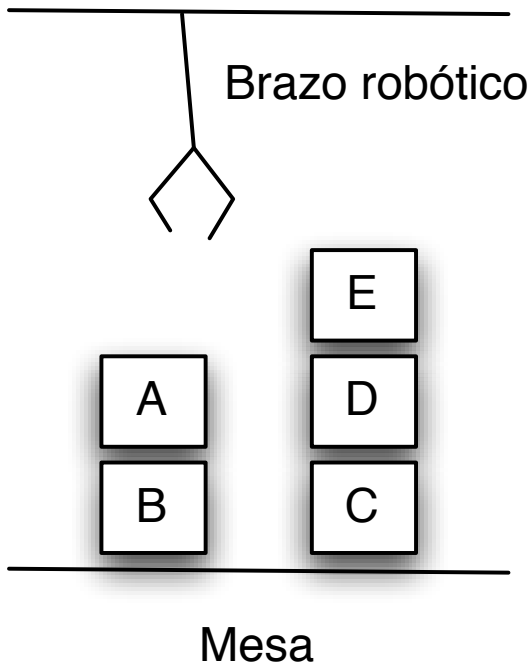
Teoría del modelo

- Para obtener un modelo para el lenguaje $\mathcal{L}_{FOL}$ definimos la tupla $M = \langle D, V \rangle$, donde D es el universo de discurso y V es una interpretación Con respecto a D.
- V satisface las siguientes propiedades:
  - Si $\alpha \in Const$, entonces $V(\alpha) = \alpha$;
  - Si $\alpha/n \in Pred$, tal que $n \geq 1$, entonces $V(\alpha) \subseteq D^n$.
- Algunas veces la expresión $V(\alpha)$ se abrevia α^V.
- Ejemplos:
  - $libre^V \subseteq D = \{a, b, c, d, e\}$.
  - $sobre^V \subseteq D \times D = \{(a, b), (e, d), (d, c)\}$.

Universidad Veracruzana

Interpretación para el mundo de bloques

$$
\begin{aligned}
a^V &= a \\
b^V &= b \\
c^V &= c \\
d^V &= d \\
e^V &= e \\
sobre^V &= \{(a,b),(e,d),(d,c)\} \\
enLaMesa^V &= \{b,c\} \\
libre^V &= \{a,e\} \\
porEncima^V &= \{(a,b),(e,d),(e,c),(d,c)\}
\end{aligned}
$$



Universidad Veracruzana

Asignación de variables y términos

- Decimos que U es una asignación de variables basada en el modelo $M = \langle D, V \rangle$ si para todo $\alpha \in \mathit{Var}$, $U(\alpha) \in D$.
- 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:
  - Si $\alpha \in \mathit{Const}$, entonces $T_{VU}(\alpha) = V(\alpha)$.
  - Si $\alpha \in \mathit{Var}$, entonces $T_{VU}(\alpha) = U(\alpha)$.
  - Si $\alpha \in \mathit{Term}$ y es de la forma $\alpha(\phi_1, \ldots, \phi_n)$; y $V(\alpha) = g$; y $T_{VU}(\phi_i) = x_i$, entonces $T_{VU}(\alpha(\phi_1, \ldots, \phi_n)) = g(x_1, \ldots, x_n)$.

Universidad Veracruzana

Satisfacción

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

$$\models_V \alpha[U]$$

- Entonces podemos escribir $M \models V_U(\alpha)$ para expresar que α es verdadera en el modelo $M = \langle D, V \rangle$ cuando las variables en α toman valores de acuerdo a la asignación U.
- Ejemplo:

$$M \models V_U(sobre(X, b)) \mid U = \{X \setminus a\}$$

Universidad Veracruzana

Satisfacción

- Dado un modelo $M = \langle D, V \rangle$ y una asignación de términos T_{VU}:

