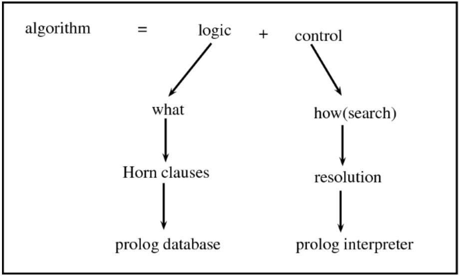
# Tema 5: Resolución




Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Lógica Informática
(Tecnologías Informáticas)

# Contenido

# La regla de resolución

Generaliza algunas de las reglas de inferencia clásicas:

$$\text{Modus Ponens}: \quad \frac{p, \quad p \to q}{q} \qquad \frac{\{p\}, \quad \{\neg p, q\}}{\{q\}}$$

$$\text{Modus Tollens}: \quad \frac{p \to q, \quad \neg q}{\neg p} \qquad \frac{\{\neg p, q\}, \quad \{\neg q\}}{\{\neg p\}}$$

$$\text{Encadenamiento}: \quad \frac{p \to q, \quad q \to r}{p \to r} \qquad \frac{\{\neg p, q\}, \quad \{\neg q, r\}}{\{\neg p, r\}}$$

**Regla de resolución**:

$$\frac{\{L_1, \dots, L, \dots, L_m,\}, \quad \{M_1, \dots, L^c, \dots, M_k\}}{\{L_1, \dots, L_m, M_1, \dots, M_k\}}$$

# Resolución entre cláusulas

**Definición.**
Si $L \in C_1$ y $L' \in C_2$ son literales complementarios, entonces la **resolvente** de $C_1$ y $C_2$ respecto a $L$ es

$$res_L(C_1, C_2) = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{L'\})$$

El conjunto de las resolventes de $C_1$ y $C_2$ es:

$$Res(C_1, C_2) = \{res_L(C_1, C_2) : \; L \in C_1 \text{ y } L^c \in C_2\}.$$

- Ejemplos:
  Sea $C_1 = \{p, q, \neg r\}$ y $C_2 = \{\neg p, r, s\}$. Entonces

$$res_p(C_1, C_2) = \{q, \neg r, r, s\}$$

$$res_{\neg r}(C_1, C_2) = \{p, \neg p, q, s\}$$

# Demostraciones por resolución

Dado un conjunto de cláusulas, $S$, podemos considerar el sistema deductivo que tiene a $Ax(\mathbf{T}) = S$ y usa resolución como única regla de inferencia.

Las definiciones anteriores se adaptan directamente:

- Una **demostración por resolución** a partir de $S$ es una sucesión de cláusulas $C_1, \dots, C_n$ tal que para cada $i = 1, \dots, n$ se verifica:
  - $C_i \in S$, o bien
  - Existen $j, k < i$ tales que $C_i \in Res(C_j, C_k)$.

  Si $C_n = \square$ diremos que $C_1, \dots, C_n$ es una **refutación** de $S$.
- Una cláusula $C$ es **demostrable por resolución** a partir de $S$ si existe una demostración a partir de $S$, $C_1, \dots, C_n$, tal que $C_n = C$.
  Notación: $S \vdash_r C$.
- Decimos que $S$ es **refutable** si $S \vdash_r \square$.

# Ejemplos

Sea $S = \{\{p,q\}, \{\neg p, q\}, \{p, \neg q\}, \{\neg q, \neg p, s\}\}$.

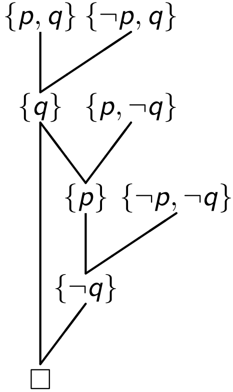
Veamos que $S \vdash_r \{s\}$:

| | | |
|---|---|---|
| 1. | $\{p,q\}$ | Hipótesis |
| 2. | $\{\neg p, q\}$ | Hipótesis |
| 3. | $\{q\}$ | Resolvente de 1 y 2 |
| 4. | $\{\neg q, p\}$ | Hipótesis |
| 5. | $\{p\}$ | Resolvente de 3 y 4 |
| 6. | $\{\neg q, \neg p, s\}$ | Hipótesis |
| 7. | $\{\neg q, s\}$ | Resolvente de 5 y 6 |
| 8. | $\{s\}$ | Resolvente de 7 y 3 |

# Ejemplos (II)

Es habitual presentar las demostraciones por resolución utilizando un árbol.

$S_1 = \{\{p, q\}, \{p, \neg q\}, \{\neg p, q\}, \{\neg p, \neg q\}\}$ es refutable:




Luego $S \vdash_r \Box$.

# Adecuación y Completitud

**Lema**. Sean $C_1$, $C_2$ y $C$ cláusulas. Si $C$ es una resolvente de $C_1$ y $C_2$, entonces $\{C_1, C_2\} \models C$.

**Teorema de adecuación**: Sean $S$ un conjunto de cláusulas y $C$ una cláusula. Entonces

$$S \vdash_r C \quad \Longrightarrow \quad S \models C$$

**Incompletitud de resolución**:

$$\{\{q\}\} \models \{q, r\} \quad \textbf{pero} \qquad \{\{q\}\} \not\vdash_r \{q, r\}$$

**Teorema de completitud de la refutación**:

$S$ es insatisfactible $\quad \Longleftrightarrow \quad S \vdash_r \square$

# Algoritmo de Resolución por Saturación

- **Algoritmo de resolución por saturación**.

  Entrada: $S$, un conjunto finito de cláusulas.

