Representación del Conocimiento

Lógica de Primer Orden

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/rc

Maestría en Inteligencia Artificial 2025

Universidad Veracruzana

Índice

Universidad Veracruzana

Objetivo

- El uso de la lógica de primer orden para representar y resolver problemas, nos es familiar por la programación lógica que abordamos en el curso de Programación para la IA.
- Recuerden que usamos una lógica de primer orden restringida, *i.e.*, cláusulas de Horn (cláusulas y metas definitivas) y resolución-SLD: Prolog.
- Repasaremos estos conceptos y haremos algunas anotaciones sobre su uso en representación del conocimiento.

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 [2], 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:

$$\frac{\phi, \phi \implies \psi}{\psi} \; (\implies e)$$

Universidad Veracruzana

Componentes de un Sistema Formal

$$
\text{Sistema Formal}
\begin{cases}
\text{Sintáxis} \begin{cases} \text{Alfabeto} \\ \text{Reglas sintácticas} \end{cases} \\
\text{Semantica} \begin{cases} \text{Teoría del Modelo} \begin{cases} \text{Universo de Discurso} \\ \text{Intepretación} \end{cases} \\ \text{Teoría de Prueba} \begin{cases} \text{Axiomas} \\ \text{Reglas de Inferencia} \end{cases} \end{cases}
\end{cases}
$$

Universidad Veracruzana

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 ($Const \subset Func$);
- $\neg$ El operador monario de negación;
- $\vee$ El operador binario de disyunción;
- $\forall$ El símbolo de cuantificación universal;
- () Los paréntesis.

Universidad Veracruzana

Reglas sintácticas (términos)

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

- Ejemplos:
  1. $a, b, c, \mathit{mesa}, \dots$
  2. $X, Y, Z, \mathit{Cubo}1, \mathit{Cubo}2, \dots$
  3. $\mathit{base}(b), \mathit{sombrero}(X), \dots$

Universidad Veracruzana

Reglas sintácticas (fórmulas bien formadas)

1. Si $\phi/n \in Pred$, entonces $\phi(\phi_0, \ldots, \phi_n) \in \mathcal{L}_{FOL}$ si y sólo si $\phi_i \in Term, i = 0 \ldots n$;
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. $sobre(a, b), libre(X), ama(X, hijo(X)), \ldots$
  2. $\neg libre(a), \neg libre(Y), \ldots$
  3. $sobre(X, Y) \wedge \neg libre(X), \ldots$
  4. $\forall X \ ama(X, hijo(X)), \ldots$

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

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

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

Universidad Veracruzana

Términos en notación BNF

- Un término se define como:

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

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

Universidad Veracruzana

Fórmulas bien formadas en BNF

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

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

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

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 $\exists X \ sobre(X, Y)$ la variable X está acotada, mientras que Y está libre.
- Un término sin variables se conoce como término de base.
- Ejemplo. $sobre(a, b)$.

Universidad Veracruzana

Interpretación

- Para expresar que al menos hay un bloque que no tiene nada encima, escribimos: $\exists X \; bloque(X) \wedge libre(X)$.
- Cuando usamos cuantificadores siempre tenemos en mente al $\mathcal{U}$, en este caso $\{a, b, c, d, e, brazo, mesa\}$.
- 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

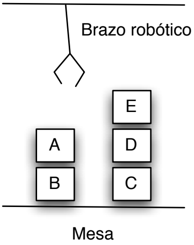
Teoría del modelo

- 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 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. $sobre^V \subset D \times D$.
- Algunas veces la expresión $V(\phi)$ se abrevia ϕ^V.

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 $\phi \in Var$, $U(\phi) \in D$.
- 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:
  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 $\psi(\phi_1, \ldots, \phi_n)$; y $V(\psi) = g$; y $T_{VU}(\phi_i) = x_i$, entonces $T_{VU}(\psi(\phi_1, \ldots, \phi_n)) = g(x_1, \ldots, x_n)$.

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. $\models_V (\forall\nu\ \phi)[U]$ ssi 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. $\models_V (\exists\nu\ \phi)[U]$ ssi para algún $d \in D$ es el caso que $\models_V \phi[W]$, donde $\nu^W = d$ y $\mu^W = \mu^U$ para $\mu \neq \nu$.
- A las asignaciones de variables como W, se les conoce como ν-alternativas.

Universidad Veracruzana

Definiciones complementarias

- Si una interpretación V safisface 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 intepretación 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. $\forall X\ \forall Y\ madre(X) \wedge hijo\_de(Y, X) \rightarrow ama(X, Y)$
    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.
  - Modus Ponens. O eliminación de la implicación:

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

  - 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. $\forall X \forall Y madre(X) \wedge hijo\_de(Y, X) \implies ama(X, Y)$
  2. $madre(julia) \wedge hijo\_de(luis, julia)$
- Al aplicar la eliminación de cuantificador universal ($\forall E$) a (1) obtenemos:
  3. $\forall Y(madre(julia) \wedge hijo\_de(Y, julia) \implies ama(julia, Y)$
- Al aplicar nuevamente ($\forall E$) a (3) obtenemos:
  4. $madre(julia) \wedge hijo\_de(luis, julia) \implies ama(julia, luis)$
- 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, 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 cuando $\Delta \models \phi$ siempre es el caso que $\Delta \vdash \phi$.

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. $\forall X \forall Y (nieto\_de(X, Y) \leftarrow \exists Z (hijo\_de(Z, Y) \land hijo\_de(X, Z)))$

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:

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

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.