1. $\models_V (\phi = \psi)[U]$ si y sólo si $T_{VU}(\phi) = T_{VU}(\psi)$.
2. $\models_V \phi(\tau_1, \ldots, \tau_n)[U]$ si y sólo si $(T_{VU}(\tau_1), \ldots, T_{VU}(\tau_n)) \in \phi^V$.
3. $\models_V (\neg\phi)[U]$ si y sólo si $\not\models_V \phi[U]$.
4. $\models_V (\phi \wedge \psi)[U]$ si y sólo si $\models_V \phi[U]$ y $\models_V \psi[U]$.
5. $\models_V (\phi \vee \psi)[U]$ si y sólo si $\models_V \phi[U]$ o $\models_V \psi[U]$.
6. $\models_V (\phi \rightarrow \psi)[U]$ si y sólo si $\not\models_V \phi[U]$ o $\models_V \psi[U]$.
7. $\models_V (\forall\nu\ \phi)[U]$ si y sólo si para todo $d \in D$ es el caso que $\models_V \phi[W]$, donde $\nu^W = d$ y $\mu^W = \mu^U$ para $\mu \neq \nu$.
8. $\models_V (\exists\nu\ \phi)[U]$ si y sólo si para algún $d \in D$ es el caso que $\models_V \phi[W]$, donde $\nu^W = d$ y $\mu^W = \mu^U$ para $\mu \neq \nu$.

Universidad Veracruzana

Definiciones complementarias

- Si una interpretación V satisface a un enunciado α para toda asignación de variables, se dice que V es un modelo de α.
- Un enunciado se dice satisfacible si existe alguna interpretación y asignación de variables que lo satisfaga.
- De otra forma, se dice que el enunciado es insatisfacible.
- Se dice que una fbf α 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: $p(X) \vee \neg p(X)$ 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. $\forall X \; \forall Y \;\; madre(X) \wedge hijo\_de(Y, X) \rightarrow ama(X, Y)$
  2. $madre(julia) \wedge hijo\_de(luis, julia)$
  3. $ama(julia, luis)$

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.
- Ejs. Las reglas de inferencia incluyen:

Modus Ponens:

$$\frac{\alpha \quad \alpha \rightarrow \beta}{\beta} \quad (\rightarrow E)$$

Eliminación de cuantificador universal:

$$\frac{\forall X \ \alpha(X)}{\alpha(t)} \quad (\forall E \ )$$

Introducción de conjunción:

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta} \quad (\wedge I)$$

Universidad Veracruzana

Ejemplo de derivación

- Inicio:
  1. $\forall X\ \forall Y\ madre(X) \wedge hijo\_de(Y, X) \rightarrow ama(X, Y)$
  2. $madre(julia) \wedge hijo\_de(luis, julia)$
- Al aplicar la eliminación de cuantificador universal ($\forall E$) a (1) obtenemos:
  3. $\forall Y\ (madre(julia) \wedge hijo\_de(Y, julia) \rightarrow ama(julia, Y)$
- Al aplicar nuevamente ($\forall E$) a (3) obtenemos:
  4. $madre(julia) \wedge hijo\_de(luis, julia) \rightarrow ama(julia, luis)$
- Finalmente, al aplicar Modus Ponens a (2) y (4):
  5. $ama(julia, luis)$

Universidad Veracruzana

Correctez y completitud

- Un conjunto de reglas de inferencia se dice correcto (*sound*), si para todo conjunto de fbf cerradas (sin ocurrencia de variables libres) Δ y cada fbf cerrada α, siempre que $\Delta \vdash \alpha$ se tiene que $\Delta \models \alpha$.
- Un conjunto de reglas de inferencia se dicen completo si siempre que $\Delta \models \alpha$ entonces $\Delta \vdash \alpha$.

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:
  - $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$ ó
  - $\forall X\alpha \equiv \neg\exists X\neg\alpha$.

Universidad Veracruzana

Literales

- Un átomo o fórmula atómica es un predicado de aridad n aplicado a n términos.
- 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.
- Ejemplo. Asumiendo que *hijo_de* es un predicado de aridad 2:
  - $hijo\_de(juan, marcos)$.
  - $\neg hijo\_de(juan, alicia)$.
- Las literales son los bloques de construcción α_i en:

$$\alpha_0 \leftarrow \alpha_1 \wedge \cdots \wedge \alpha_n \quad (n \geq 0)$$

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.

$$\alpha_0 \vee \neg\alpha_1 \vee \cdots \vee \neg\alpha_n \quad (n \geq 0)$$

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

$$\alpha_0 \leftarrow \alpha_1 \wedge \cdots \wedge \alpha_n \quad (n \geq 0)$$

- La cláusula vacía (sin literales) se denota como $\square$ y es equivalente a $\square \leftarrow \blacksquare$ (en otra notación alternativa $\bot \leftarrow \top$).

