

Cálculo de Predicados (2)

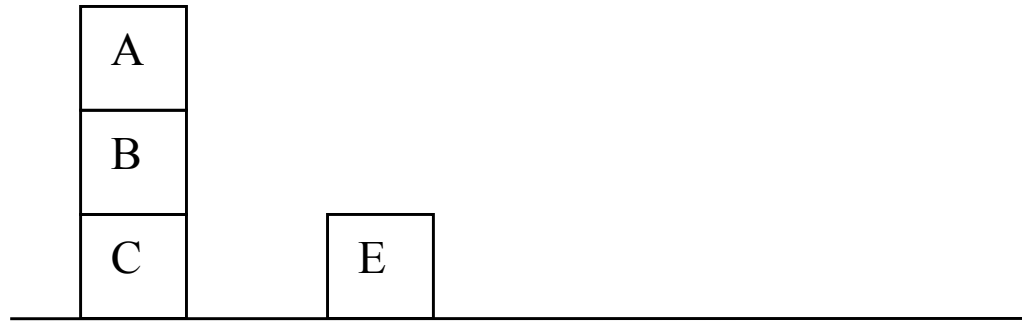
Carlos Hurtado L.

Depto de Ciencias de la Computación,
Universidad de Chile

Clase Anterior

- Sintaxis y semántica de lógica de predicados
- Noción de interpretación y modelo.
- Noción de Validez y Satisfabilidad.
- Consecuencia lógica.

Clase Anterior: Ejemplo: Mundo de los Bloques



- Vocabulario (F, P, V):
 - $F = \{a, b, c, e\}$
 - $P = \{on, above, clear, ontable\}$
 - $V = \{x, y, z, \dots\}$

Clase Anterior: Ejemplo: Interpretación

Interpretación para $\forall x,y.(on(x,y) \Rightarrow above(x,y))$:

$I=(D,\varphi,\pi,\mu)$, donde:

- $D=\{A,B,C,E\}$
- φ :
 $\varphi(a)=A, \varphi(b)=B, \varphi(c)=C, \varphi(e)=E$
- π :
 $\pi(on)=\{(A,B),(B,C)\}$
 $\pi(above)=\{(A,B), (B,C), (A,C)\}$
 $\pi(clear) = \{A,E\}$
 $\pi(ontable) = \{C,E\}$
- μ :
vacía para las sentencias



Clase Anterior: Consecuencia Lógica

- Una fórmula S es consecuencia lógica de una fórmula P , denotado $P \models S$ ssi

$$M(P) \subseteq M(S)$$

- S es equivalente a P ssi $P \models S$ y $S \models P$

Clase Anterior: Ejemplo: Consecuencia Lógica

- Sea $KB = \{on(b,c), on(c,d), clear(e), \forall x,y,z.(on(x,y) \wedge on(y,z) \Rightarrow above(x,z))\}$.
- ¿ $KB \models above(b,d)$?
 - Sea I un modelo de KB ,
 - Luego $I(on(b,c) \wedge on(c,d) \Rightarrow above(b,d)) = V$
 - Es decir, $I(on(b,c) \wedge on(c,d) \Rightarrow above(b,d)) = V$
 - Por lo tanto, $I(above(b,d)) = V$.

Resolución en Lógica de Predicados

- La refutación mediante resolución se puede adaptar a lógica de predicados.
- Con esto se obtiene un método de prueba correcto y completo.
- Refutación mediante resolución es la base de los demostradores automáticos de teoremas, razonamiento en sistemas expertos y Prolog.

Resolución en Lógica de Predicados

- Para aplicar la regla de resolución debemos transformar toda fórmula como un conjunto (conjunción) de cláusulas.
- Idea central: Antes de resolver encontraremos una sustitución de variables que "unifique" las dos cláusulas.

Ejemplo: Mundo de los Bloques

- Sea $KB = \{on(b,c), on(c,d), clear(e), \forall x,y,z.(on(x,y) \wedge on(y,z) \Rightarrow above(x,z))\}$.
- ¿ $KB \models above(b,d)$?
- Transformamos $KB \wedge \neg above(b,d)$ a conjunto de cláusulas:
 $\{ (on(b,c)), (on(c,d)), (clear(e)), (\neg on(x,y), \neg on(y,z), above(x,z)), \neg above(b,d) \}$.

Cláusulas

- Cláusulas en cálculo de predicados:

$$\forall x_1, x_2, \dots, x_n. (A_1 \vee A_2 \vee \dots \vee A_n),$$

donde cada A_i es un literal (átomo positivo o negativo) que puede contener ocurrencias de variables x_i 's.

- Usamos notación similar a Cálculo Proposicional.
- Ejemplos:

$$(P(\text{Juan}), Q(\text{Pedro}), R(X))$$

$$(\neg P(Y), R(\text{Suzana}), R(Y))$$

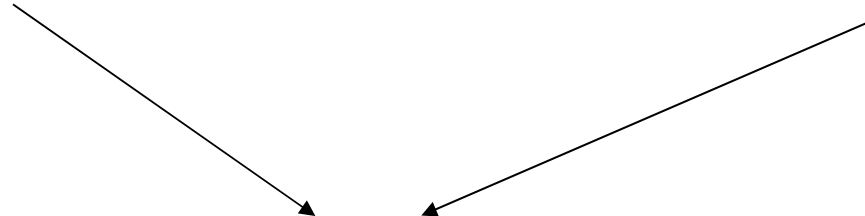
- Toda fórmula en lógica de predicados se puede transformar en un conjunto de cláusulas equivalentes.
- **Cláusulas base:** cláusula sin variables.

