

Semántica Operacional para la Programación en Lógica

Lógica para Ciencias de la Computación

Primer Cuatrimestre de 2008

– Material Adicional –

Semánticas para la Programación en Lógica

- Se han explorado distintas alternativas para capturar **formalmente** la semántica de los programas lógicos:
 - ➡ Semántica Operacional
 - ➡ Semántica Declarativa
 - ➡ Semántica de Punto Fijo
- En lo que resta, abordaremos esta primer variante, la **Semántica Operacional**.

Basamentos Teóricos

- La Programación en Lógica descansa sobre un subconjunto del Cálculo de Predicados, conocido como **Lógica de Horn**.
 - ➡ Un **programa lógico** es un conjunto de cláusulas de Horn.
 - ➡ Una **cláusula de Horn** es una cláusula con a lo sumo un literal positivo.
 - ➡ Una **cláusula** es una disyunción de literales.
 - ➡ Un **literal** es un predicado o su negación.

Posibles variantes de cláusulas de Horn

❖ Sólo un literal positivo:

➡ A

❖ Un literal positivo y algunos negativos:

➡ $A \vee \neg B_1 \vee \neg B_2 \vee \dots \vee \neg B_n$

➡ $A \vee \neg (B_1 \wedge B_2 \wedge \dots \wedge B_n)$

➡ $A \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_n$

❖ Sólo literales negativos:

➡ $\neg B_1 \vee \neg B_2 \vee \dots \vee \neg B_n$

➡ $\neg (B_1 \wedge B_2 \wedge \dots \wedge B_n)$

➡ $\leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_n$

meta definida o
consulta

En Prolog

$A.$
(hecho)

$A:- B_1, B_2, \dots, B_n$
(regla)

$?- B_1, B_2, \dots, B_n$
(consulta)

Composición de Sustituciones

Def: Una **sustitución** es un conjunto finito de pares ordenados $[X_i, t_i]$ donde X_i es una variable y t_i es un término, tales que para cada $i \neq j$, se verifica que $X_i \neq X_j$, y a su vez para cada i, j se verifica que X_i no aparece en t_j .

Intuitivamente, la composición de dos sustituciones σ y θ es una nueva sustitución λ tal que $(A\sigma)\theta = A\lambda$

Def: Sean $\sigma = \{ [U_1, s_1], \dots, [U_m, s_m] \}$ y $\theta = \{ [V_1, t_1], \dots, [V_n, t_n] \}$ sendas sustituciones, tales que $U_i \neq V_j$ y U_i no aparece en t_j , para todo par i, j tal que $1 \leq i \leq m$ y $1 \leq j \leq n$. Entonces, definimos la **composición** de σ con θ , notado $\sigma\theta$, de la siguiente manera:

$$\sigma\theta = \{ [U_1, s_1\theta], \dots, [U_m, s_m\theta] \} \cup \{ [V_1, t_1], \dots, [V_n, t_n] \}$$

Comp. de Sust.: Ejemplo

$$\sigma = \{ [X, f(S)], [Y, a], [Z, T] \}$$

$$\theta = \{ [S, W], [T, b] \}$$

$$\sigma\theta =$$

$$\{ [X, f(S)\theta], [Y, a\theta], [Z, T\theta] \} \cup \{ [S, W], [T, b] \} = \\ \{ [X, f(W)], [Y, a], [Z, b], [S, W], [T, b] \}$$

Def: Sean $\sigma = \{ [U_1, s_1], \dots, [U_m, s_m] \}$ y $\theta = \{ [V_1, t_1], \dots, [V_n, t_n] \}$ sendas sustituciones, tales que $U_i \neq V_j$ y U_i no aparece en t_j , para todo par i, j tal que $1 \leq i \leq m$ y $1 \leq j \leq n$. Entonces, definimos la **composición** de σ con θ , notado $\sigma\theta$, de la siguiente manera:

$$\sigma\theta = \{ [U_1, s_1\theta], \dots, [U_m, s_m\theta] \} \cup \{ [V_1, t_1], \dots, [V_n, t_n] \}$$