Universidad Veracruzana

Hechos y reglas revisitados

- Para $\alpha_0 \leftarrow \alpha_1 \wedge \ldots \wedge \alpha_n \quad (n \geq 0)$, tenemos que:
  - Si $n = 0$, la literal α_0 será positiva, por lo que la cláusula definitiva será un hecho, si α_0 es de base.
  - Si $n > 0$ la cláusula definitiva toma la forma de una regla, donde α_0 es la cabeza de la regla; y la conjunción $\alpha_1 \wedge \ldots \wedge \alpha_n$ su cuerpo.
- Las reglas tienen 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.

$$\leftarrow \alpha_1 \wedge \ldots \wedge \alpha_n \quad (n \geq 1)$$

Universidad Veracruzana

Ejemplo

- Considere las siguientes consultas:

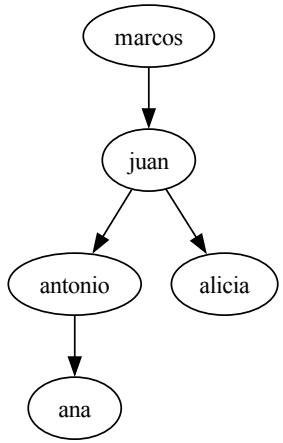
Consulta	Meta definitiva
¿Es Ana hija de Antonio?	$\leftarrow hijo(ana, antonio)$
¿Quién es nieto de Ana?	$\leftarrow nieto(X, ana)$
¿De quién es nieto Antonio?	$\leftarrow nieto(antonio, X)$
¿Quién es nieto de quién?	$\leftarrow nieto(X, Y)$

Universidad Veracruzana

Respuestas obtenidas

Programa:

1. $hijo\_de(antonio, juan)$
2. $hijo\_de(ana, antonio)$
3. $hijo\_de(juan, marcos)$
4. $hijo\_de(alicia, juan)$
5. $nieto\_de(X, Y) \leftarrow hijo\_de(Z, Y) \wedge hijo\_de(X, Z))$



Metas:

a. $\leftarrow hijo\_de(ana, antonio)$

b. $\leftarrow nieto\_de(X, ana)$

c. $\leftarrow nieto\_de(antonio, X)$

d. $\leftarrow nieto\_de(X, Y)$

Respuestas:

a. $true$.

b. $false$.

c. $X = marcos$.

d. $X = ana\ Y = juan$,
$X = antonio\ Y = marcos$,
$X = alicia\ Y = marcos$.

Universidad Veracruzana

Cláusulas de Horn

- Una cláusula de Horn es una cláusula ó una meta definitivas.
- La cláusula vacía $\square$, que evalúa siempre a *false*, es una cláusula de Horn.
- Las cláusulas de Horn implican restricciones expresivas: 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 recordando que las variables que ocurren en una meta, están cuantificadas existencialmente:

$$\neg\exists X_1 \ldots X_m \ (\alpha_1 \ \wedge \ldots \wedge \ \alpha_n)$$

- Prolog tratará de encontrar términos $t_i \ldots t_m$ tales que al remplazar X_i por t_i $(1 \leq i \leq m)$ en la fbf $(\alpha_1 \wedge \cdots \wedge \alpha_n)$, ésta se satisface.
- Una demostración por contra ejemplo.

Universidad Veracruzana

Conocimiento positivo

- Los programas definitivos solo pueden expresar conocimiento positivo.
- Hechos y reglas nos dicen que objetos del universo de discurso están en una relación, pero no nos dicen cuales no.
- Al usar programas definitivos, no es posible construir descripciones contradictorias, i.e., conjuntos de fbf no satisfacibles.
- Todo programa definitivo tiene un modelo.
- ¿Cual es ese modelo?

Universidad Veracruzana

Modelos e interpretaciones (recordatorio)

- Sea α una fbf y V una intepretación. V es un modelo de α si $\models_V \alpha$.
- Ej. Dada la interpretación V del mundo de los cubos:

$$\models_V sobre(a, b)$$

  por lo que V es un modelo de $sobre(a, b)$.
- Sea Δ un conjunto finito de fbf y V una interpretación. V es un modelo de Δ si $\models_V \alpha$ para toda $\alpha \in \Delta$.
- Una clase interesante de interpretaciones 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_L) es el conjunto de todos los términos formados con las constantes y functores de L.
- La Base de Herbrand (B_L) es el conjunto de todos los átomos que pueden formarse con los predicados y los términos en U_L.