Tarea

- Investigar cómo se transforma una fórmula en lógica de predicados en un conjunto de cláusulas equivalentes.

Ejemplo: Mundo de los Bloques

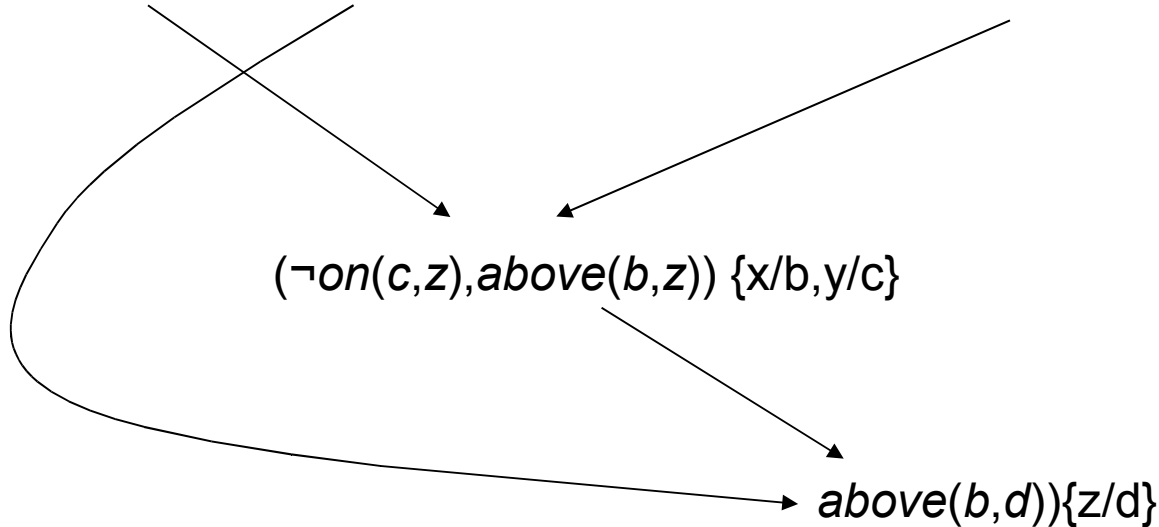
$\{(on(b,c)), (on(c,d)), (clear(e)), (\neg on(x,y), \neg on(y,z), above(x,z)), (\neg above(b,d))\}.$