Comp. de Sust.: Ejemplo

$$\sigma = \{ [X, f(S)], [Y, a], [Z, T] \}$$

$$\theta = \{ [S, W], [Z, b] \}$$

$$\sigma = \{ [X, f(S)], [Y, a] \}$$

$$\theta = \{ [Z, Y] \}$$

$$\sigma = \{ [X, f(S)], [Y, a] \}$$

$$\theta = \{ [S, g(X)] \}$$

Def: Sean $\sigma = \{ [U_1, s_1], \dots, [U_m, s_m] \}$ y $\theta = \{ [V_1, t_1], \dots, [V_n, t_n] \}$ sendas sustituciones, tales que $U_i \neq V_j$ y U_1 no aparece en t_j , para todo par i, j tal que $1 \leq i \leq m$ y $1 \leq j \leq n$. Entonces, definimos la **composición** de σ con θ , notado $\sigma\theta$, de la siguiente manera:

$$\sigma\theta = \{ [U_1, s_1\theta], \dots, [U_m, s_m\theta] \} \cup \{ [V_1, t_1], \dots, [V_n, t_n] \}$$

Sustituciones Unificadoras y m.g.u./s

Def: Sea S un conjunto de términos/fbfs. Diremos que θ es una **sustitución unificadora** (o simplemente **unificador**) para S si $S\theta$ es un conjunto unitario.

Def: Sea θ un unificador para S . Diremos que θ es el **unificador más general** (o bien, **m.g.u.**) de S si se verifica que para todo otro unificador σ para S , existe una sustitución γ tal que $\sigma = \theta\gamma$.

Sust. Unif. y m.g.u./s: Ejemplos

$S = \{p(X), p(Y)\}$ tiene varios posibles unificadores:

- ➡ $\theta_1 = \{[X, a], [Y, a]\}$
- ➡ $\theta_2 = \{[X, f(Z)], [Y, f(Z)]\}$
- ➡ $\theta_3 = \{[X, Y]\}$

En particular, θ_3 es un unificador más general para S .

Ejercicio:

Hallar unificadoras para $S = \{p(X), p(Y), p(g(Z))\}$.
Obtener un unificador más general para S .

Construcción de Árboles SLD

Def: Sea P un programa definido y G una meta definida. El árbol SLD para $P \cup G$ es un árbol que satisface las siguientes condiciones:

1. Cada nodo del árbol se etiqueta con una meta definida.
2. El nodo raíz debe estar etiquetado con G .
3. Sea $\leftarrow A_1 \wedge \dots \wedge A_n$, $n > 0$, la etiqueta de algún nodo y sea A_k la submeta seleccionada. Para cada cláusula $A \leftarrow B_1 \wedge \dots \wedge B_m$ de P tal que A_k y A unifiquen con un m.g.u. θ , debe existir un nodo hijo del nodo en cuestión que esté etiquetado con:

$$\leftarrow A_1\theta \wedge \dots \wedge A_{k-1}\theta \wedge B_1\theta \wedge \dots \wedge B_m\theta \wedge A_{k+1}\theta \wedge \dots \wedge A_n\theta$$