  Salida: **SI**, si $S$ es insatisfactible

  **NO**, en caso contrario.

  Procedimiento:

  1. $S' \leftarrow S$
  2. $S'' \leftarrow S' \cup \bigcup_{C_1, C_2 \in S'} Res(C_1, C_2)$
  3. Mientras $\square \notin S''$ y $S' \neq S''$ hacer:
     - $S' \leftarrow S''$
     - $S'' \leftarrow S' \cup \bigcup_{C_1, C_2 \in S'} Res(C_1, C_2)$
  4. Si $\square \in S''$ devolver **SI** (i.e., insatisfactible)
  5. Si $\square \notin S''$ devolver **NO** (i.e., satisfactible)

# Resolución por saturación

- El algoritmo de resolución por saturación genera una sucesión de conjuntos de cláusulas:

$$S_0 = S, \quad S_{i+1} = S_i \cup \bigcup_{C_1, C_2 \in S_i} Res(C_1, C_2)$$

  de tal modo que para toda cláusula, $C$, se tiene

$$S \vdash_r C \quad \Longleftrightarrow \quad \text{Existe j tal que } C \in S_j$$

- En consecuencia, por el teorema de completitud, el algoritmo de resolución por saturación es correcto (aunque muy ineficiente).

# Ejemplos (I)

- $S = \{\{p, q\}, \{\neg p, q\}, \{p, \neg q\}, \{\neg q\}\}$, aplicando el algoritmo:

$$S_1 = S \cup \{\{q\}, \{p\}, \{\neg p\}, \{\neg p, p\}, \{q, \neg q\}\}$$

$$S_2 = S_1 \cup \{\square, \ldots\}$$

Por tanto, $S$ es **insatisfactible**.

- $S = \{\{p, q\}, \{p, \neg q\}, \{q, r, s\}, \{\neg s, \neg r\}\}$, aplicando el algoritmo:

$$S_1 = S \cup \{\{p\}, \{p, r, s\}, \{q, s, \neg s\}, \{q, r, \neg r\}\}$$

$$S_2 = S_1 \cup \{\{p, s, \neg s\}, \{p, r, \neg r\}, \{q, \neg r, \neg s\}, \{p, q, r, s\}\}$$

$$S_3 = S_2 \cup \{\{p, \neg r, \neg s\}, \{p, q, \neg s, s\}, \{p, q, \neg r, r\}, \{p, q, \neg r, \neg s\}\}$$

Y $S_4 = S_3$. Por tanto, $S$ es **satisfactible**.

# Refinamientos y ejemplos (II)

- **La aparición de tautologías** suele provocar el cálculo de **muchas resolventes repetidas y la aparición de nuevas tautologías**.
- Sin embargo, las tautologías son cláusulas **esencialmente redundantes**, ya que tenemos el siguiente resultado:
  - Si $C \in TAUT$ y $S$ un conjunto de fórmulas entonces
  $$S \text{ es satisfactible} \iff S - \{C\} \text{ es satisfactible}$$
- Por tanto, en cada etapa del algoritmo de saturación podemos **eliminar las tautologías** obtenidas.
- Ejemplo: $S = \{\{p, q\}, \{p, \neg q\}, \{q, r, s\}, \{\neg s, \neg r\}\}$, y aplicando el algoritmo con eliminación de tautologías

$$S_1 = S \cup \{\{p\}, \{p, r, s\}\}, \qquad S_2 = S_1$$

Por tanto, $S$ es **satisfactible**.

# Reducir la redundancia

**Definición.** Dado un conjunto de cláusulas $S$ decimos que una cláusula $C$ **es redundante** si $S$ y $S - \{C\}$ son equiconsistentes.

- **Ejemplo**: Las tautologías son siempre redundantes.
- La **eliminación de cláusulas redundantes reduce el número total** de cláusulas sin alterar la satisfactibilidad del conjunto.
- Una forma no trivial de eliminar cláusulas redundantes se basa en la relación de **subsunción**.

**Definición.** Decimos que una cláusula $C$ **subsume** a $C'$ si $C \subseteq C'$ (como conjuntos de literales).

- **Observación**: Si $C, C' \in S$ y $C$ subsume a $C'$ entonces $C'$ es redundante.

# Resolución regular

- Otra forma de mejorar la eficiencia del algoritmo de resolución consiste en **limitar el cálculo de resolventes** respecto de un mismo literal.

**Definición.**
Un deducción por resolución $C_1, \dots, C_n$ es **regular** si ninguna de sus resolventes $C_j$ contiene un literal respecto del cual se resolvió para calcular alguna de las resolventes $C_i$ previas $(i < j)$.

- Esta restricción (o estrategia) se utiliza en el siguiente **algoritmo para decidir la satisfactibilidad** de un conjunto de cláusulas proposicionales.

# Procedimiento de resolución regular

1. Dado un conjunto de cláusulas $S$, **fijamos una ordenación** $p_1, \ldots, p_k$ de las variables proposicionales que aparecen en $S$.
2. Hagamos $S_0 = S$.
3. Para $j = 1, \ldots, k$, calculamos
   3.1 El conjunto de cláusulas $S'_j$ que se obtiene de $S_{j-1}$ añadiéndole a $S_{j-1}$ todas las resolventes que pueden calcularse respecto de $p_j$ usando cláusulas de $S_{j-1}$.
   3.2 A continuación, $S_j$ se obtiene de $S'_j$ eliminado todas las cláusulas que contengan $p_j$ o $\neg p_j$.
4. Si $\square \in S_k$, entonces $S$ **es insatisfactible**. En caso contrario, $S$ **es satisfactible**.

Se puede demostrar que **el resultado del algoritmo no depende del orden seleccionado** en las variables proposicionales.