$(\neg on(c,z), above(b,z)) \{x/b, y/c\}$

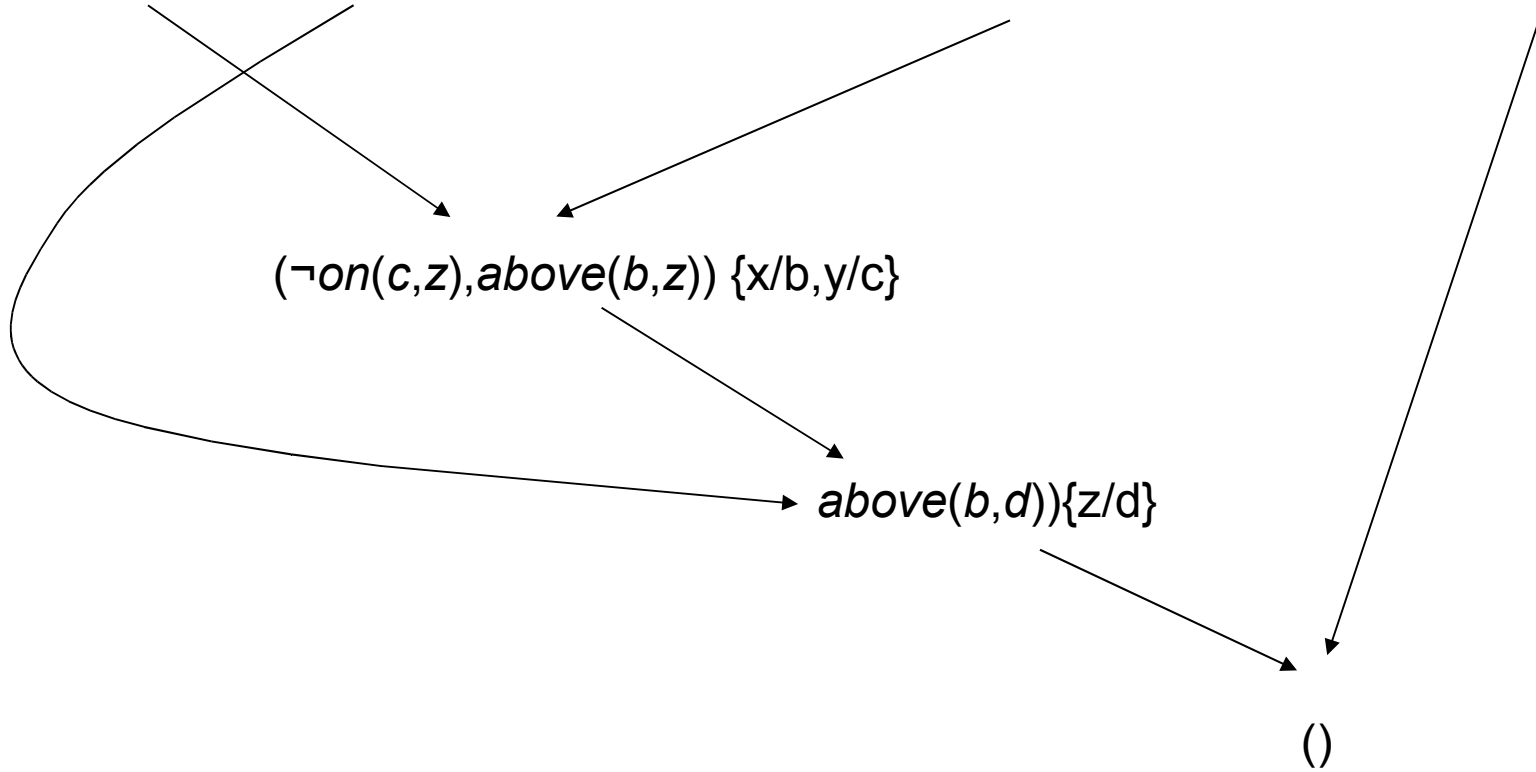
Ejemplo: Mundo de los Bloques

$\{(on(b,c)), (on(c,d)), (clear(e)), (\neg on(x,y), \neg on(y,z), above(x,z)), (\neg above(b,d))\}.$



Ejemplo: Mundo de los Bloques

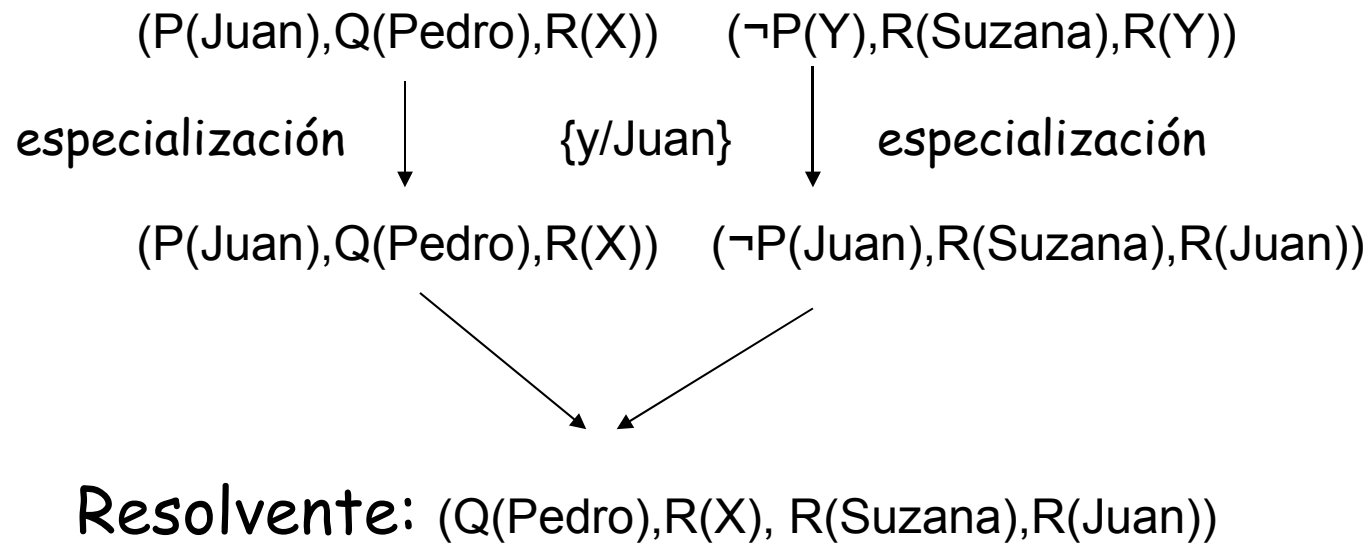
$\{(on(b,c)),(on(c,d)),(clear(e)),(\neg on(x,y),\neg on(y,z),above(x,z)),(\neg above(b,d))\}.$



