Tema 5: Sistemas Deductivos y Resolución

Dpto. Ciencias de la Computación Inteligencia Artificial
Universidad de Sevilla

Lógica Informática
(Tecnologías Informáticas)
Curso 2019–20

Contenido

Pruebas formales

- Los tableros semánticos proporcionan un algoritmo para la deducción basado en este hecho:

$$\Sigma \models A \quad \Longleftrightarrow \quad \Sigma \cup \{\neg A\} \ \textbf{insatisfactible}$$

- Un enfoque más natural del problema básico **(PB)** que presentamos en el Tema 1, se obtiene a través de la noción de **demostración**:
  1. Consideramos el conjunto de enunciados, $\mathcal{BC}$, como un conjunto de **axiomas** (o **hipótesis** que asumimos como ciertas inicialmente).
  2. El enunciado ϕ será consecuencia de $\mathcal{BC}$ si podemos obtener una **demostración** de ϕ a partir de $\mathcal{BC}$ (de manera similar a como en matemáticas se demuestra un teorema).

Sistemas deductivos

Un **sistema deductivo**, **T**, (o teoría proposicional) consta de:

- Un conjunto, $Ax(\mathbf{T})$, de fórmulas proposicionales que llamamos los axiomas de **T**.
- Reglas de inferencia de la forma:

$$\frac{A_1, \ldots, A_n}{A}$$

siendo $A_1, \ldots, A_n, A$ fórmulas proposicionales. Las fórmulas $A_1, \ldots, A_n$ se denominan *premisas* y la fórmula A *conclusión*.

Demostraciones y Teoremas

Definición.
Una **demostración** en **T** es una sucesión de fórmulas proposicionales $A_1, \ldots, A_k$ cada una de las cuales es un axioma de **T**, o bien, se obtiene a partir de fórmulas anteriores de la sucesión mediante la aplicación de una regla de inferencia.

Definición.
Una fórmula A es un **teorema** de **T**, $\vdash_{\mathbf{T}} A$, si existe una demostración en **T**, $A_1, \ldots, A_k$ tal que $A = A_k$. La sucesión $A_1, \ldots, A_k$ se denomina una **demostración de** A en **T**.

Ejemplo: Sistema deductivo

- $Ax(\mathbf{T}) = \{p, q, p \wedge q \rightarrow (\neg s \vee p \rightarrow r)\}$
- Reglas de inferencia: (A y B fórmulas cualesquiera)

$$I_\wedge : \quad \frac{A,\ B}{A \wedge B} \qquad I_\vee : \quad \frac{A}{A \vee B}$$

$$C_\vee : \quad \frac{A \vee B}{B \vee A} \qquad MP : \quad \frac{A,\ A \rightarrow B}{B}$$

La siguiente sucesión es una demostración de r en $\mathbf{T}$ ($\vdash_{\mathbf{T}} r$):

1.	p	[[Hip.]]
2.	q	[[Hip.]]
3.	$p \wedge q$	[[$I_\wedge$ aplicada a 1. y 2.]]
4.	$p \wedge q \rightarrow (\neg s \vee p \rightarrow r)$	[[Hip.]]
5.	$\neg s \vee p \rightarrow r$	[[MP aplicada a 3. y 4.]]
6.	$p \vee \neg s$	[[$I_\vee$ aplicada a 1.]]
7.	$\neg s \vee p$	[[$C_\vee$ aplicada a 6.]]
8.	r	[[MP aplicada a 7. y 5.]]

Procedimientos de demostración

- Un sistema deductivo introduce una noción de prueba formal, pero no proporciona ningún procedimiento efectivo para generar demostraciones de las fórmulas que deseamos probar.
- Un **sistema de razonamiento** proporciona, además de un sistema deductivo, un **procedimiento de demostración**, que implementa una estrategia de búsqueda de demostraciones.
- En general, no podemos encontrar estrategias que sean efectivas para todos los sistemas deductivos, y habrá que buscar estrategias que funcionen solo para casos particulares.
- Ilustraremos un ejemplo de estrategia eficiente en el caso del encadenamiento con cláusulas de Horn.