# Ejemplos

- Sea $S = \{\{p, q\}, \{\neg p, q\}, \{p, \neg q\}, \{\neg q\}\}$.
  Fijamos el orden $p, q$ para aplicar resolución regular:
  $p$: $S_1' = S_0 \cup \{\{q\}, \{q, \neg q\}\}$, luego
  $$S_1 = \{\{q\}, \{\neg q\}, \{q, \neg q\}\}$$
  $q$: $S_2' = S_1 \cup \{\square, \{q, \neg q\}\}$, luego $S_2 = \{\square\}$.
  Por tanto, $S$ es insatisfactible.
- Sea $S = \{\{p, q\}, \{p, \neg q\}, \{q, r, s\}, \{\neg s, \neg r\}\}$. Fijando el orden $p, q, r, s$
  $p$: $S_1 = \{\{q, r, s\}, \{\neg s, \neg r\}\}$.
  $q$: $S_2 = \{\{\neg s, \neg r\}\}$.
  $r$: $S_3 = \emptyset$.
  $s$: $S_4 = S_3$.
  Por tanto, $S$ es satisfactible.

# Completitud y eficiencia

- Hemos estudiado la **deducción** (por resolución) como un método mecánico para decidir la validez, consistencia, consecuencia lógica, etc.
- La completitud es una propiedad fundamental de los procedimientos de deducción estudiados.
- En el caso de resolución la existencia de una demostración es decidible, y el *conjunto de teoremas* es finito (si el conjunto de cláusulas inicial es finito): usamos para obtenerla *Resolución por Saturación*.
- Sin embargo, los métodos de decisión conocidos **no son eficientes, en general**.

# Eficiencia (I): restringiendo la complejidad sintáctica

- Una solución: Restringir el tipo de cláusulas consideradas.
- Una **cláusula de Horn** es una cláusulas con **a lo sumo un literal positivo**. Ejemplos:
  - $\{p\}$
  - $\{\neg p, \neg q, \neg r\}$
  - $\{r, \neg q, \neg p\}$
- El problema de decidir si un conjunto de cláusulas de Horn (proposicionales) es satisfactible es **decidible de manera eficiente.**

# Eficiencia (II): Estrategias para buscar refutación

Para reducir la explosión combinatoria de aplicación de la regla de resolución se pueden considerar algunas estrategias. **Las siguientes estrategias son refutacionalmente completas**:

- **Resolución positiva**: Sólo se calculan resolventes **si una de las dos cláusulas contiene únicamente literales positivos**.
- **Resolución negativa**: Sólo se calculan resolventes **si una de las dos cláusulas contiene únicamente literales negativos**.
- **Resolución lineal**: Una deducción por resolución a partir de un conjunto $S$, $C_1, \dots, C_n$, es lineal si para cada $i < n$ la cláusula $C_{i+1}$ **es una resolvente de $C_i$ y otra cláusula previamente obtenida por resolución o perteneciente a $S$**.

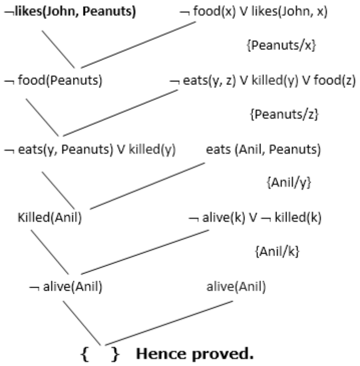
# Eficiencia (III): Estrategias no completas

Las siguientes estrategias **NO son refutacionalmente completas**.

**Pero sí lo son sobre conjuntos de cláusulas de Horn.**

- **Resolución unidad**: Sólo se permiten resolventes si una de las cláusulas es unitaria.
- **Resolución por entradas**: Sólo se permiten resolventes si una de las cláusulas pertenece al conjunto original.

# Resolución en Lógica de Primer Orden



# Resolución en LPO (I): Usando el teorema de Herbrand

- Por el Teorema de Herbrand, un conjunto de cláusulas, $\Sigma$, es inconsistente si y sólo si su extensión de Herbrand, $EH(\Sigma)$, es inconsistente (proposicionalmente).
- Esto proporciona una forma rudimentaria del método de resolución para probar que un conjunto, $\Sigma$, de cláusulas de un lenguaje de primer orden es inconsistente:
  1. Generar $EH(\Sigma)$ y,
  2. Probar que $EH(\Sigma) \vdash_r \square$.
- **Caso importante**: Si no aparecen símbolos de función en las cláusulas, el método de saturación sobre $EH(\Sigma)$ **es un procedimiento adecuado y completo**.

# Resolución en LPO (II)

- El método de resolución en lógica de primer orden **incorpora varias mejoras** al procedimiento anterior:
  1. Es posible generar $EH(\Sigma)$ *poco a poco*, calculando sólo las sustituciones necesarias para obtener cada una de las cláusulas con las que se calculan la resolventes.
  2. **No es necesario restringir el cálculo de resolventes proposicionales a cláusulas sin variables**, y
  3. Para calcular una *resolvente proposicional* entre dos cláusulas, podemos conseguir que ambas cláusulas se obtengan aplicando la misma sustitución a las dos cláusulas de $\Sigma$ (y dicha sustitución **se obtiene algorítmicamente**).

# Ejemplo

- Sea $\Sigma = \{C_1,\ C_2,\ C_3\}$ el conjunto de las cláusulas

$$\begin{array}{ll} C_1: & \neg P(a,y) \lor \neg P(y,z) \lor \neg P(z,y), \\ C_2 & P(x,f(x)) \lor P(a,x), \\ C_3: & P(f(x),x) \lor P(a,x) \end{array}$$

  siendo $a$ una constante.
- Para probar que $\Sigma$ es inconsistente basta probar que su extensión de Herbrand $EH(\Sigma)$ es inconsistente (proposicionalmente).
- Consideremos las siguientes sustituciones:

$$\begin{array}{lcl} \theta_1 & = & \{y/a,\ z/a\} \\ \theta_2 & = & \{y/a,\ z/f(a)\} \\ \theta_3 & = & \{y/f(a),\ z/a\} \\ \theta_4 & = & \{x/a\} \end{array}$$

- Entonces $C_1\theta_1,\ C_1\theta_2,\ C_1\theta_3,\ C_2\theta_4,\ C_3\theta_4 \in EH(\Sigma)$.