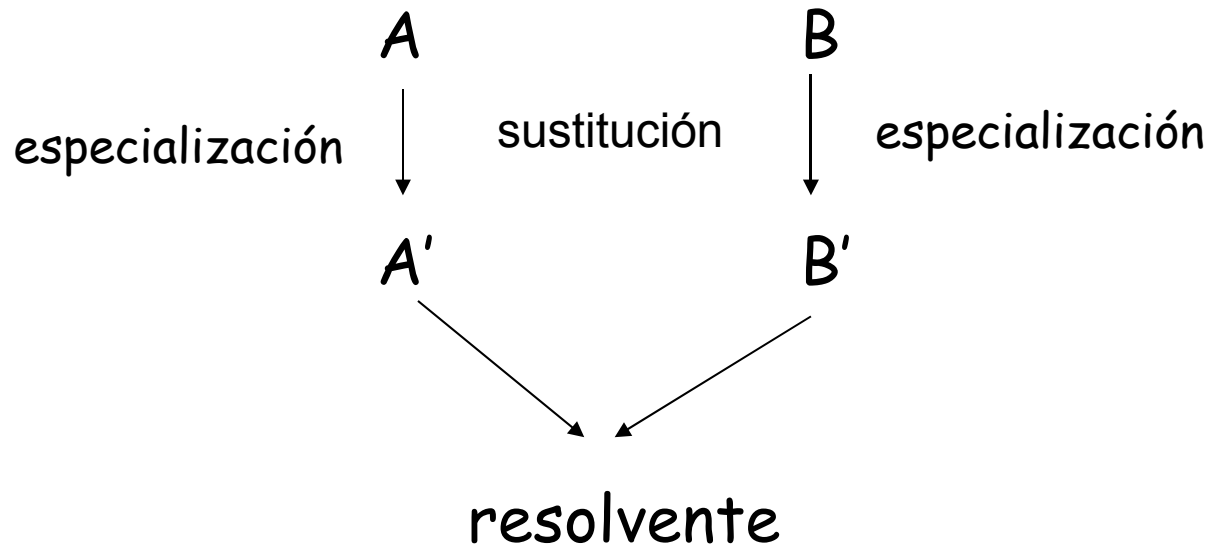
Resolución en Cálculo de Predicados

- ¿Qué sucede si intentamos resolver las siguientes dos cláusulas?
A: $(P(\text{Juan}), Q(\text{Pedro}), R(X))$
B: $(\neg P(Y), R(\text{Suzana}), R(Y))$?
- Intuitivamente, B representa un conjunto de cláusulas base:
B1: $(\neg P(\text{Pedro}), R(\text{Suzana}), R(\text{Pedro}))$
B2: $(\neg P(\text{Juan}), R(\text{Suzana}), R(\text{Juan}))$
... etc
- Luego existe una "especialización" de B que puede ser resuelta con A.

Resolución en Cálculo de Predicados

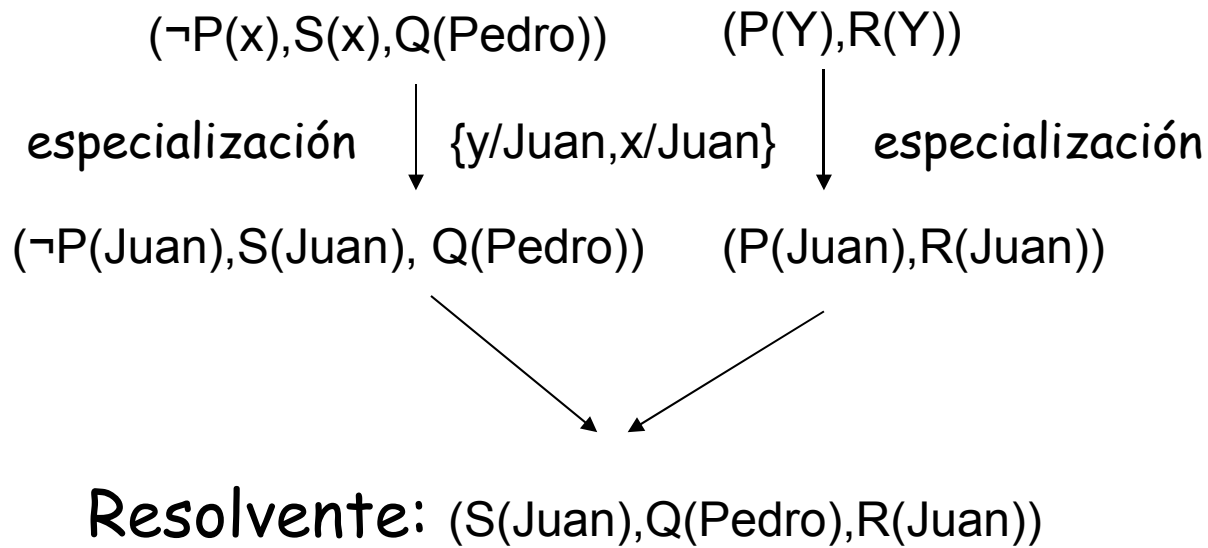


Resolución en Cálculo de Predicados

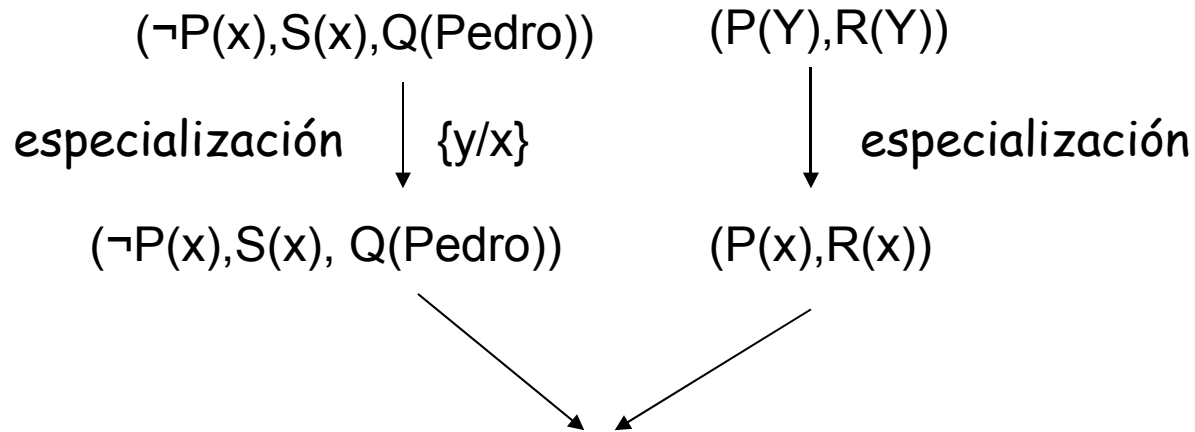


Problema: si la especialización es mayor que la necesaria perdemos información al resolver.