Cláusulas de Horn

- Una **cláusula de Horn** es una disyunción de literales que contiene a lo sumo un literal positivo.
  - Ejemplos: $\neg p \vee \neg q \vee r, \quad \neg p \vee \neg r, \quad p$
- Una cláusula de Horn **positiva** es una cláusula que contiene exactamente un literal positivo.
  - Si contiene otros literales negativos se denomina **regla**. Ejemplo: $\neg p \vee \neg q \vee r$.
  - Si sólo contiene el literal positivo se denomina **hecho**. Ejemplo: q.

La regla $\neg p \vee \neg q \vee r$ se escribe como

$$\underbrace{p \wedge q}_{cuerpo} \rightarrow \underbrace{r}_{cabeza}$$

Deducción con cláusulas de Horn

Consideraremos el siguiente sistema deductivo, $EA(\Sigma)$, para trabajar con cláusulas de Horn positivas.

- **Axiomas**: Un conjunto finito, Σ, de cláusulas de Horn positivas.
- **Reglas de inferencia**: Si A y B son conjunciones de literales, entonces:

$$I_\wedge : \quad \frac{A,\ B}{A \wedge B} \qquad MP : \quad \frac{A,\ A \to B}{B}$$

Teorema. Se tiene que, para todo **hecho**, H

$$\Sigma \models H \quad \Longleftrightarrow \quad \vdash_{EA(\Sigma)} H$$

- Además, podemos diseñar un algoritmo de decisión eficiente para demostrar hechos en $EA(\Sigma)$, que se denomina **encadenamiento hacia adelante**.

Ejemplo: deducción en $EA(\Sigma)$

$$\Sigma = \{P \to Q,\ L \land M \to P,\ B \land L \to M,\ A \land P \to L,\ A \land B \to L,\ A,\ B\}$$

Veamos que $\vdash_{EA(\Sigma)} Q$,

1.	A	$[\![$Hip.$]\!]$
2.	B	$[\![$Hip.$]\!]$
3.	$A \land B$	$[\![I_\land$ aplicada a 1. y 2..$]\!]$
4.	$A \land B \to L$	$[\![$Hip.$]\!]$
5.	L	$[\![$MP aplicada a 3. y 4.$]\!]$
6.	$B \land L$	$[\![I_\land$: 2. y 5.$]\!]$
7.	$B \land L \to M$	$[\![$Hip.$]\!]$
8.	M	$[\![$MP: 6. y 7.$]\!]$
9.	$L \land M$	$[\![I_\land$: 5. y 8.$]\!]$
10.	$L \land M \to P$	$[\![$Hip.$]\!]$
11.	P	$[\![$MP: 9. y 10.$]\!]$
12.	$P \to Q$	$[\![$Hip.$]\!]$
13.	Q	$[\![$MP: 11. y 12.$]\!]$

Procedimiento de encadenamiento hacia adelante

- Entrada:
  - Un hecho H.
  - Una enumeración $R_1, \ldots, R_n$ de los elementos de Σ.
- Salida:
  - SI, si $\vdash_{EA(\Sigma)} H$.
  - NO, en caso contrario.
- Inicialmente, C es el conjunto de todos los hechos que pertenecen a Σ, $j = 1$, $i = 0$.
  1. Si $H \in C$, paramos y devolvemos SI.
  2. Si $j \leq n$ y todos los literales del cuerpo de R_j están en C, entonces
     - 2.1 Si la cabeza de j no está en C se la añadimos a C, hacemos $j = j + 1$, $i = 1$ y volvemos a 1.
     - 2.2 Si la cabeza de j está en C hacemos $j = j + 1$ y volvemos a 2.
  3. Si $j \leq n$ y algún literal del cuerpo de R_j no está en C, entonces hacemos $j = j + 1$ y volvemos a 2.
  4. Si $j > n$ y $i = 1$, hacemos $j = 1$ y $i = 0$ y volvemos a 2.
  5. Si $j > n$ y $i = 0$, paramos y devolvemos NO.