# Ejemplo (II)

- Las cláusulas

$$\begin{aligned}
E_1 = C_1\theta_1 &: \quad \neg P(a,a) \\
E_2 = C_1\theta_2 &: \quad \neg P(a,a) \lor \neg P(a,f(a)) \lor \neg P(f(a),a) \\
E_3 = C_1\theta_3 &: \quad \neg P(a,f(a)) \lor \neg P(f(a),a) \\
E_4 = C_2\theta_4 &: \quad P(a,f(a)) \lor P(a,a), \\
E_5 = C_3\theta_4 &: \quad P(f(a),a) \lor P(a,a)
\end{aligned}$$

  son fórmulas abiertas sin variables.
- Veamos que $EH(\Sigma)$ es inconsistente. Para ello probamos que $S = \{E_1, E_2, E_3, E_4, E_5\} \subseteq EH(\Sigma)$ es inconsistente utilizando resolución **proposicional**:

$$\begin{array}{ccccccccc}
E_1 & & & E_3 & & & E_5 & & & E_1 & \\
 & \searrow & & & \searrow & & & \searrow & & & \searrow & \\
E_4 & \rightarrow & P(a,f(a)) & & \rightarrow & \neg P(f(a),a) & & \rightarrow & P(a,a) & & \rightarrow & \square
\end{array}$$

# Ejemplo (III)

- Cada resolvente proposicional de $EH(\Sigma)$ **puede obtenerse directamente a partir de las cláusulas de $\Sigma$** si se toma nota de las sustituciones usadas para generar las cláusulas de $EH(\Sigma)$.
- **Ejemplo**: La resolvente $P(a, f(a))$ obtenida a partir de $E_1$ y $E_4$, puede considerarse obtenida a partir de $C_1$ y $C_2$ usando sustituciones $\theta_1$ y $\theta_4$; gráficamente:

$$\begin{array}{rcccl} C_1 & & & & C_2 \\ \text{sustitución } \theta_1 \downarrow & & & & \downarrow \text{sustitución } \theta_4 \\ C_1\theta_1 & & & & C_2\theta_4 \\ & \searrow & & \swarrow & \\ & & P(a, f(a)) & & \text{(resolvente prop.)} \end{array}$$

- Esta idea es **la base de la resolución en LPO** junto con: **Podemos imponer que $\theta_1 = \theta_4 = \theta$** y existe un algoritmo para calcular dicha sustitución $\theta$.

# Unificadores

- Sea $C$ una cláusula de primer orden y $D$ una subcláusula de $C$ tal que existe una sustitución $\theta$ para la que $D\theta$ se reduce a un único literal $L$. Entonces decimos que
  - La sustitución $\theta$ **unifica** el conjunto de expresiones $D$.
  - $D$ es **unificable** y $\theta$ es un **unificador** de $D$.
  - **Ejemplo**: En $C = \{Q(x,z), P(y,a), P(x,u)\}$ el conjunto
  $$D = \{\ P(y,a), P(x,u)\}$$
  es unificable (tomando $\theta = \{y/x, u/a\}$)
- Para **adaptar el cálculo de resolventes a la LPO**, dadas dos cláusulas $C_1$ y $C_2$ necesitamos encontrar
  - las subcláusulas $D_1 \subseteq C_1$ y $D_2 \subseteq C_2$ **unificables**, y
  - los **unificadores** $\theta_1$ y $\theta_2$ que hacen que $D_1\theta_1$ y $D_2\theta_2$ **se reduzcan a literales complementarios**.

# Unificadores (II): Cómo proceder

- Como se comentó, el problema puede reducirse a la búsqueda de una sola sustitución. Para ello:
  - Podemos suponer que $C_1$ y $C_2$ **no tienen variables comunes** (**condición de variables separadas**).
- Buscamos $D_1 \subseteq C_1$ y $D_2 \subseteq C_2$ tales que $D_1 \cup D_2^c$ (o bien $D_1^c \cup D_2$) satisface:
  - El conjunto es unificable, y
  - sus literales son positivos y tienen el mismo predicado.

  donde $D_i^c$ está formada por los literales complementarios de los literales de $D_i$.

# Ejemplo

- **Consideremos**:

$$C_1 = \{p(x), \neg q(x), p(f(u))\}, \;\; C_2 = \{\neg p(f(f(x))), q(a)\}$$

- **Separamos**:

$$C_1' = \{p(x_1), \neg q(x_1), p(f(u))\}, C_2' = \{\neg p(f(f(x_3))), q(a)\}$$

- **Elegimos**

$$D_1 = \{p(x_1), p(f(u))\}, \quad D_2 = \{\neg p(f(f(x_2)))\}$$

- **Unificamos** $D_1 \cup D_2^c = \{p(x_1), p(f(u)), p(f(f(x_2)))\}$ en

$$(D_1 \cup D_2^c)\theta = \{p(f(f(x_2)))\}$$

usando $\theta = \{x_1/f(f(x_2)), u/f(x_2)\}$

# Unificadores

- Si $D$ es un conjunto de literales, una sustitución $\theta$ es un **unificador de $D$** si $D\theta$ es unitaria.
- Un unificador $\theta$ de $D$ es un **unificador de máxima generalidad** (u.m.g.) si

  Para todo otro unificador $\sigma$ de $D$, existe una sustitución $\alpha$ tal que
  $$\sigma = \theta\alpha$$
  (siendo $\theta\alpha$ la composición de $\theta$ y $\alpha$, es decir, la sustitución obtenida al aplicar **primero** la sustitución $\theta$ y luego $\alpha$).