Universidad Veracruzana

Ejemplo: impar

- Sea $\Delta = \{impar(s(0)), impar(s(s(X))) \leftarrow impar(X)\}$. Entonces:
  - $U_L = \{0, s(0), s(s(0)), s(s(s(0))), \ldots\}$
  - $B_L = \{impar(0), impar(s(0)), impar(s(s(0))), \ldots\}$

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_L.
  - Para cada constante $c \in L$, $c^V = c$.
  - Para cada functor $f/n \in L$, $f^V : U_L^n \mapsto U_L$.
  - Para cada predicado $p/n \in L$, $p^V \subseteq U_L^n$.

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

- Sea $\Delta = \{impar(s(0)), impar(s(s(X))) \leftarrow impar(X)\}$.
- Una posible modelo de este programa es

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

Universidad Veracruzana

Resultados de interés

- Una interpretación de Herbrand V es un modelo de $\forall\alpha$, si α se satisface para todas las posibles asignaciones de las variables en α.
- Para las cláusulas definitivas, $\Delta \models \alpha$ si y sólo si todo modelo de Herbrand de Δ es también un modelo de α (donde α es una literal positiva).

Universidad Veracruzana

Programa extendido con meta

- 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 $V = \{\alpha \in B_L \mid \models_{V'} \alpha\}$ es un modelo de Herbrand de $\Delta \cup \{\gamma\}$.

Universidad Veracruzana

Consistencia

- Sea Δ un programa definitivo y α una cláusula definitiva.
- Sea $\Delta' = \Delta \cup \{\neg\alpha\}$.
- Entonces $\Delta \models \alpha$ 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. $\{masculino(adan), femenino(eva), masculino(eva)\}$
  3. $\{masculino(adan), femenino(eva), femenino(adan)\}$
  4. $\{masculino(adan), femenino(eva), masculino(eva), femenino(adan)\}$
- El modelo mínimo es el único que corresponde con el modelo pretendido del programa, pero esto no siempre es el caso.
- ¿Cuando lo es?

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:

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

- La prueba (Ver Nilsson y Maluszynski [3], pág. 29) pasa por demostrar que $M_\Delta \supseteq \{\alpha \in B_L | \Delta \models \alpha\}$ y $M_\Delta \subseteq \{\alpha \in B_\Delta | \Delta \models \alpha\}$.

Universidad Veracruzana

Operador de consecuencia inmediata

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

$$T_\Delta(V) = \{\alpha_0 \mid \alpha_0 \leftarrow \alpha_1, \ldots, \alpha_n \in base(\Delta) \wedge \{\alpha_1, \ldots, \alpha_n\} \subseteq V\}$$

- Se puede demostrar que existe un punto fijo $T_\Delta(V) = V$ y que V es idéntica al modelo mínimo de Herbrand de Δ.

Universidad Veracruzana

Notación

- Existe una notación estándar para denotar a los miembros de esta secuencia de interpretaciones construidas 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 construido de esta manera es idéntico al modelo mínimo de Herbrand de Δ.

Universidad Veracruzana

Ejemplo

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

$$\begin{aligned} T_\Delta \uparrow 0 &= \emptyset \\ T_\Delta \uparrow 1 &= \{impar(s(0))\} \\ T_\Delta \uparrow 2 &= \{impar(s(0)), impar(s(s(s(0))))\} \\ &\vdots \\ T_\Delta \uparrow m &= \{impar(s^n(0)) \mid n \in \{1, 3, 5, \dots\}\} \end{aligned}$$

Universidad Veracruzana

Programas y metas definitivos

- Consideren el siguiente programa definitivo Δ:

$$
\begin{aligned}
papa(juan, marta).& \\
recien\_nacido(marta).& \\
orgulloso(X) &\leftarrow padre(X, Y), recien\_nacido(Y). \\
padre(X, Y) &\leftarrow papa(X, Y). \\
padre(X, Y) &\leftarrow mama(X, Y).
\end{aligned}
$$

- ¿Cómo interpretamos la siguiente meta definitiva?

$$\leftarrow orgulloso(Z).$$

Universidad Veracruzana

Metas

- La meta $\leftarrow orgulloso(Z)$. es una abreviatura de $\forall Z \neg orgulloso(Z)$
- Que a su vez es equivalente a $\neg \exists Z\ orgulloso(Z)$.
- Si ese enunciado es contradictorio en Δ, entonces sabemos que $\Delta \models \exists Z\ orgulloso(Z)$. Eso respondería *true*, a la meta.
- La otra parte de la respuesta es encontrar una substitución θ (asignación de variables) t.q. el conjunto $\Delta \cup \{\neg orgulloso(Z)\theta\}$ sea insatisfacible, i.e., $\Delta \models orgulloso(Z)\theta$.
- Ejemplo: $\Delta \cup \neg orgulloso(Z)\{Z/juan\}$