Ejemplo: encadenamiento hacia adelante

Veamos que $\vdash_{EA(\Sigma)} Q$,

Hechos obtenidos	**Reglas usadas**
A, B	
A, B, L	$A \wedge B \to L$
A, B, L, M	$A \wedge B \to L,\ B \wedge L \to M$
A, B, L, M, P	$A \wedge B \to L,\ B \wedge L \to M,\ L \wedge M \to P$
A, B, L, M, P, Q	Σ

$$\Sigma = \{ \quad P \to Q,\ L \wedge M \to P,\ B \wedge L \to M,\ A \wedge P \to L,\ A \wedge B \to L,\ A,\ B \}$$

Deducción con cláusulas de Horn (II)

Consideremos otrs sistema deductivo, $EB(\Sigma)$, para trabajar con cláusulas de Horn positivas.

- Axiomas: Un conjunto finito, Σ de cláusulas de Horn positivas.
- Reglas de inferencia: (A y C pueden ser conjunciones vacías)

$$SR: \quad \frac{A \wedge L \wedge L \wedge C \to D}{A \wedge L \wedge C \to D} \qquad CR: \quad \frac{A \wedge L_1 \wedge L_2 \wedge C \to D}{A \wedge L_2 \wedge L_1 \wedge C \to D}$$

$$BC: \quad \frac{A \to B,\ B \wedge C \to D}{A \wedge C \to D}$$

Teorema. Se tiene que, para todo **hecho**, H

$$\Sigma \models H \iff \vdash_{EB(\Sigma)} H$$

Deducción en $EB(\Sigma)$

Veamos que $\vdash_{EB(\Sigma)} Q$,

1.	$P \to Q$	$[\![$Hip.$]\!]$
2.	$L \wedge M \to P$	$[\![$Hip.$]\!]$
3.	$L \wedge M \to Q$	$[\![$$BC$ aplicada a 1. y 2.$]\!]$
4.	$M \wedge L \to Q$	$[\![$$CR$ aplicada a 3.$]\!]$
5.	$B \wedge L \to M$	$[\![$Hip.$]\!]$
6.	$B \wedge L \wedge L \to Q$	$[\![$$BC$ aplicada a 4. y 5.$]\!]$
7.	$B \wedge L \to Q$	$[\![$$SR$ aplicada a 6.$]\!]$
8.	B	$[\![$Hip.$]\!]$
9.	$L \to Q$	$[\![$$BC$: 7. y 8.$]\!]$
10.	$A \wedge B \to L$	$[\![$Hip.$]\!]$
11.	$A \wedge B \to Q$	$[\![$$BC$ aplicada a 9. y 10.$]\!]$
12.	A	$[\![$Hip.$]\!]$
13.	$B \to Q$	$[\![$$BC$ aplicada a 11. y 12.$]\!]$
14.	Q	$[\![$$BC$ aplicada a 13. y 8.$]\!]$

$\Sigma = \{P \to Q,\ L \wedge M \to P,\ B \wedge L \to M,\ A \wedge P \to L,\ A \wedge B \to L,\ A,\ B\}$

Procedimiento de encadenamiento hacia atrás

- Entrada:
  - Un hecho H.
  - Una enumeración $R_1, \dots, R_n$ de los elementos de Σ.
- Salida:
  - SI, si $\vdash_{EBH(\Sigma)} H$.
  - NO, en caso contrario.
- Inicialmente $G = (H)$ (objetivos pendientes) y $j = 1$.
  1. Si $j \leq n$ y G es vacía, paramos y devolvemos SI.
  2. Si $j \leq n$ y G no es vacía, denotamos por E el primer elemento de G y actualizamos j y G como sigue:
     2.1 Si E es la cabeza de R_j actualizamos G sustituyendo el primer elemento, E, por la sucesión de los literales del cuerpo de R_j, hacemos $j = 1$ y volvemos a 1.
     2.2 Si la cabeza de R_j no es E, hacemos $j = j + 1$ y volvemos a 2.
  3. Si $j > n$ y $G = (H)$, paramos y devolvemos NO.
  4. Si $j > n$, $G \neq (H)$ y E es el primer elemento de G, damos a G y j sus valores anteriores a la aplicación de la regla que, según 2.1, introdujo el literal E en G. Hacemos $j = j + 1$ y volvemos a 2.