# ¿Cuál es la **idea**?:

- **Recuerda**: una sustitución siempre produce *casos particulares* (cambia variables por términos).
- **Ejemplo**: en $L_A$, $\sigma = \{x/2 \cdot x\}$ instancia en números pares.
  - $L$ es $\exists y(x = 2 \cdot y)$ [el número $x$ divisible por 2]
  - $L\sigma$ es $\exists y(2 \cdot x = 2 \cdot y)$ [*el número par es divisible por* 2]
- Si $\sigma = \theta\alpha$, entonces $\sigma$ **se puede considerar un caso particular de** $\theta$. Por ejemplo:
  - $\theta = \{x/f(y),\ y/u\}$ es más general que $\{x/f(g(z))\}$ porque
  - para cualquier $C$, de $C\theta$ se puede obtener $C\sigma$ pues una *particularización* de la primera usando otra sustitución:

$$\{x/f(g(z))\} = \{x/y,\ y/u\}\{y/g(z)\}$$

- **Un u.m.g. calcula la unificación más general de los literales, cualquier otra es un caso particular**.

# Ejemplo y cuestión

- Ejemplo de u.m.g. de literales: $\theta_1 = \{y/x, u/a, z/a\}$ es un u.m.g. de

$$P(x, z), P(y, a), P(x, u)$$

y $\theta_2 = \{x/a, y/a, u/a, z/a\}$ es unificador, pero no es un u.m.g.:

$$\theta_2 = \{y/x, u/a, z/a\}\{x/a\}$$

- **Cuestión**: ¿Cómo se puede obtener un u.m.g. (cuando son unificables)?

# Algoritmo para obtener un u.m.g. (I)

- Para unificar dos fórmulas atómicas, $pt_1, \ldots, t_n$ y $pt'_1, \ldots, t'_n$, debemos obtener una sustitución $\theta$ que sea solución de las ecuaciones: $t_1\theta = t'_1\theta, \ldots, t_n\theta = t'_n\theta$.
- Si denotamos por $\approx$ la igualdad sintáctica, es como si nos planteamos un sistema de ecuaciones simbólicas

$$\left\{ \begin{array}{l} t_1 \approx t'_1 \\ \vdots \\ t_n \approx t'_n \end{array} \right. \overset{?}{\rightsquigarrow} \theta$$

- **Para simplificar la notación**, cambiaremos $\approx$ por $=$
- **Para simplificar la descripción del algoritmo**,
  - obtendremos una representación de $\theta$ como $\{x_1 = r_1, \ldots, x_n = r_n\}$ en vez de $\{x_1/r_1, \ldots, x_n/r_n\}$
  - representamos el sistema de ecuaciones como $\{t_1 = t'_1, \ldots, t_n = t'_n\}$

# Algoritmo para obtener un u.m.g. (I)

- **Aplicamos al conjunto $\{t_1 = t'_1, \dots, t_n = t'_n\}$ las siguientes reglas, mientras sea posible**:

  $R_1$: $\{x = x\} \cup E \quad \Longrightarrow \quad E$

  $R_2$: $\{x = t\} \cup E \quad \Longrightarrow \quad \{x = t\} \cup E\{x/t\}$

  cuando $x$ ocurre en $E$ y no en $t$

  $R_3$: $\{t = x\} \cup E \quad \Longrightarrow \quad \{x = t\} \cup E$ si $t$ no es variable.

  $R_4$: $\{f(t_1, \dots, t_n) = f(t'_1, \dots, t'_n)\} \cup E \quad \Longrightarrow$

  $\{t_1 = t'_1, \dots, t_n = t'_n\} \cup E$

  $R_5$: $\{f(t_1, \dots, t_n) = g(t'_1, \dots, t'_n)\} \cup E \quad \Longrightarrow$ FALLO (si $f \neq g$).

  $R_6$: $\{x = t\} \cup E \quad \Longrightarrow$ FALLO si $x$ ocurre en $t$.

- **Si en algún momento obtenemos un conjunto de ecuaciones** $\{x_1 = s_1, x_2 = s_2, \dots, x_r = s_r\}$ al que **no es posible aplicar ninguna regla**, devolvemos UNIFICABLE, y $\theta = \{x_1/s_1,\ x_2/s_2, \dots, x_r/s_r\}$ es un u.m.g.

# Ejemplos

- Unificar
  $\texttt{Divide}(x+(a\cdot y), x\cdot S(y)),\ \texttt{Divide}(S(y)+(y\cdot a), z\cdot z)$

  1. $\{x+(a\cdot y)=(S(y)+(y\cdot a)), x\cdot S(y)=z\cdot z\}$ $[R_4]$
  2. $\{x=S(y), a\cdot y=y\cdot a, x=z, S(y)=z\}$ $[R_4]$
  3. $\{x=S(y), a=y, y=a, x=z, S(y)=z\}$ $[R_4]$
  4. $\{x=S(y), y=a, x=z, z=S(y)\}$ $[R_3]$
  5. $\{\underline{x=S(y)}, y=a, S(y)=z, z=S(y)\}$ $[R_2]$
  6. $\{x=S(a), \underline{y=a}, S(a)=z, z=S(a)\}$ $[R_2]$
  7. $\{x=S(a), y=a, z=S(a)\}$ $[R_3]$

- No es posible unificar
  $\texttt{Divide}(x+(a\cdot y), x\cdot S(y)),\ \texttt{Divide}(S(x)+(y\cdot a), z\cdot z)$
  pues

  1. $\{x+(a\cdot y)=(S(x)+(y\cdot a), x\cdot S(y))=z\cdot z\}$
  2. $\{x=S(x), a\cdot y=y\cdot a, x=z, S(y)=z\}$
  3. FALLO, $x$ ocurre en $S(x)$.