Ejemplo

1. $p(X) :- q(X), r(X).$

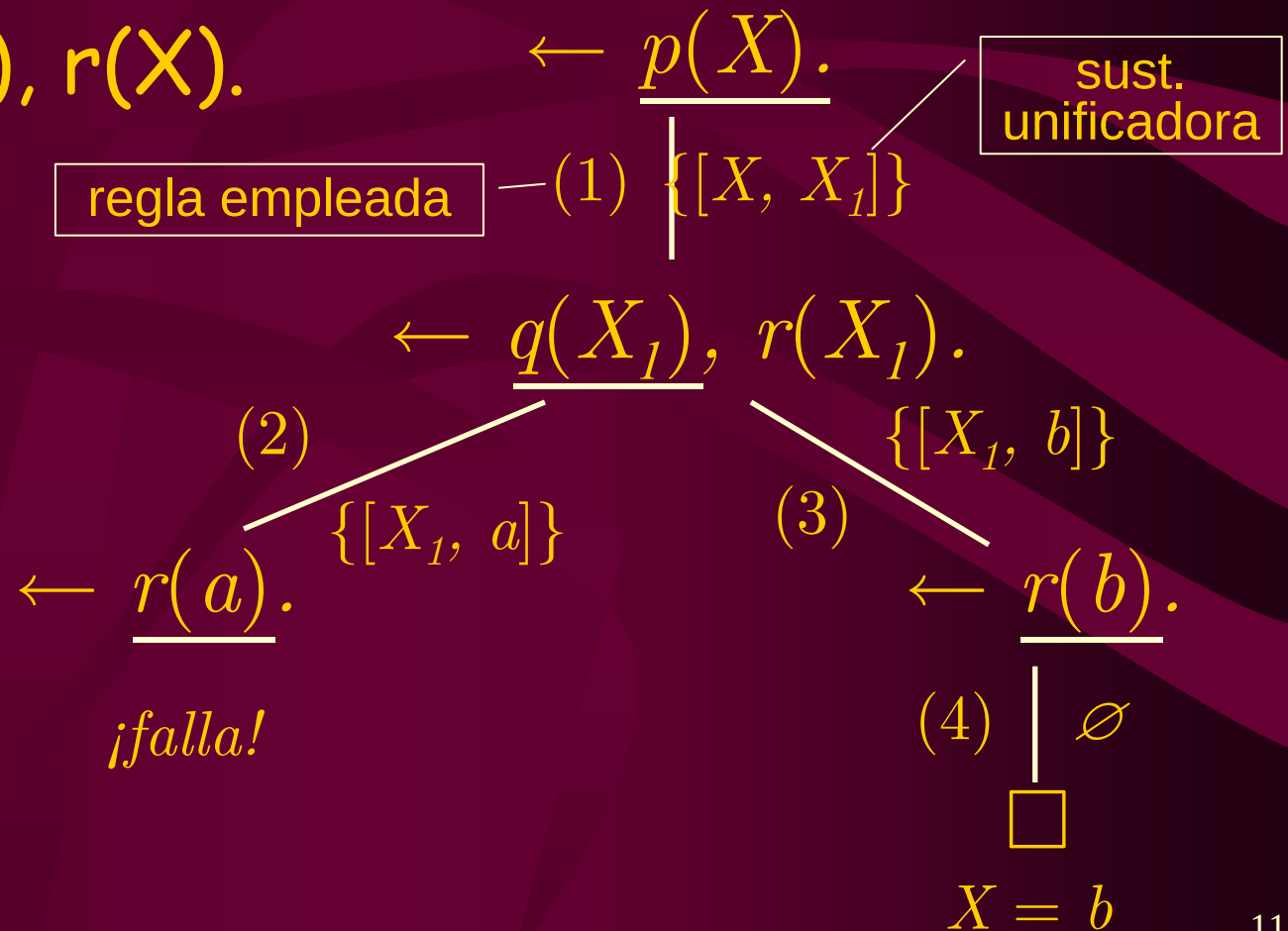
2. $q(a).$

3. $q(b).$

4. $r(b).$

?- $p(X).$

$X = b$



Tipos de nodos hoja en los Árboles SLD

- Existen dos tipos de nodo hoja:
 - ➡ **Nodos hoja exitosos:** son aquellos etiquetados con la cláusula vacía.
 - ➡ **Nodos hoja fallidos:** son aquellos que no están etiquetados con la cláusula vacía.
- Los nodos exitosos señalan que se ha encontrado una **respuesta** a la consulta inicial.
- Dicha **respuesta** se obtiene **componiendo** a las distintas sustituciones realizadas a lo largo de la rama que contiene al nodo hoja exitoso en cuestión.