Ejemplo: encadenamiento hacia atrás

Veamos que $\vdash_{\mathbf{EB}(\Sigma)} Q$,

Objetivos	**Reglas usadas**
Q	
P	$P \to Q$
L, M	$L \wedge M \to P$
A, B, M	$A \wedge B \to L$
B, M	A
M	B
B, L	$B \wedge L \to M$
L	B
A, B	$A \wedge B \to L$
B	A
–	B

$$\Sigma = \{P \to Q,\ L \wedge M \to P,\ B \wedge L \to M,\ A \wedge B \to L,\ A \wedge P \to L,\ A,\ B\}$$

La regla de resolución

Generaliza algunas de las reglas de inferencia clásicas:

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

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

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

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 $C_n = \Box$ 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 $S \vdash_r \Box$.

Ejemplos

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

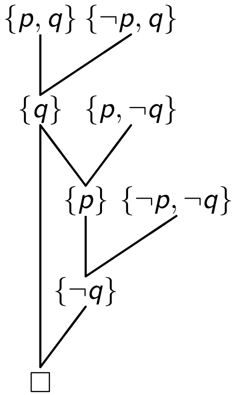
Veamos que $S \vdash_r \{s\}$:

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 $S \vdash_r \square$.

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:

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

Teorema de completitud de la refutación:

$$S \text{ es insatisfactible} \quad \Longleftrightarrow \quad S \vdash_r \Box$$

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

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

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

Ejemplos (I)

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

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

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.

- Por 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 es limitar el cálculo de resolventes respecto de un mismo literal.

Definición.
Un deducción por resolución $C_1, \ldots, C_n$ 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, fijemos 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 probar 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: $S_2' = S_1 \cup \{\Box, \{q, \neg q\}\}$, luego $S_2 = \{\Box\}$.

  Por tanto, S es insatisfactible.
- Sea $S = \{\{p, q\}, \{p, \neg q\}, \{q, r, s\}, \{\neg s, \neg r\}\}$.
  Aplicando el algoritmo para la ordenación p, q, r, s obtenemos

  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 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): *Resolución por Saturación*.
- Sin embargo, los métodos de decisión conocidos no son eficientes, en general.
- Una solución: Restringir el tipo de cláusulas consideradas.
  - Una **cláusula de Horn** es una cláusulas con a lo sumo un literal positivo.
  - El problema de decidir si un conjunto de cláusulas de Horn (proposicionales) es satisfactible es decidible de manera eficiente.

Estrategias

Para reducir la explosión combinatoria de aplcación de la regla de resolución se pueden considerar algunas estrategias.

- **Resolución positiva** (resp. negativa): Sólo se calculan resolventes si una de las dos cláusulas contiene únicamente literales positivos (resp. negativos). Es (refutacionalmente) completa.
- **Resolución lineal**: Una deducción por resolución a partir de un conjunto S, $C_1, \ldots, C_n$, 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.
  - La resolución lineal es (refutacionalmente) completa:
  - **Teorema**. Si S es un conjunto insatisfactible y $S - \{C\}$ es satisfactible, entonces existe una refutación de S por resolución lineal cuya cláusula inicial es C.

Estrategias (II)

- **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 a S.
  - En general, resolución unidad y por entradas **NO** son refutacionalmente completas, pero sí lo son restringidas a conjuntos de cláusulas de Horn.
  - **Teorema**. Sea S es un conjunto insatisfactible formado por cláusulas de Horn. Entonces S es refutable mediante resolución unidad y mediante resolución por entradas.

Resolución en LPO

- Por el Teorema de Herbrand, un conjunto de cláusulas, Σ, 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, Σ, de cláusulas de un lenguaje de primer orden es inconsistente:
  1. Generar $EH(\Sigma)$ y,
  2. Probar que $EH(\Sigma) \vdash_r \square$.
- El método de resolución en lógica de primer orden incorpora varias mejoras en este procedimiento básico:
  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 dos cláusulas de Σ (y dicha sustitución se obtiene algorítmicamente).

Ejemplo

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

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

siendo a una constante.

