Cláusulas de Horn, Unificación y Resolución (CALP 2022)

FaMAF

16 de novembro de 2022

Partes

Cláusulas de Horn

Unificación

Resolución

Bibliografía

Cláusulas de Horn

¿Qué son las cláusulas de Horn?

- **Definición.** Una cláusula de Horn proposicional es una fórmula en *CNF* tal que tiene a lo sumo un literal no negado.
- **Definición.** Una cláusula de Horn de primer orden es una fórmula de primer orden de la siguiente forma

$$\forall \vec{x}.(J \leftarrow J_1 \wedge \ldots \wedge J_n)$$

  Con $FV(J_i) \cup FV(J) \subseteq \vec{x}$ y J_i, J puede ser $\perp$ o $\top$.
- Trabajamos con los cuantificadores implícitos, es decir, vemos la cláusla de Horn así: $J \leftarrow J_1 \wedge \ldots \wedge J_n$

Cláusulas de Horn

$$
\begin{aligned}
& J \leftarrow J_1 \wedge \ldots \wedge J_n \\
\equiv & \{P \rightarrow Q \leftrightarrow \neg P \vee Q\} \\
& \neg(J_1 \wedge \ldots \wedge J_n) \vee J \\
\equiv & \{De\ Morgan\} \\
& \neg J_1 \vee \ldots \vee \neg J_n \vee J
\end{aligned}
$$

- Es claro que hay a lo sumo un literal positivo y el resto son negados.

Reglas, hechos y goals

Las cláusulas de Horn se clasifican así,

- **Regla**: $J \leftarrow J_1 \wedge \ldots \wedge J_n$
- **Hecho**: $J \leftarrow \top$
- **Goal**: $\bot \leftarrow J_1 \wedge \ldots \wedge J_n$
- Las reglas y los hechos son **cláusulas definitivas**.

Hechos

$$
\begin{aligned}
& J \leftarrow \top \\
\equiv & \{P \rightarrow Q \leftrightarrow \neg P \vee Q\} \\
& \neg \top \vee J \\
\equiv & \{neutro\ de\ \vee\} \\
& J
\end{aligned}
$$

Goals

$$
\begin{aligned}
& \bot \leftarrow J_1 \wedge \ldots \wedge J_n \\
\equiv & \{P \rightarrow Q \leftrightarrow \neg P \vee Q\} \\
& \neg(J_1 \wedge \ldots \wedge J_n) \vee \bot \\
\equiv & \{De\ Morgan\} \\
& \neg J_1 \vee \ldots \vee \neg J_n \vee \bot \\
\equiv & \{neutro\ de\ \vee\} \\
& \neg J_1 \vee \ldots \vee \neg J_n
\end{aligned}
$$

Resolución (proposicional)

$$\underbrace{\frac{\varphi \vee \ell \quad \psi \vee \bar{\ell}}{\varphi \vee \psi}}_{\textit{RESOLVENTE}}$$

Notemos que modus ponens es un caso especial de resolución,

$$\frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \qquad \frac{\neg\varphi \vee \psi \quad \varphi}{\psi}$$

Propiedad de las clásulas de Horn

Las clásulas de Horn son intersante porque,

Propiedad: La resolución entre dos cláusulas de Horn tiene una cláusula de Horn como resolvente.

Programa lógico

- **Definición:** Un programa lógico Γ está compuesto por cláusulas de Horn definitivas, es decir, reglas y hechos.
- **Convención:** constantes y nombres de relacioens están en minúsculas pero las variables en mayúscula.
- **Ejemplo:**

$$
\begin{aligned}
&\mathit{cuatropatas}(\mathit{firulais}) \\
&\mathit{ladra}(\mathit{firulais}) \\
&\mathit{perro}(X) \leftarrow \mathit{cuatropatas}(X) \wedge \mathit{ladra}(X)
\end{aligned}
$$

Queries sobre un programa lógico

Hagamos de cuenta que queremos encontrar un X tal que $perro(X)$ en el programa Γ_{perro}

$$
\begin{aligned}
& \exists \vec{x}. J_1 \wedge \ldots \wedge J_n \\
\equiv & \{\exists x \varphi \leftrightarrow \neg \forall x \neg \varphi\} \\
& \neg \forall \vec{x}. \neg (J_1 \wedge \ldots \wedge J_n) \\
\equiv & \{goal\} \\
& \neg \forall \vec{x}. (\perp \leftarrow J_1 \wedge \ldots \wedge J_n)
\end{aligned}
$$

Esto último quiere decir que para ver la respuesta a la querie no se tiene que cumplir la goal $\perp \leftarrow J_1 \wedge \ldots \wedge J_n$. Esto se entenderá mejor cuando veamos resolución SLD.