Obtención de la respuesta: Ejemplo

1. $p(X) :- q(X), r(X).$

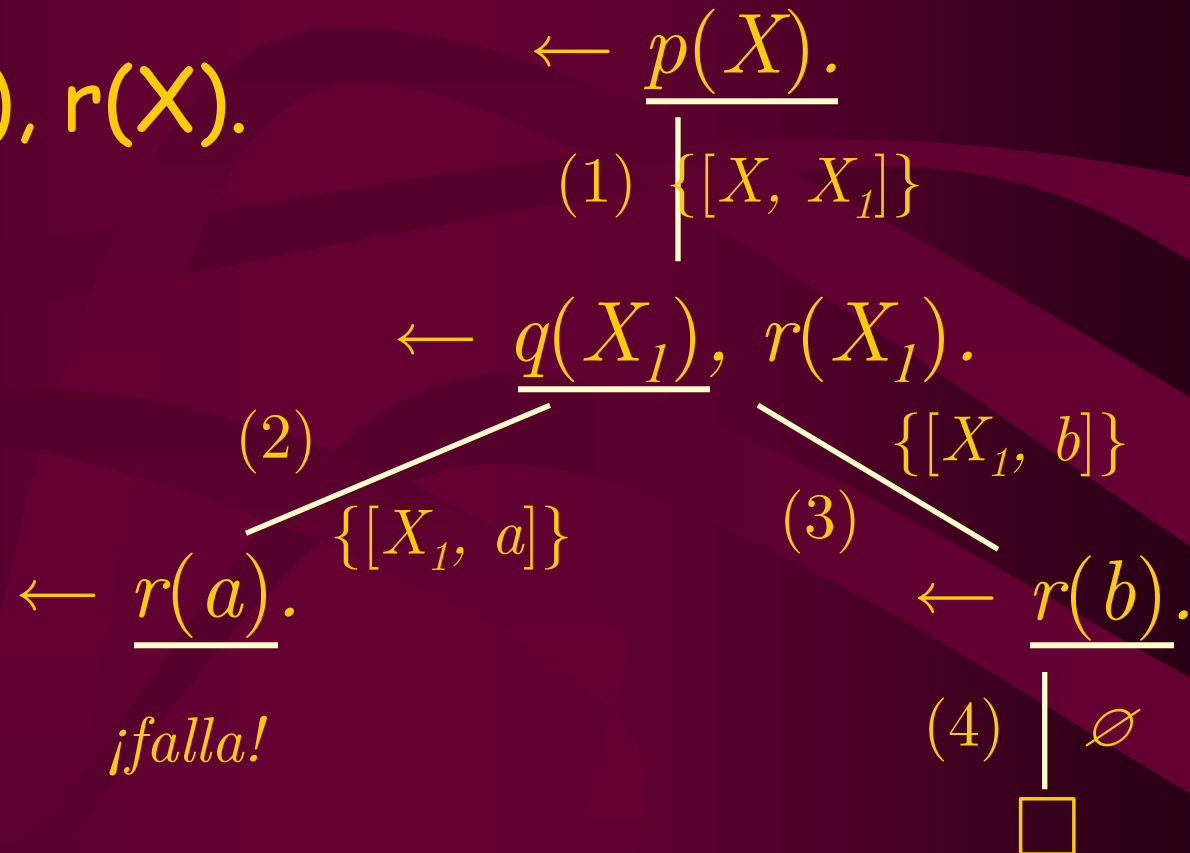
2. $q(a).$

3. $q(b).$

4. $r(b).$

?- $p(X).$

$X = b$



$$((\{[X, X_1]\} \{[X_1, b]\}) \emptyset) = \{[X, b] [X_1, b]\}$$

Semántica Operacional

- La **Semántica Operacional** para la Programación en Lógica se basa en poder capturar de manera formal la **existencia de los nodos exitosos** dentro de un árbol SLD.
- El siguiente conjunto de definiciones formales apunta a ese objetivo, explorando en profundidad **sólo una rama del árbol**.