Universidad Veracruzana

Razonamiento

- Asumimos la meta G_0 – Para todo Z, Z no está orgulloso.
- La inspección del programa Δ revela que una regla describe una condición para que alguien esté orgulloso:

$$orgulloso(X) \leftarrow padre(X, Y), recien\_nacido(Y).$$

- Lo cual es lógicamente equivalente a:

$$\forall(\neg orgulloso(X) \rightarrow \neg(padre(X, Y) \wedge recien\_nacido(Y)))$$

Universidad Veracruzana

Estrategia

- Partiendo de:

$$\forall(\neg orgulloso(X) \rightarrow \neg(padre(X, Y) \wedge recien\_nacido(Y)))$$

- Al renombrar X por Z, eliminar el cuantificador universal y usar *modus ponens* con respecto a G_0, obtenemos G_1:

$$\neg(padre(Z, Y) \wedge recien\_nacido(Y))$$

- o su equivalente:

$$\leftarrow padre(Z, Y), recien\_nacido(Y).$$

- G_1 que es verdadera en todo modelo $\Delta \cup \{G_0\}$.

Universidad Veracruzana

Resolución

- Ahora solo queda probar que $\Delta \cup \{G_1\}$ es no satisfacible. Observen que G_1 es equivalente a la fbf:

$$\forall Z \forall Y (\neg padre(Z, Y) \vee \neg recien\_nacido(Y))$$

- G_1 no es satisfacible en Δ, si en todo modelo de Δ hay una persona que es padre de un recién nacido.
- Revisemos la primer condición:

$$padre(X, Y) \leftarrow papa(X, Y).$$

- Por lo que G_1 se reduce a G_2:

$$\leftarrow papa(Z, Y), recien\_nacido(Y).$$

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:

$$\leftarrow recien\_nacido(marta).$$

- pero el programa contiene el hecho:

$$recien\_nacido(marta).$$

- equivalente a $\neg recien\_nacido(marta) \rightarrow \square$
- 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

Regla de inferencia

- La resolución puede resumirse con la siguiente regla de inferencia:

$$\frac{\forall\neg(\alpha_1 \wedge \cdots \wedge \alpha_{i-1} \wedge \alpha_i \wedge \alpha_{i+1} \wedge \cdots \wedge \alpha_m) \quad \forall(\beta_0 \leftarrow \beta_1 \wedge \cdots \wedge \beta_n)}{\forall\neg(\alpha_1 \wedge \cdots \wedge \alpha_{i-1} \wedge \beta_1 \wedge \cdots \wedge \beta_n \wedge \alpha_{i+1} \wedge \cdots \wedge \alpha_m)\theta}$$

- donde:
  1. $\alpha_1, \ldots, \alpha_m$ son fbf atómicas.
  2. $\beta_0 \leftarrow \beta_1, \ldots, \beta_n$ es una cláusula definitiva en el programa Δ ($n \geq 0$).
  3. $MGU(\alpha_i, \beta_0) = \theta$.

Universidad Veracruzana

Observaciones

- La regla tiene dos premisas: una meta y una cláusula definitivas.
- Solo hay un cuantificador universal para la conclusión. Por lo tanto, 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

- Una meta definitiva puede incluir varias fbf atómicas que unifican con la cabeza de alguna cláusula en el programa.
- En este caso, es deseable contar con un mecanismo determinista para seleccionar un átomo α_i a unificar.
- Se asume una función de selección para las sub-metas de la meta definitivas.
- Ejemplo. De izquierda a derecha.

Universidad Veracruzana

Usando la resolución-SLD

- El punto de partida es una meta definitiva G_0:

$$\leftarrow \alpha_1, \ldots, \alpha_m \quad (m \geq 0)$$

- La sub-meta α_1 será seleccionada.
- Una nueva meta G_1 se construye con una cláusula del programa $\beta_0 \leftarrow \beta_1, \ldots, \beta_n$ $(n \geq 0)$ tal que $\theta_1 = MGU(\alpha_1, \beta_0)$.
- G_1 tiene la forma:

$$\leftarrow (\beta_1, \ldots, \beta_n, \alpha_2, \ldots, \alpha_m)\theta_1$$

- Ahora es posible aplicar el principio de resolución a G_1 para obtener G_2, y así sucesivamente.