Unificación: sustitución

- **Defínción:** Una sustitución θ mapea variables libres a términos, o sea, $\theta = x_1 \mapsto t_1, ..., x_n \mapsto t_n$
- $x_i\theta = t_i$
- $x\theta = x$, si x no está entre las variables libres.
- $f(t_1, ..., t_n)\theta = f(t_1\theta, ..., t_n\theta)$

Unificación: sustitución

- **Unificador:** Un unificador θ entre dos términos t y s es una sustitución tal que $t\theta = s\theta$
- Una sustitución θ_2 es más general otra sustitución θ_1 si existe alguna sustitución τ tal que $\theta_1 = \theta_2 \circ \tau$
- **MGU:** un unificador θ_{MGU} entre dos términos, es aquel tal que todo otro unificador es menos general.

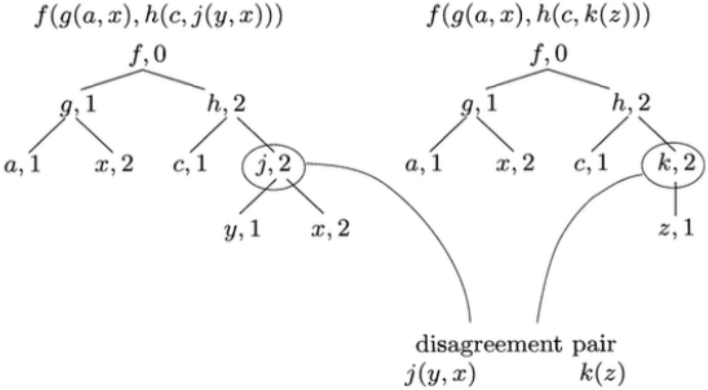
Unificación: ejemplo

Sean $t = p(f(g(x)), y, z)$ y $s = p(u, u, f(u))$,

- $\theta_{MGU} = y \mapsto f(g(x)),\ z \mapsto f(f(g(x))),\ u \mapsto f(g(x))$
- $\theta_1 = x \mapsto a,\ y \mapsto f(g(a)),\ z \mapsto f(f(g(a))),\ u \mapsto f(g(a))$
- $\theta_2 = x \mapsto h(a), y \mapsto f(g(h(a)))\ z, \mapsto f(f(g(h(a)))),\ u \mapsto f(g(h(a)))$

Unificación: ¿Cómo encontrar el MGU?

- **Par en desacuerdo** d_1, d_2 es un par en desacuerdo de dos términos t_1, t_2 si d_1, d_2 son subtérminos de t_1, t_2 y es el primer nodo del árbol sintáctico entre términos que difiere.
- **Ejemplo**



Unificación: algoritmo

$$
\begin{array}{l}
\{PRE : t_1, t_2 \textit{ terminos a unificar}\} \\
\quad \textit{let } \theta := \epsilon \\
\quad \textit{while } t_1\theta \neq t_2\theta \textit{ do} \\
\qquad \textit{choose } d_1\ d_2 \ = \ parEnDesacuerdo(t_1\theta, t_2\theta) \\
\qquad \textit{if}(\textit{ni } d_1 o\ d_2 \textit{ son variables}) \textit{ then FAIL} \\
\qquad \textit{let } x, t = (d_1, d_2 x \textit{ es variable } t \textit{ el otro}) \\
\qquad \textit{if } (x \textit{ ocurre en } t) \textit{ then FAIL} \\
\qquad \textit{let } \theta := \theta\{x \mapsto t\} \\
\{POST : \theta \textit{ MGU}\}
\end{array}
$$

Resolución

$$\frac{\varphi \vee \mathsf{t} \qquad \psi \vee \neg t'}{(\varphi \vee \psi)\theta_{MGU(t,t')}}$$

Resolución: ejemplo

Asumiendo como goal $\neg perro(firulais)$,

$$\dfrac{\dfrac{\dfrac{\neg \mathrm{ladra}(X) \lor \neg \mathrm{cuatropatas}(X) \lor \mathrm{perro}(X) \qquad \mathrm{ladra}(\mathrm{firulais})}{\neg \mathrm{cuatropatas}(\mathrm{firulais}) \lor \mathrm{perro}(\mathrm{firulais}) \qquad \mathrm{cuatropatas}(\mathrm{firulais})}\{X \to \mathrm{firulais}\}}{\mathrm{perro}(\mathrm{firulais}) \qquad \neg \mathrm{perro}(\mathrm{firulais})}\varepsilon}{\bot}\varepsilon$$

Llegamos a una contradicción o derivar una cláusula vacía, por lo tanto un $X = firulais$ sirve para $perro(X)$.

Bibliografía

- Logic Programming, Mark Pfenning.
- FOL and automated theorem proving, Melvin Fitting.