# Ejemplos

- $\mathrm{P}(x + (a \cdot y), x \cdot f(y)),\ \mathrm{P}(f(y) + (y \cdot a), z \cdot z)$
  Unificables:
  1. $\{x + (a \cdot y) = (f(y) + (y \cdot a)), x \cdot f(y) = z \cdot z\}$
  2. $\{x = f(y), a \cdot y = y \cdot a, x = z, f(y) = z\}$
  3. $\{x = f(y), a = y, y = a, x = z, f(y) = z\}$
  4. $\{x = f(y), y = a, x = z, z = f(y)\}$
  5. $[x/f(y)]\{y = a, f(y) = z, z = f(y)\}$
  6. $[x/f(a),\ y/a]\{f(a) = z, z = f(a)\}$
  7. $[x/f(a),\ y/a,\ z/f(a)]\{\}$
- $\mathrm{P}(x + (a \cdot y), x \cdot f(y)),\ \mathrm{P}(f(x) + (y \cdot a), z \cdot z)$
  No unificables:
  1. $\{x + (a \cdot y) = f(x) + (y \cdot a), x \cdot f(y) = z \cdot z\}$
  2. $\{x = f(x), a \cdot y = y \cdot a, x = z, f(y) = z\}$
  3. $[x/f(x)]\{\dots\}$
  4. FALLO, $x$ ocurre en $f(x)$.

# Regla de resolución en Primer Orden

# Resolución

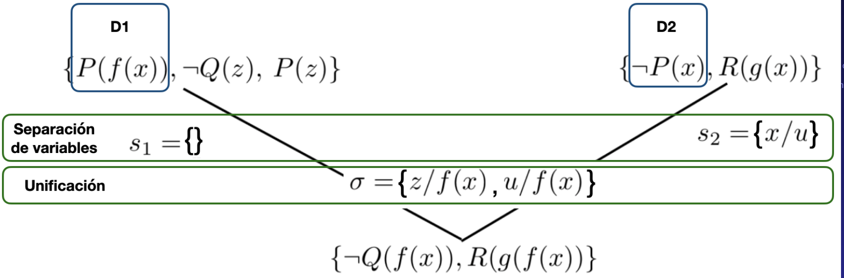
- Un **cambio de variables (o renombramiento)** es una sustitución $\alpha$ que cambia variables $v$ por variables.
- Diremos que $C$ es una **resolvente** de $C_1$ y $C_2$ si existen
  1. (separando variables) dos cambios de variables $\alpha_1$ y $\alpha_2$ tales que

$$C_1\alpha_1 = \{A_1, \ldots, A_n, L_1, \ldots, L_k\}$$
$$C_2\alpha_2 = \{B_1, \ldots, B_m, M_1, \ldots, M_r\}$$

     **no tienen variables comunes**, y
  2. dos subconjuntos $D_1 \subseteq C_1$ y $D_2 \subseteq C_2$, tales que Si $D_1 = \{L_1, \ldots, L_k\}$ y $D_2 = \{M_1, \ldots, M_r\}$, entonces $D_1 \cup D_2^c$ **es unificable** con un u.m.g. $\sigma$.
  3. Se verifica que $C = \{A_1\sigma, \ldots, A_n\sigma, B_1\sigma, \ldots, B_m\sigma\}$
- Es decir,

$$C = (C_1\alpha_1\sigma \setminus D_1\sigma) \cup (C_2\alpha_2\sigma \setminus D_2\sigma)$$

# Ejemplo


D1
D2
$\{P(f(x)), \neg Q(z),\ P(z)\}$
$\{\neg P(x), R(g(x))\}$
Separación de variables
$s_1 = \{\}$
$s_2 = \{x/u\}$
Unificación
$\sigma = \{z/f(x), u/f(x)\}$
$\{\neg Q(f(x)), R(g(f(x)))\}$

# Ejemplos

- Si las cláusulas son cerradas, entonces **una resolvente es una resolvente proposicional**: Si $C_1 = \{p(a), \neg q(b)\}$ y $C_2 = \{\neg p(a), q(b), p(b), r(c)\}$, entonces una resolvente es

$$\{\neg q(b), q(b), p(b), r(c)\}$$

tomando $D_1 = \{p(a)\}$ y $D_2 = \{\neg p(a)\}$

- Si las cláusulas no tienen variables comunes no es necesario renombrar:

$$\frac{\{\underline{P(z,a)}, H(a,z)\}, \quad \{\underline{\neg P(y,x)}, \neg M(y,x)\} \\ \downarrow \sigma = \{z/y, x/a\} \quad \text{u.m.g.}}{\{H(a,y), \neg M(y,a)\}}$$

# Demostraciones por resolución

Dado un conjunto de cláusulas, $\Sigma$, podemos considerar el *sistema deductivo* que tiene a $\Sigma$ como *conjunto de axiomas* y la *resolución* como única regla de inferencia.

- Una **demostración por resolución** a partir de $\Sigma$ es una sucesión de cláusulas $C_1, \ldots, C_n$ tal que para cada $i = 1, \ldots, n$ se verifica:
  - $C_i \in \Sigma$, o bien
  - Existen $j, k < i$ tales que $C_i$ es un resolvente de $C_j$ y $C_k$.

  Si $C_n = \Box$ diremos que $C_1, \ldots, C_n$ es una **refutación** de $\Sigma$.
- Una cláusula $C$ es **demostrable por resolución** a partir de $\Sigma$ si existe una demostración a partir de $\Sigma$, $C_1, \ldots, C_n$, tal que $C_n = C$.
  Notación: $\Sigma \vdash_r C$.
- Decimos que $\Sigma$ es **refutable** si $\Sigma \vdash_r \Box$.