Derivación entre cláusulas

Def: Sea G una meta de la forma $\leftarrow A_1 \wedge \dots \wedge A_n$, $n > 0$, y sea C una cláusula de la forma $A \leftarrow B_1 \wedge \dots \wedge B_m$, $m \geq 0$. Diremos que G' se deriva de G y C , usando un m.g.u. θ , si se verifican las siguientes condiciones:

1. A_k es uno de las submetas que aparecen en G .
2. θ es el unificados más general entre entre A_k y A .
3. G' guarda exactamente la forma:

$$\leftarrow A_1\theta \wedge \dots \wedge A_{k-1}\theta \wedge B_1\theta \wedge \dots \wedge B_m\theta \wedge A_{k+1}\theta \wedge \dots \wedge A_n\theta$$

Derivación y Refutación SLD

Def: Sea P un programa definido y G una meta definida. Una derivación SLD para $P \cup \{G\}$ consiste de:

- ➡ una secuencia de metas

$G_0, G_1, G_2, \dots, y$

- ➡ una secuencia de variantes de cláusulas de programa

$C_1, C_2, \dots, y$

- ➡ una secuencia de m.g.u.'es

$\theta_1, \theta_2, \dots,$

tales que $G_0 = G$, y cada G_{i+1} , $i \geq 0$, se deriva de G_i y de C_{i+1} mediante el m.g.u. θ_{i+1} .

Def: Sea P un prog. definido y G una meta definida. Una **refutación SLD** para $P \cup \{G\}$ es una derivación SLD para $P \cup \{G\}$ que tiene a la cláusula vacía como última meta de la derivación.