Ejemplo



Ejemplo



Ejemplo

A: $(\neg P(x), S(x), Q(\text{Pedro}))$

B: $(P(Y), R(Y))$

- Posibles resolventes:
 - $C1: (S(\text{Juan}), Q(\text{Pedro}), R(\text{Juan}))$ vía $\{y/\text{Juan}, x/\text{Juan}\}$
 - $C2: (S(x), Q(\text{Pedro}), R(x))$ vía $\{y/x\}$
- Tenemos que $C2 \models C1$
- Decimos que $C2$ es "mas general" que $C1$.
- Resolución en cálculo proposicional consiste en encontrar el resolvente más general.

Sustitución

- Una sustitución es un conjunto de expresiones x/t , donde x es una variable y t es un término que no contiene x .
- Dada una fórmula A y una sustitución σ , denotamos $A\sigma$ la fórmula que resulta de aplicar σ simultáneamente a las variables de A .

- Ejemplo:

$$(P(x), G(y, z))\{x/y, y/f(a)\} = (P(y), G(f(a), z))$$

- Observar que la sustitución no se aplica secuencialmente sino simultáneamente.

- Ejemplo:

$$(P(x), G(y, z))\{x/y, y/f(a)\} \neq (P(f(a)), G(f(a), z))$$

Composición de Sustituciones

Dados:

$$\sigma = \{x_1/s_1, x_2/s_2, \dots, x_n/s_n\}$$

$$\varphi = \{y_1/t_1, y_2/t_2, \dots, y_m/t_m\}$$

Definimos $\sigma\varphi$ como la siguiente sustitución:

1. Sea $\sigma\varphi = \{x_1/s_1\varphi, x_2/s_2\varphi, \dots, x_n/s_n\varphi, y_1/t_1, \dots, y_m/t_m\}$
2. Borrarnos de $\sigma\varphi$ los pares s/s
3. Borrarnos de $\sigma\varphi$ pares y_i/t_i donde y_i es igual a algún x_j

Ejemplo: Composición de Sustituciones

$\{z/g(x,y)\}\{x/a,y/b,w/c,z/d\}$, siguiendo los pasos descritos:

1. Obtenemos $\{z/g(a,b),x/a,y/b,w/c,z/d\}$
2. No eliminamos pares por esta regla.
3. Eliminamos z/d obteniendo $\{z/g(a,b),x/a,y/b,w/c\}$

Sustituciones

- La sustitución vacía actúa como la identidad: $\sigma\{\}=\sigma$

- Composición es asociativa:

$$(\sigma\varphi)\pi = \sigma(\varphi\pi)$$

- Composición no es conmutativas:

$$\sigma\varphi \neq \varphi\sigma$$

Unificadores

- Un *unificador* para dos fórmulas A y B es una sustitución σ tal que $A\sigma$ es idéntico a $B\sigma$.
- No todas las fórmulas pueden ser unificadas.
- Por ejemplo: no existe un unificador para $(P(f(x),a))$ y $(P(y,f(w)))$ ya que $a/f(w)$ no es un par "válido" en una sustitución (a es una constante).

Renombre de Variables

Para unificar dos fórmulas puede ser necesario renombrar variables antes de unificar.

$$\{p(a, x)\}$$

$$\{\neg p(x, b)\}$$

Por simplicidad, suponemos que las dos cláusulas a resolver no contienen variables comunes.

Unificador más General

- Un unificador σ para A y B es el unificador más general (UMG) ssi para todo unificador φ para A y B existe una sustitución π tal que

$$\varphi = \sigma\pi.$$

Ejemplo: UMG

- Sean $P(f(x),z)$ y $P(y,a)$
- $\sigma=\{y/f(a),z/a,x/a\}$ es un unificador para estas fórmulas ya que

$$P(f(x),z)\sigma = P(y,a)\sigma$$

- Sin embargo, σ no es el UMG ya que tenemos un unificador $\varphi=\{y/f(x),z/a\}$ y

$$\sigma = \varphi\{x/a\}$$

- En este caso φ es el UMG.

Unificador Más General

- Se puede demostrar que si existe un UMG para A y B , este es único (módulo renombre de variables).

Regla Generalizada de Resolución

Dadas dos cláusulas

$(L, Q_1, \dots, Q_n)$ y $(\neg M, R_1, \dots, R_m)$

tales que L y M tienen un UMG σ ,
obtenemos el resolvente:

$(Q_1\sigma, \dots, Q_n\sigma, R_1\sigma, \dots, R_m\sigma)$

Ejemplo: Regla Generalizada de Resolución

Dadas las cláusulas:

$(P(x), Q(g(x)))$ y $(\neg P(a), R(a), Q(z))$

En este caso $L=P(X)$, $M=P(a)$ y su UMG es $\{x/a\}$. Luego obtenemos el resolvente:

$(Q(g(a)), R(a), Q(z))$

Resolución y UMG

- Para Lógica de Predicados la aplicación de la regla de resolución involucra calcular el UMG de dos átomos.
- Es decir, debemos encontrar la forma más general de hacer corresponder las variables de los átomos.
- Algoritmo UNIFICAR: entrega el UMG de dos átomos.

Algoritmo UNIFICAR

Ejemplo: $A := P(x, f(a, y))$ y $B := P(x, f(z, b))$

A y B transforman en listas (notación infija).