# Ejemplo

$$\Sigma = \{ \neg P(a,y) \vee \neg P(y,z) \vee \neg P(z,y), \; P(x,f(x)) \vee P(a,x), \; P(f(x),x) \vee P(a,x)\}$$

Veamos que $\Sigma \vdash_r \Box$:

1. $\{\neg P(a,y),\ \neg P(y,z),\ \neg P(z,y)\}$ [[Hip.]]
2. $\{P(x,f(x)),\ P(a,x)\}$ [[Hip.]]
3. $\{P(a,f(a))\}$ [[Res(1, 2), $\theta = \{y/a,\ z/a,\ x/a\}$]]
4. $\{\neg P(f(a),a)\}$ [[Res(1, 3), $\theta = \{y/f(a),\ z/a\}$]]
5. $\{P(f(x),x),\ P(a,x)\}$ [[Hip.]]
6. $\{P(a,a)\}$ [[Res(5,4), $\theta = \{x/a\}$]]
7. $\Box$ [[Res(1, 6), $\theta = \{y/a,\ z/a\}$]]

# Otro ejemplo

$\Sigma = \{R(x,y) \vee R(y,z),\ \neg R(x,f(x))\}$

Veamos que $\Sigma \vdash_r \square$.

1. $\{R(x,y),\ R(y,z)\}$ $[\![$Hip.$]\!]$
2. $\{\neg R(u,f(u))\}$ $[\![$Hip. (renombramos)$]\!]$
3. $\{R(f(u),z)\}$ $[\![\text{Res}(1,2),\ \theta = \{x/u, y/f(u)\}]\!]$
4. $\square$ $[\![$ren. $u/v$ en 2, $\text{Res}(2,3),\ \theta = \{v/f(u),\ z/f(f(v))\}]\!]$

# Adecuación y Completitud

- **Teorema de adecuación (corrección)**. Sean $\Sigma$ un conjunto de cláusulas y $C$ una cláusula. Entonces

$$\Sigma \vdash_r C \quad \Longrightarrow \quad \Sigma \models C$$

- **Teorema de completitud de la refutación**:

$$\Sigma \text{ es inconsistente} \quad \Longleftrightarrow \quad \Sigma \vdash_r \Box$$

- La completitud de la resolución **para la refutación** se sigue del teorema de Herbrand y del correspondiente resultado de completitud para la resolución proposicional.

# Ejemplo de refutación

- Ejemplo: En el lenguaje $LC = \{\in\}$, definimos la relación de inclusión como sigue:

$$H := \ \forall x \forall y (x \subseteq y \leftrightarrow \forall w (w \in x \to w \in y))$$

Veamos cómo se prueba que la relación $\subseteq$ es transitiva;

$$C := \forall x \forall y \forall z (x \subseteq y \land y \subseteq z \to x \subseteq z)$$

- Paso 1: Paso a forma clausal de $H$ y $\neg C$:

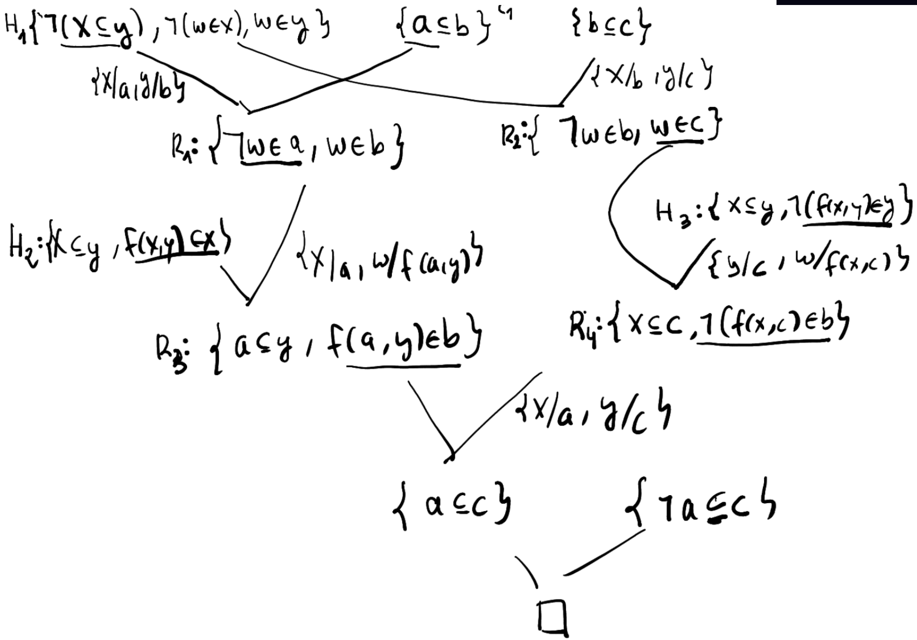
$H_1$ : $\{\neg(x \subseteq y), \neg(w \in x), w \in y\}$ $[' \to '$ de $H]$
$H_2$ : $\{x \subseteq y, f(x,y) \in x\}$ $[' \to '$ de $H]$
$H_3$ : $\{x \subseteq y, \neg(f(x,y) \in y)\}$ $[' \to '$ de $H]$
$C_1$ : $a \subseteq b$ $[a, b, c$ nuevas constantes, $\neg C]$
$C_2$ : $b \subseteq c$ $[\neg C\ ]$
$C_3$ : $\neg(a \subseteq c)$ $[\neg C]$

# Ejemplo de refutación (II)

- Una refutación es (sin llaves):

| | | |
|---|---|---|
| $R_1$: | $\neg(w \in a), w \in b$ | $[H_1, 1 \text{ con } C_1, 1]$ $[\{x/a, y/b\}]$ |
| $R_2$: | $\neg(w \in b), w \in c$ | $[H_1, 1 \text{ con } C_2, 1]$ $[\{x/b, y/c\}]$ |
| $R_3$: | $a \subseteq y, f(a, y) \in b$ | $[H_2, 2 \text{ con } R_1, 1]$ $[\{x/a, w/f(a, y)\}]$ |
| $R_4$: | $x \subseteq c, \neg(f(x, c) \in b)$ | $[H_3, 2 \text{ con } R_2, 2]$ $[\{y/c, w/f(x, c)\}]$ |
| $R_5$: | $a \subseteq c$ | $[R_3, 2 \text{ con } R_4, 2]$ $[\{x/a, y/c\}]$ |
| $R_6$: | $\square$ | $[R_5, 1 \text{ con } C_3, 1]$ |