1. $p(X) :- q(X), r(X).$
2. $q(a).$
3. $q(b).$
4. $r(b).$

Ejemplo

Refutación SLD

$\leftarrow p(X).$

$p(X_1) :- q(X_1), r(X_1).$

$\{[X, X_1]\}$

$\leftarrow q(X_1), r(X_1).$

$q(b).$

$\{[X_1, b]\}$

$\leftarrow r(b).$

$r(b).$

$\emptyset$



Rama exitosa del árbol SLD

$\leftarrow \underline{p(X)}.$

(1) $\mid \{[X, X_1]\}$

$\leftarrow \underline{q(X_1)}, r(X_1).$

(3) $\mid \{[X_1, b]\}$

$\leftarrow \underline{r(b)}.$

(4) $\mid \emptyset$



1. $p(X) :- q(X), r(X).$

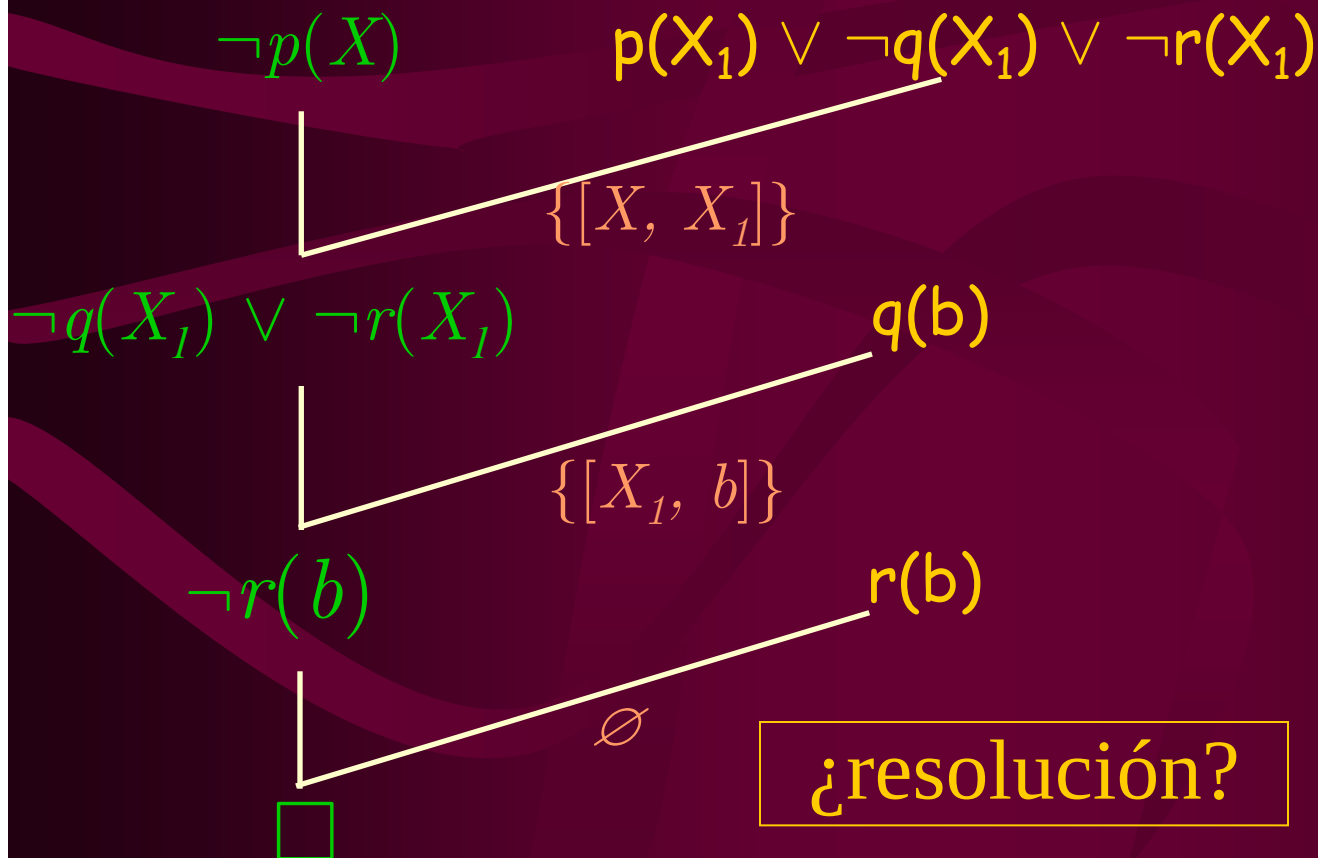
2. $q(a).$

3. $q(b).$

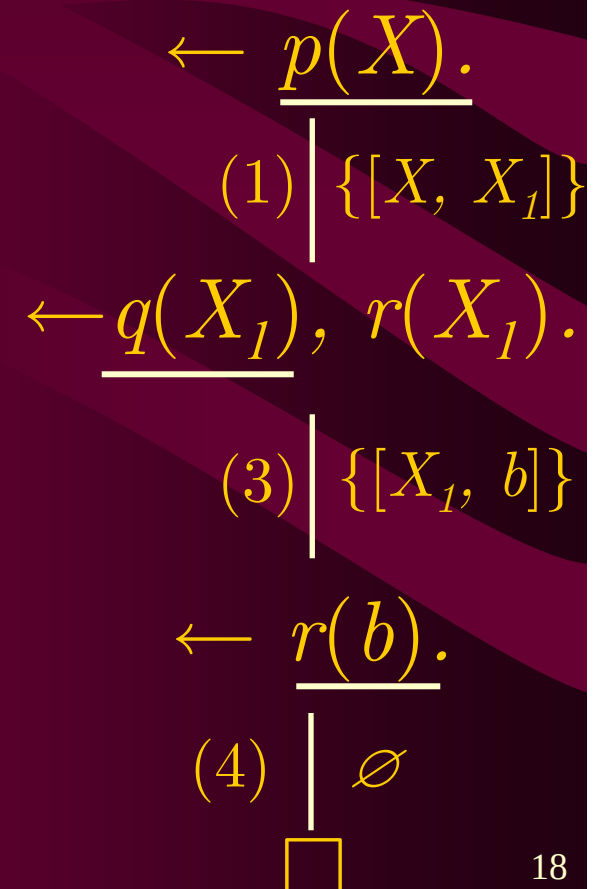
4. $r(b).$

Ejemplo

Refutación SLD



Rama exitosa del árbol SLD



Conjunto de Éxito

Def: Sea P un programa definido. El conjunto de éxito de P es el conjunto de todos los $A \in \mathbb{B}_P$ tales que $P \cup \{ \leftarrow A. \}$ tiene una refutación SLD.

Intuitivamente, es el conj. de todos los átomos fijos (sin variables) construibles utilizando predicados, funtores y constantes que aparecen en P

El conjunto de éxito de un programa definido P constituye su semántica operacional.