Ejemplo: $A := (P \times (f \ a \ y))$ $B := (P \times (f \ z \ b))$

UNIFICAR(A,B)

4. Inicialmente $\sigma := \{\}$
5. Si A y B son expresiones compuestas
 - Para cada par de elementos A_i de A y B_i de B:
 - $\varphi := \text{UNIFICAR}(A_i, B_i)$
 - $A := A\varphi, B := B\varphi, \sigma := \sigma\varphi$
6. En caso contrario:
 - Si $A = \{\}$, $\sigma := \{\}$
 - Si A (resp. B) es una constante y no se puede unificar con B (resp. A), entonces retornamos fallo (no hay unificador).
 - Si A (resp. B) es una variable que no aparece en B (resp. A), entonces $\sigma := \{A/B\}$ (resp. $\sigma := \{B/A\}$)
7. Retornar(σ)

Ejemplo 1

$A := P(x, f(a, y))$ y $B := P(x, f(z, b))$

- $mgu := \text{UNIFICAR}((P \times (f \ a \ y)), (P \times (f \ z \ b)))$
 - $\varphi := \text{UNIFICAR}(P, P)$
 - $\sigma := \{\}$
 - $\varphi := \text{UNIFICAR}(x, x)$
 - $\sigma := \{\}$
 - $\varphi := \text{UNIFICAR}((f \ a \ y), (f \ z \ b))$
 - $\varphi := \text{UNIFICAR}(f, f)$
 - $\sigma := \{\}$
 - $\varphi := \text{UNIFICAR}(a, z)$
 - $\sigma := \{z/a\}$
 - $\varphi := \text{UNIFICAR}(y, b)$
 - $\sigma := \{z/a\}\{y/b\}$
 - $\sigma := \{\{z/a, y/b\}$
- $mgu := \{z/a, y/b\}$

Ejemplo 2

$A := P(f(y), y) \quad B := P(f(u), v)$

- $mgu := \text{UNIFICAR}((P(f(y), y), (P(f(u), v)))$
 - $\varphi := \text{UNIFICAR}(P, P)$
 - $\sigma := \{\}\{\}$
 - $\varphi := \text{UNIFICAR}((f(y), (f(u)))$
 - $\sigma := \{\}\{y/u\}$
 - $A := (P(f(u), u) \quad B := (P(f(u), v)$
 - $\varphi := \text{UNIFICAR}(u, v)$
 - $\sigma := \{y/u\}\{u/v\} = \{y/v, u/v\}$
- $mgu := \{y/v, u/v\}$

Ejercicios

- $\text{UNIFICAR}(P(x), P(a)) = \{x/a\}$
- $\text{UNIFICAR}(P(f(x), y, g(y)), P(f(x), z, g(x))) = \{y/x, z/x\}$
- $\text{UNIFICAR}(P(f(x, g(a)), g(a)), P(f(x, z), z)) = \{z/a\}$

Verificación de Ocurrencia

Ejemplo: $A := P(x, f(a, y))$ y $B := P(x, f(z, b))$

$A := (P \times (f \ a \ y))$ $B := (P \times (f \ z \ b))$

UNIFICAR(A,B)

3. Inicialmente $\sigma := \{\}$
4. Si A y B son expresiones compuestas
 - Para cada par de subexpresiones A_i de A y B_i de B:
 - $\varphi := \text{UNIFICAR}(A_i, B_i)$
 - $A := A\varphi, B := B\varphi, \sigma := \sigma\varphi$
5. En caso contrario:
 - Si $A =$, $\sigma := \{\}$
 - Si A (resp. B) es una constante y no se puede unificar con B (resp. A), entonces retornamos fallo (no hay unificador).
 - Si A (resp. B) es una variable **que no aparece en B** (resp. A), entonces $\sigma := \{A/B\}$ (resp. $\sigma := \{B/A\}$)
6. Retornar(σ)

Verificación de Ocurrencia

Sin verificación de ocurrencia:

$A := P(x, x)$ $B := P(f(z), z)$

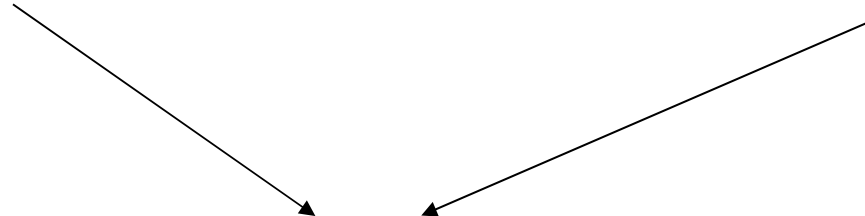
- $mgu := \text{UNIFICAR}((P \ x \ x), (P \ (f \ z) \ z))$
 - $\varphi := \text{UNIFICAR}(P, P)$
 - $\sigma := \{\}$
 - $\varphi := \text{UNIFICAR}(x, (f \ z))$
 - $\sigma := \{x/f(z)\}$
 - $A := (P \ (f \ z) \ (f \ z))$ $B := (P \ (f \ z) \ z)$
 - $\varphi := \text{UNIFICAR}((f \ z), z)$
 - $\sigma := \{x/f(z)\} \{z/f(z)\}$
 - $\sigma := \{x/f(f(z)), z/f(z)\}$
 - $A := (P \ (f \ z) \ (f \ (f \ z)))$ $B := (P \ (f \ z) \ (f \ z))$
- Luego $\text{UNIFICAR}(A, B)$ retorna $\{x/f(f(z)), z/f(z)\}$ que no es un unificador para A y B.