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

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

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

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

Universidad Veracruzana

Hechos y reglas revisitados

- En $\phi_0 \leftarrow \phi_1 \wedge \cdots \wedge \phi_n \quad (n \geq 0)$, 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 $\phi_1 \wedge \cdots \wedge \phi_n$ 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.

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

Universidad Veracruzana

Ejemplos

- 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

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.
- 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 intepretación. 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 functores 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 $\Delta = \{impar(s(0)), impar(s(s(X))) \leftarrow impar(X)\}$.
- $U_\Delta = \{0, s(0), s(s(0)), s(s(s(0))), \dots\}$
- $B_\Delta = \{impar(0), impar(s(0)), impar(s(s(0))), \dots\}$

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

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

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

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

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 meta $\leftarrow orgulloso(Z).$?

Universidad Veracruzana

Metas

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

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:

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

- 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 γ_0, obtenemos γ_1:

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

- o su equivalente:

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

- 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:

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

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

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

- Por lo que γ_1 se reduce a γ_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

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

Propiedades

- 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 I

- 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 atomo 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

Observaciones II

- 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 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

Formalizando

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

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

- donde:
  1. $\phi_1, \ldots, \phi_m$ son fbf atómicas.
  2. $\psi_0 \leftarrow \psi_1, \ldots, \psi_n$ 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:

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

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

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

- 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:

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

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

Universidad Veracruzana

Substitución computada

- Cada derivación SLD nos lleva a una 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

$$\begin{aligned} \gamma_0 &= \leftarrow orgulloso(Z). \\ \phi_0 &= orgulloso(X_0) \leftarrow padre(X_0, Y_0), recien\_nacido(Y_0). \end{aligned}$$

- 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:

$$\begin{aligned} \gamma_1 &= \leftarrow padre(Z, Y_0), recien\_nacido(Y_0). \\ \phi_1 &= padre(X_1, Y_1) \leftarrow papa(X_1, Y_1). \end{aligned}$$

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

Universidad Veracruzana

Ejemplo

- La derivación continua como sigue:

$$
\begin{aligned}
\gamma_2 &= \leftarrow papa(Z, Y_0), recien\_nacido(Y_0).\\
\phi_2 &= papa(juan, marta).\\
\gamma_3 &= \leftarrow recien\_nacido(marta).\\
\phi_3 &= recien\_nacido(marta).\\
\gamma_4 &= \square
\end{aligned}
$$

- 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:

$$\gamma_0 \overset{\phi_0}{\rightsquigarrow} \gamma_1 \cdots \gamma_{n-1} \overset{\phi_{n-1}}{\rightsquigarrow} \gamma_n$$

  donde $\gamma_n = \square$, se llama refutación SLD de γ_0.
- Una derivación de la metaγ_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 dervidada 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

$$\leftarrow orgulloso(Z)$$

$$\leftarrow padre(Z, Y_0), recien\_nacido(Y_0)$$

$$\leftarrow papa(Z, Y_0), recien\_nacido(Y_0) \qquad \leftarrow mama(Z, Y_0), recien\_nacido(Y_0)$$

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

$$\square$$

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 $\leftarrow \phi_1, \ldots, \phi_m$. Entonces $\forall\ ((\phi_1 \wedge \cdots \wedge \phi_m)\theta)$ es una consecuencia lógica del programa Δ.

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

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. `alejandro_guerra`, `rc`, etc.
- Es posible que necesitemos multiples identificadores: Ej. `6183` puede denotar al mismo individuo que `alejandro_guerra`.
- Otros individuos con nombre: Entidades legales, lugares, objetos. Ej. `iphone15`, `iiia`, 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. `no_personal(6183)`, `prof(nicandro_cruz)`, `inst(iiia)`, etc.

Universidad Veracruzana

Atributos

- Los predicados unarios también pueden representar propiedades de nuestros objetos:
  Ej. `sni(6183)`, `sni(nicandro_cruz)`, `pnpc(mia)`, 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. `adscripcion(6183,iiia)`, `prof(rc,6183)`, etc.
- No olviden que hay relaciones que no son binarias:
  Ej. `horario(rc,10,12,[martes, jueves])`, etc.
- Las funciones pueden tomar varios argumentos, pero suelen ser unarias:
  Ej. $s(6) \mapsto 7$, etc.
- Todas las funciones se asumen totales, si hay alguien sin sucesor en el dominio, deberíamos considerar definir $s/1$ como una relación binaria:
  Ej. `s(6,7)`, etc.

Universidad Veracruzana

Hechos y reglas

- Con este vocabulario nuestros hechos simples pueden representarse como literales, i.e., predicados atómicos o su negación:
  Ej. `prof(6183)`, ¬`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. `martin_aguilar == Rector(uv)`, etc.
- También podemos definir reglas:
  Ej. `investigador_en(Invest) :- sni1(Invest).`

Universidad Veracruzana

Reglas terminológicas I

Disjunto. Dos predicados son disjuntos si la neg. de uno implica al otro:
Ej. `menor_edad(X) :- \+ mayor_edad(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).`

Universidad Veracruzana

Reglas terminológicas II

Restricción. Establecen una restricción de tipo de objeto:
Ej. `profEn(Prof,iiia) :- sni(Prof).`

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,PUs) :- precio(C,PMx), PUs = PMx/19.80`

Universidad Veracruzana

Otros hechos

Estadísticos. Incluyen información sobre la probabilidad o la proporción de individus que satisfacen el predicado.
Ej. La mayoría de los empleados está de vacaciones.

Defaults. Incluye caracerísticas 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 hores hora de irse de vacaciones y lo desea.

Universidad Veracruzana

Referencias I

[1] RJ Brachman y HJ Levesque. *Knowledge representation and reasoning*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2004.

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

Universidad Veracruzana