Universidad Veracruzana

Terminación

- El proceso puede terminar o no.
- Hay dos situaciones donde no es posible obtener G_{i+1} a partir de G_i:
  1. Cuando la sub-meta seleccionada no puede ser resuelta (no unifica con la cabeza de alguna cláusula del programa).
  2. Cuando $G_i = \square$.
- En cualquier otro caso la resolución-SLD se sigue aplicando.

Universidad Veracruzana

Derivación-SLD

- Sea G_0 una meta definitiva, Δ un programa definitivo y $\mathcal{R}$ una función de selección. Una derivación SLD de G_0 (usando Δ y $\mathcal{R}$) es una secuencia finita o infinita de metas:

$$G_0 \overset{\alpha_0}{\rightsquigarrow} G_1 \ldots G_{n-1} \overset{\alpha_{n-1}}{\rightsquigarrow} G_n$$

- $\alpha_i \in \Delta$. Las variables en α_i se renombran con subíndices i.

Universidad Veracruzana

Substitución computada

- Cada derivación SLD nos lleva a una secuencias de MGUs $\theta_1, \ldots, \theta_n$. La composición

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

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

Universidad Veracruzana

Ejemplo

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

$$\begin{aligned} G_0 &= \leftarrow orgulloso(Z). \\ \alpha_0 &= orgulloso(X_0) \leftarrow padre(X_0, Y_0), recien\_nacido(Y_0). \end{aligned}$$

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

$$\begin{aligned} G_1 &= \leftarrow padre(Z, Y_0), recien\_nacido(Y_0). \\ \alpha_1 &= padre(X_1, Y_1) \leftarrow papa(X_1, Y_1). \end{aligned}$$

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

Universidad Veracruzana

Ejemplo

- La derivación continua como sigue:

$$
\begin{aligned}
G_2 &= \leftarrow papa(Z, Y_0), recien\_nacido(Y_0). \\
\alpha_2 &= papa(juan, marta). \\
G_3 &= \leftarrow recien\_nacido(marta). \\
\alpha_3 &= recien\_nacido(marta). \\
G_4 &= \square
\end{aligned}
$$

- la substitución computada para esta derivación es:

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

Universidad Veracruzana

Refutación-SLD y derivación fallida

- Una derivación SLD finita:

$$G_0 \overset{\alpha_0}{\rightsquigarrow} G_1 \ldots G_{n-1} \overset{\alpha_{n-1}}{\rightsquigarrow} G_n$$

  donde $G_n = \square$, se llama refutación SLD de G_0.
- Una derivación de la meta G_0 cuyo último elemento no es la meta vacía y no puede resolverse con ninguna cláusula del programa, es llamada derivación fallida.