Ejemplo: Mundo de los Bloques

- Sea $KB = \{on(b,c), on(c,d), clear(e), \forall x,y,z.(on(x,y) \wedge on(y,z) \Rightarrow above(x,z))\}$.
- ¿ $KB \models above(b,d)$?
- Transformamos $KB \wedge \neg above(b,d)$ a conjunto de cláusulas:
$$\{(on(b,c)), (on(c,d)), (clear(e)),$$
$$(\neg on(x,y), \neg on(y,z), above(x,z)),$$
$$\neg above(b,d)\}.$$

Ejemplo: Mundo de los Bloques

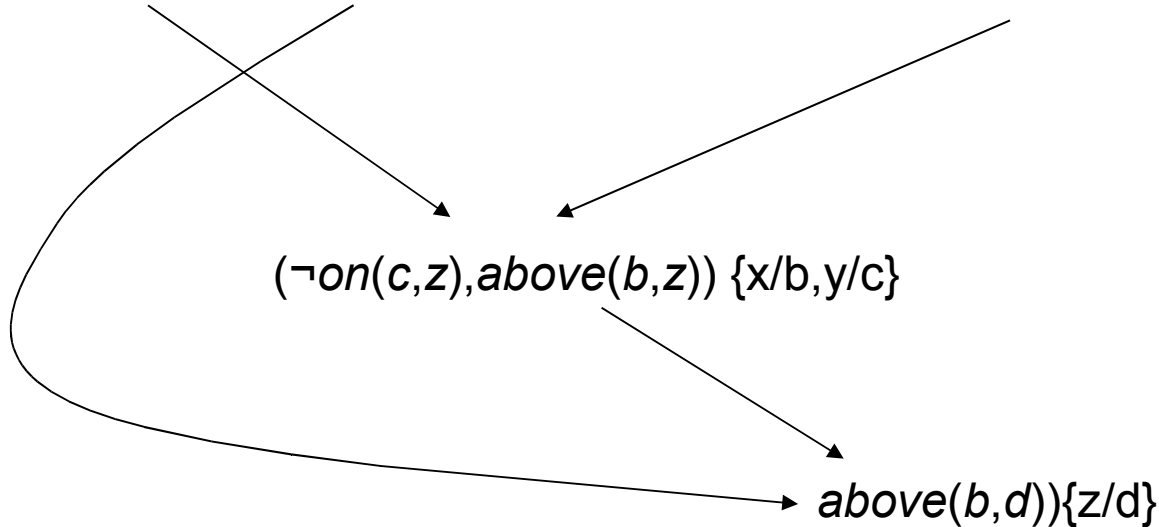
$\{(on(b,c)), (on(c,d)), (clear(e)), (\neg on(x,y), \neg on(y,z), above(x,z)), (\neg above(b,d))\}.$