# Gráficamente



# Caso particular: cláusulas en Lógica predicativa (sin símbolos de función ni igualdad)

- Como en este caso el universo de Herbrand es finito, entonces el problema es decidible
- Se puede aplicar **saturación proposicional**, pero no es necesario calcular todas las cláusulas, basta hacer todas las posibles resolventes (que es una cantidad finita)
- **Ejemplo 1**: $\{\forall x\ [P(x) \to Q(x)], \exists x\ P(x)\} \models \exists x\ Q(x)$ se reduce a

  $\{\{\neg P(x), Q(x)\}, \{P(a)\}, \{\neg Q(z)\}\}$ es inconsistente.

  Demostración:

  | | | |
  |---|---|---|
  | 1 | $\{\neg P(x), Q(x)\}$ | Hipótesis |
  | 2 | $\{P(a)\}$ | Hipótesis |
  | 3 | $\{\neg Q(z)\}$ | Hipótesis |
  | 4 | $\{Q(a)\}$ | Resolvente de 1 y 2 con $\sigma = [x/a]$ |
  | 5 | $\square$ | Resolvente de 3 y 4 con $\sigma = [z/a]$ |

# Decisión de no–consecuencia por resolución

- Enunciado: Comprobar, por resolución, que
  $\forall x\, [P(x) \vee Q(x)] \not\models \forall x\, P(x) \vee \forall x\, Q(x)$.

  Reducción 1: Comprobar que es consistente
  $\{\forall x\, [P(x) \vee Q(x)],\ \neg(\forall x\, P(x) \vee \forall x\, Q(x))\}$

  Reducción 2: Comprobar que es consistente
  $\{\{P(x), Q(x)\}, \{\neg P(a)\}, \{\neg Q(b)\}\}$
- Resolución:

| | | |
|---|---|---|
| 1 | $\{P(x), Q(x)\}$ | Hipótesis |
| 2 | $\{\neg P(a)\}$ | Hipótesis |
| 3 | $\{\neg Q(b)\}$ | Hipótesis |
| 4 | $\{Q(a)\}$ | Resolvente de 1 y 2 |
| 5 | $\{P(b)\}$ | Resolvente de 1 y 3 |

  Hemos **saturado**, no podemos hacer más resolventes
- Modelo: $U = \{a, b\}, I(P) = \{b\}, I(Q) = \{a\}$.

# Estrategias

Al igual que en el caso proposicional, para reducir la explosión combinatoria se pueden considerar estrategias:

- **Resolución positiva**: Sólo se calculan resolventes si una de las dos cláusulas contiene únicamente literales positivos.
- **Resolución negativa**: idem con cláusulas negativas.
- **Resolución lineal**: Para cada $i < n$ la cláusula $C_{i+1}$ es una resolvente de $C_i$ y otra cláusula previamente obtenida por resolución o perteneciente a $S$.
- **Resolución unidad**: Sólo se permiten resolventes si una de las cláusulas es unitaria.
- **Resolución por entradas**: Sólo se permiten resolventes si una de las cláusulas pertenece al conjunto original.

# Razonamiento con igualdad

- Podemos utilizar el método de resolución para razonar con un LPO **con igualdad**. Para ello basta añadir axiomas que expresen las propiedades esenciales del predicado de igualdad.
- Fijado un LPO con igualdad, $L$, denotaremos por $\mathrm{EQ}(L)$ al conjunto formado por las siguientes cláusulas:
  - $x = x$.
  - $x \neq y \vee y = x$.
  - $x \neq y \vee y \neq z \vee x = z$.
  - Para cada símbolo de predicado $P$ de $L$ (de aridad $n$)
  $$x_j \neq x_0 \vee \neg P(x_1, \dots, x_j, \dots x_n) \vee P(x_1, \dots, x_0, \dots, x_n)$$
  - Para cada símbolo de función $f$ de $L$ (de aridad $n$)
  $$x_j \neq x_0 \vee f(x_1, \dots, x_j, \dots x_n) = f(x_1, \dots, x_0, \dots, x_n)$$
- Si $S$ es un conjunto de cláusulas entonces
$$S \text{ es insatisfactible} \iff S \cup \mathrm{EQ}(L) \text{ es refutable}$$

# Paramodulación

- Otra posibilidad es integrar el razonamiento con igualdad en el cálculo de resolventes añadiendo una nueva regla específica para el tratamiento de las ecuaciones.
- La regla de **paramodulación** permite obtener un nueva cláusula $C$ (una paramodulante) a partir de dos cláusulas $C_1$ y $C_2$ de un modo parecido a cálculo de resolventes.

# Bibliografía


1. Fitting, M. *First-Order Logic and Automated Theorem Proving (2nd ed.)* (Springer, 1996) pp. 137–141.
2. C.L. Chang y R.C.T. Lee *Symbolic logic and mechanical theorem proving* (Academic Press, 1973) pp. 70–99.
3. M. Ojeda e I. Pérez *Lógica para la computación (Vol. 2: Lógica de Primer Orden)* (Ágora, 1997) pp. 138–164.
4. L. Paulson, *Logic and proof* http://www.cl.cam.ac.uk/Teaching/2002/LogicProof/notes.pdf (U. Cambridge, 2002) pp. 50–61.
5. U. Schöning *Logic for computer scientists* (Birkäuser, 1989) pp. 79–96.