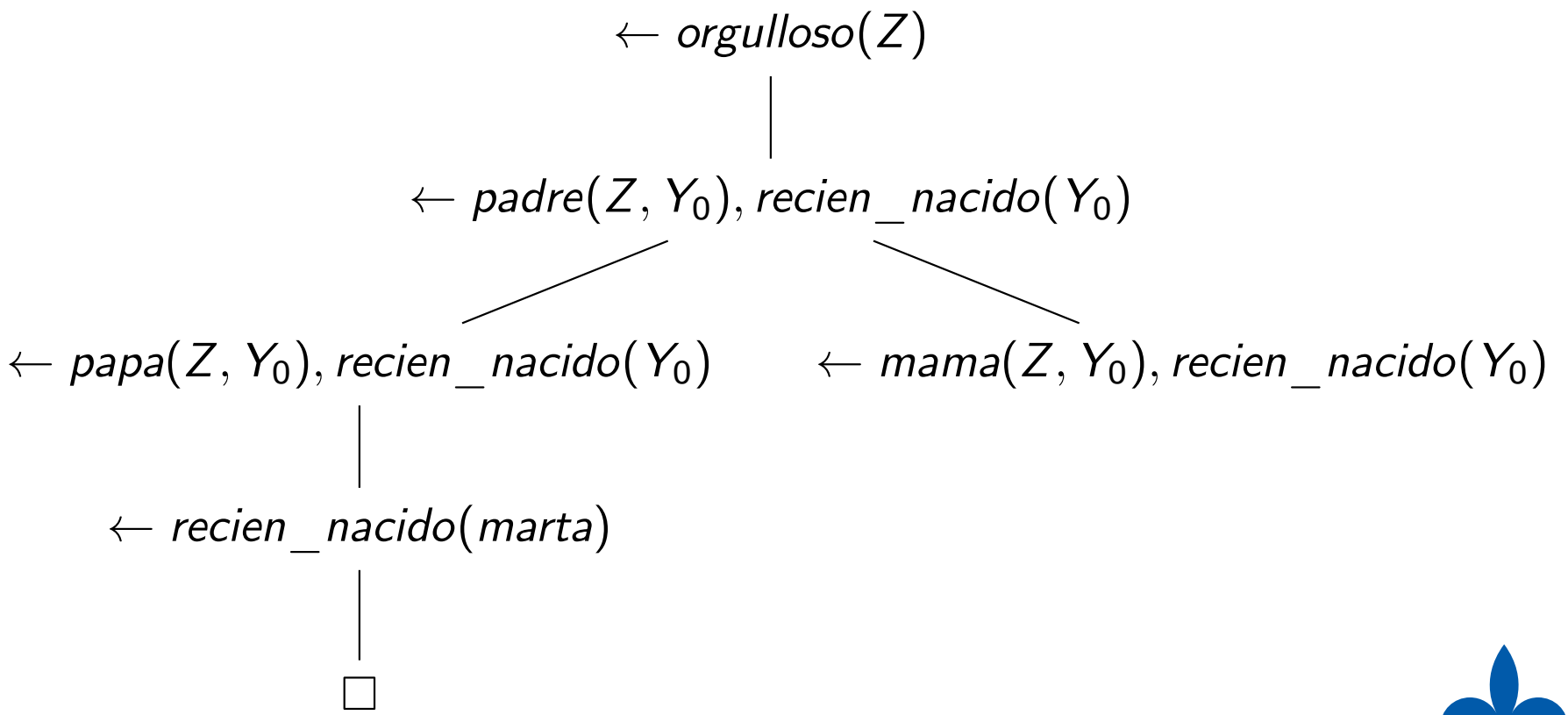
Universidad Veracruzana

Arbol-SLD

- Sea Δ un programa definitivo, G_0 una meta definitiva, y $\mathcal{R}$ una función de selección.
- El árbol-SLD de G_0 (usando Δ y $\mathcal{R}$) es un árbol etiquetado, posiblemente infinito, que cumple con:
  - La raíz del árbol está etiquetada por G_0.
  - Si el árbol contiene un nodo etiquetado como G_i y existe una cláusula renombrada $\alpha_i \in \Delta$ tal que G_{i+1} es dervidada de G_i y α_i via $\mathcal{R}$, entonces el nodo etiquetado como G_i tiene un hijo etiquetado G_{i+1} El arco entre ambos es α_i.

Universidad Veracruzana

Ejemplo



Universidad Veracruzana

Observaciones

- Esta computación no necesariamente es determinista si $\mathcal{R}$ es justa, i.e., cualquier átomo de la meta y cláusula del programa pueden ser seleccionados.
- Pueden existir unificadores alternativos para dos átomos.
- Por otra parte, es posible también que el átomo seleccionado no unifique con ninguna cláusula en el programa.
- La computación puede caer en un ciclo y de esta manera no producir solución alguna.

Universidad Veracruzana

Propiedades de la resolución-SLD

Correctez. Sea Δ un programa definitivo, $\mathcal{R}$ una función de selección, y θ una substitución de respuesta computada a partir de Δ y $\mathcal{R}$ para una meta $\leftarrow \alpha_1, \ldots, \alpha_m$. Entonces $\forall((\alpha_1 \wedge \cdots \wedge \alpha_m)\theta)$ es consecuencia lógica del programa Δ.

Completez. Sea Δ un programa definitivo, $\mathcal{R}$ una función de selección y $\leftarrow \alpha_1, \ldots, \alpha_m$ una meta definitiva. Si $\Delta \models \forall((\alpha_1 \wedge \cdots \wedge \alpha_m)\sigma)$, entonces existe una refutación de $\leftarrow \alpha_1, \ldots, \alpha_m$ vía $\mathcal{R}$ con una substitución de respuesta computada θ, tal que $(\alpha_1 \wedge \cdots \wedge \alpha_m)\sigma$ es un caso de $(\alpha_1 \wedge \cdots \wedge \alpha_m)\theta$.