$(\neg on(c,z), above(b,z)) \{x/b, y/c\}$

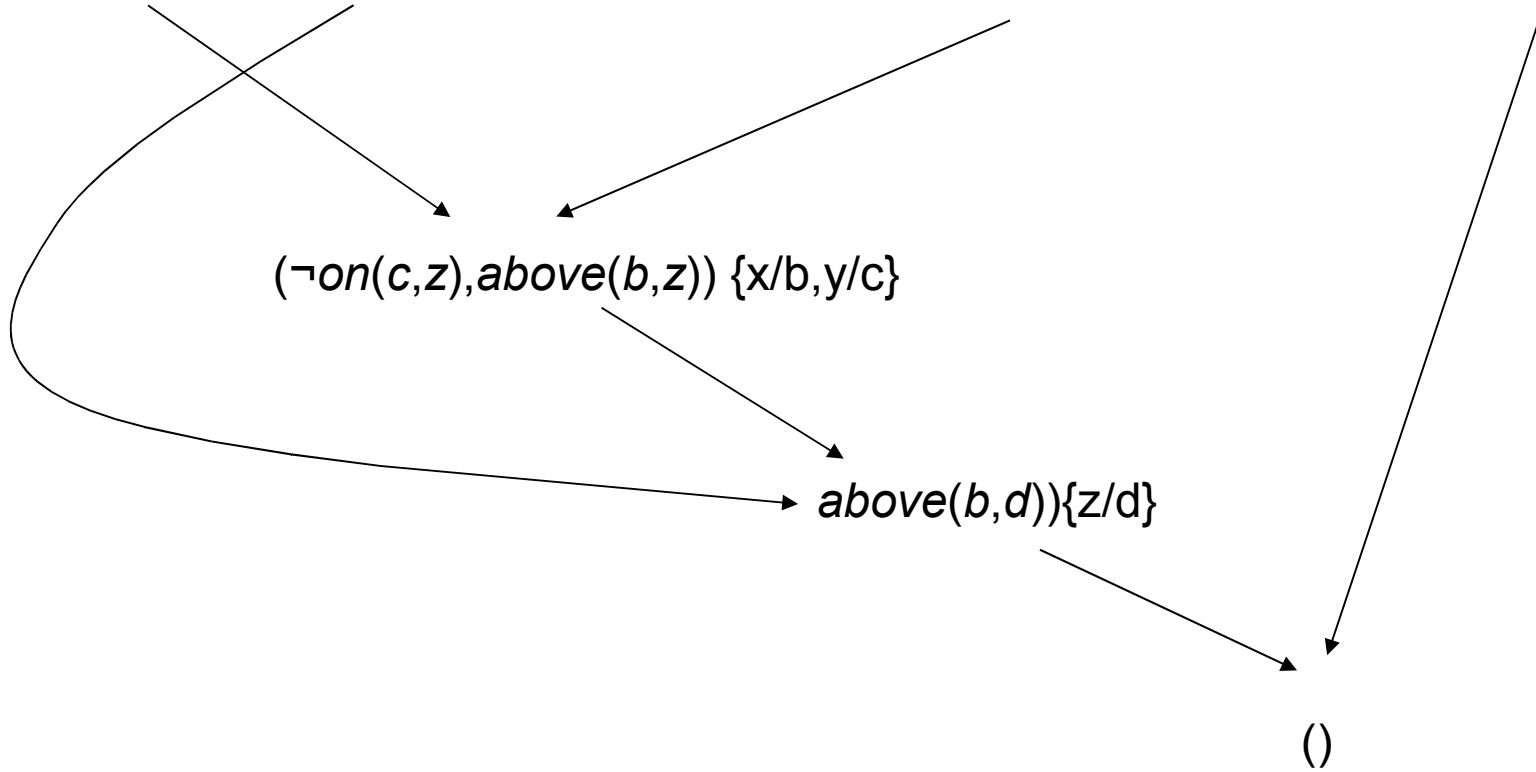
Ejemplo: Mundo de los Bloques

$\{(on(b,c)), (on(c,d)), (clear(e)), (\neg on(x,y), \neg on(y,z), above(x,z)), (\neg above(b,d))\}.$



Ejemplo: Mundo de los Bloques

$\{(on(b,c)),(on(c,d)),(clear(e)),(\neg on(x,y),\neg on(y,z),above(x,z)),(\neg above(b,d))\}.$



Obtención de Respuestas desde una Base de Conocimiento

- Hasta el momento hemos usado resolución para verificar si:

$$KB \models a$$

- También podemos usar resolución para encontrar las constantes que satisfacen una cierta condición que se infiere de KB.
- Ejemplo: Calcular el conjunto
$$\{x \mid KB \models \text{padre}(x, \text{juan})\}$$
- Similar a consultar una base de datos.

Obtención de Respuestas

- Dado un átomo a cuyas variables libres son $x_1, \dots, x_n$
- Queremos encontrar el conjunto
$$\{x_1, \dots, x_n \mid KB \models a\}$$
- Agregamos a la cláusula $\neg a$ el átomo $\text{resp}(x_1, \dots, x_n)$.
- Desde $(KB \wedge \neg a)$, vía resolución intentamos encontrar una cláusulas sólo con átomos $\text{resp}(t_1, \dots, t_n)$.
- Cada una de estas cláusulas se puede ver como una posible respuesta.

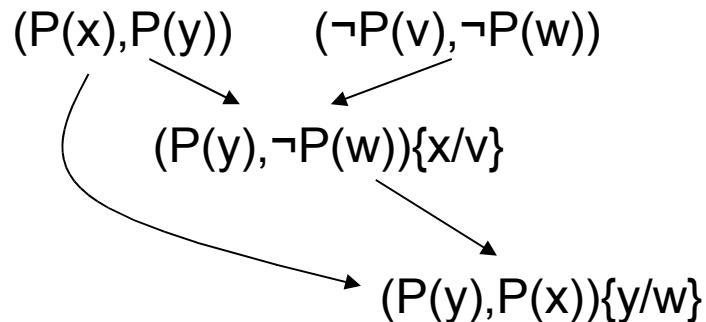
Ejemplo

1. $(\text{father}(\text{art}, \text{jon}), \text{father}(\text{bob}, \text{jon}))$
2. $(\text{father}(\text{bob}, \text{kim}))$
3. $(\neg \text{father}(y, z), \text{parent}(y, z))$
4. $\neg a = (\neg \text{parent}(x, \text{jon}), \text{answer}(x))$
5. $R[4, 3]\{y/z, z/\text{jon}\} (\neg \text{father}(x, \text{jon}), \text{answer}(x))$
6. $R[5, 1]\{x/\text{art}\} (\text{father}(\text{bob}, \text{jon}), \text{answer}(\text{art}))$
7. $R[6, 3]\{y/\text{bob}, z/\text{jon}\}$
 $(\text{parent}(\text{bob}, \text{jon}), \text{answer}(\text{art}))$
8. $R[7, 4] (\text{answer}(\text{bob}), \text{answer}(\text{art}))$

Esta es una respuesta disyuntiva.

Factorización

- Las cláusulas $(P(x), P(y))$ y $(\neg P(v), \neg P(w))$ no son satisfacibles.
- Luego si refutación mediante resolución es completa (para lógica de predicado) deberíamos encontrar la cláusula vacía.
- Sin embargo, esto no sucede:



Factorización

- Si dos o más literales de una cláusula C tienen un UMG σ , entonces $C\sigma$ (con eliminación de literales duplicados) es un factor de C .
- Ejemplo:
 - $C=(P(x),P(f(y)), \neg Q(X))$
 - $\sigma=\{x/f(y)\}$
 - Luego $(P(f(y)), \neg Q(X))$ es un factor de C .
- Regla de resolución se generaliza de modo que $C1$ y $C2$ derivan C ssi existen factores $C1'$ y $C2'$ que resuelven a C .