Universidad Veracruzana

Substitución

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

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

  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 (*grounded*) 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 $\theta = \{X_1/t_1, \ldots, X_n/t_n\}$ una substitución y α una expresión. Entonces $\alpha\theta$, el caso (*instance*) de α bajo θ, es la expresión obtenida al substituir simultáneamente X_i por t_i para $1 \leq i \leq n$.
- Si $\alpha\theta$ es una expresión de base, se dice que es un caso base y se dice que θ es una substitución de base para α.
- Si $\Sigma = \{\alpha_1, \ldots, \alpha_n\}$ es un conjunto finito de expresiones, entonces $\Sigma\theta$ denota $\{\alpha_1\theta, \ldots, \alpha_n\theta\}$.

Universidad Veracruzana

Ejemplo

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

Universidad Veracruzana

Composición de substituciones

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

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

- 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 $Y_j \in \{X_1, \ldots, X_m\}$, 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

Propiedades de las substituciones

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

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 α y β términos. Una substitución θ tal que $\alpha\theta = \beta\theta$ es llamada unificador de α y β.
- $unifica(conoce(juan, X), conoce(Y, Z)) = \{Y/juan, X/Z\}$

Universidad Veracruzana

Unificador más general (MGU)

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

Universidad Veracruzana

Forma resuelta y MGU

- Un conjunto de ecuaciones $\{X_1 = t_1, \ldots, X_n = t_n\}$ está en forma resuelta, si y sólo si $X_1, \ldots, X_n$ son variables distintas que no ocurren en $t_1, \ldots, t_n$.
- Para computar el MGU de dos términos α y β, primero intente transformar la ecuación $\{\alpha = \beta\}$ en una forma resuelta equivalente.
- Si esto falla, entonces $mgu(\alpha, \beta) = fallo$.
- Si una forma resuelta $\{X_1 = t_1, \ldots, X_n = t_n\}$ existe, entonces $mgu(\alpha, \beta) = \{X_1/t_1, \ldots, X_n/t_n\}$.

Universidad Veracruzana

Algoritmo de unificación

```
 1: function UNIFICA(E)                                   ▷ E es un conjunto de ecuaciones
 2:     repeat
 3:         (s = t) ← seleccionar(E)
 4:         if f(s_1, ..., s_n) = f(t_1, ..., t_n) (n ≥ 0) then
 5:             remplazar (s = t) por s_1 = t_1, ..., s_n = t_n
 6:         else if f(s_1, ..., s_m) = g(t_1, ..., t_n) (f/m ≠ g/n) then
 7:             return(fallo)
 8:         else if X = X then
 9:             remover la X = X
10:         else if t = X then
11:             remplazar t = X por X = t
12:         else if X = t then
13:             if subtermino(X,t) then
14:                 return(fallo)
15:             else remplazar todo X por t
16:             end if
17:         end if
18:     until No hay acción posible para E
19: end function
```

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:

$$
\begin{aligned}
\rightarrow \quad & \{X = a, g(X) = g(Z), b = Z\} \\
\rightarrow \quad & \{X = a, g(a) = g(Z), b = Z\} \\
\rightarrow \quad & \{X = a, a = Z, b = Z\} \\
\rightarrow \quad & \{X = a, Z = a, b = Z\} \\
\rightarrow \quad & \{X = a, Z = a, b = a\} \\
\rightarrow \quad & \textit{fallo}
\end{aligned}
$$

Universidad Veracruzana

Consideraciones

- Este algoritmo termina y regresa una forma resuelta equivalente al conjunto de ecuaciones de su entrada, o bien regresa fallo si la forma resuelta no existe.
- Sin embargo, el computar *subtermino*(X, t) (verificación de ocurrencia) hace que el algoritmo sea altamente ineficiente.
- El estándar ISO Prolog (1995) declara que el resultado de la unificación es no decidible.
- Al eliminar la verificación de ocurrencia es posible que al intentar resolver $X = f(X)$ obtengamos $X = f(f(X)) \cdots = f(f(f \ldots))$.

Universidad Veracruzana

Referencias I

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

[2] RA Kowalski y D Kuehner. "Linear Resolution with Selection Function". En: *Artificial Intelligence* 2.3/4 (1971), págs. 227-260.

[3] U Nilsson y J Maluszynski. *Logic, Programming and Prolog*. 2nd. John Wiley & Sons Ltd, 2000.

[4] JA Robinson. "A Machine-Oriented Logic based on the Resolution Principle". En: *Journal of the ACM* 12.1 (1965), págs. 23-41.

[5] SJ Russell y P Norvig. *Artificial Intelligence: A Modern Approach*. Third. Prentice Hall Series in Artificial Intelligence. USA: Prentice Hall, 2009.

Universidad Veracruzana