- Para probar que Σ 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

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

  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 entre cláusulas de $EH(\Sigma)$ puede obtenerse directamente a partir de las cláusulas de Σ si se toma nota de las sustituciones utilizadas 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 utilizando las sustituciones θ_1 y θ_4.
- Gráficamente,

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

Unificadores

- Un problema del cálculo de las resolventes que acabamos de ver es encontrar las subcláusulas $D_1 \subseteq C_1$ y $D_2 \subseteq C_2$ y las sustituciones θ_1 y θ_2 que hacen que $D_1\theta_1$ y $D_2\theta_2$ se reduzcan a literales complementarios.
- Sea C una cláusula de primer orden y D una subcláusula de C tal que existe una sustitución θ para la que $D\theta$ se reduce a un único literal L. Entonces decimos que
  - La sustitución θ **unifica** el conjunto de expresiones D.
  - D es **unificable** y θ es un **unificador** de D.
- El problema puede reducirse a la búsqueda de una sola sustitución. Para ello:
  1. Podemos suponer que C_1 y C_2 no tienen variables comunes (**condición de variables separadas**).
  2. 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 que:
     - Es unificable, y
     - sus literales son positivos y tienen el mismo predicado.

     (D_i^c está formada por los literales complementarios de los literales de D_i).

Unificadores

- Si D es un conjunto de literales, una sustitución θ es un unificador de D si $D\theta$ es unitaria.
- Un unificador θ de D es un **unificador de máxima generalidad** (u.m.g.) si
  - Para todo unificador σ de D, existe una sustitución α tal que $\sigma = \theta\alpha$.
    (siendo $\theta\alpha$ la composición de θ y α, es decir, la sustitución obtenida al aplicar primero la sustitución θ y luego α).
- 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.:

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

Algoritmo para obtener un u.m.g.

- Para unificar $pt_1, \ldots, t_n$ y $pt'_1, \ldots, t'_n$, debemos obtener una sustitución θ que sea solución de las ecuaciones: $t_1\theta = t'_1\theta, \ldots, t_n\theta = t'_n\theta$.
- Para ello aplicamos al conjunto $\{t_1 = t'_1, \ldots, t_n = t'_n\}$ 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: $\{f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)\} \cup E \quad \Longrightarrow$
  $\{t_1 = t'_1, \ldots, t_n = t'_n\} \cup E$

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

  R_6: $\{x = t\} \cup E \quad \Longrightarrow \quad$ FALLO si x ocurre en t.
- Si en algún paso obtenemos un conjunto de ecuaciones $\{x_1 = s_1, x_2 = s_2, \ldots, x_r = s_r\}$ al que no es posible aplicar ninguna regla, devolvemos UNIFICABLE, y $\theta = \{x_1/s_1,\ x_2/s_2, \ldots, x_r/s_r\}$ es un u.m.g.

Ejemplos

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

  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
  $\mathtt{Divide}(x + (a \cdot y), x \cdot S(y)),\ \mathtt{Divide}(S(x) + (y \cdot a), z \cdot z)$
  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 $f(x)$.

Resolución

- Un cambio de variables (o renombramiento) es una sustitución α tal que para toda variable v, $\alpha(v)$ es una variable.
- Decimos que C es una **resolvente** de las cláusulas C_1 y C_2 si existen dos cambios de variables α_1 y α_2 tales que

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

  no tienen variables comunes y
  - Si $D_1 = \{L_1, \dots, L_k\}$ y $D_2 = \{M_1, \dots, M_r\}$, entonces $D_1 \cup D_2^c$ es unificable con u.m.g. σ.
  - $C = \{A_1\sigma, \dots, A_n\sigma, B_1\sigma, \dots, B_m\sigma\}$
- Es decir,

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

Ejemplos

- Si las cláusulas son cerradas, una resolvente es una resolvente proposicional.
- 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, Σ, podemos considerar el sistema deductivo que tiene a Σ como conjunto de axiomas y resolución como única regla de inferencia.

- Una **demostración por resolución** a partir de Σ 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 Σ.
- Una cláusula C es **demostrable por resolución** a partir de Σ si existe una demostración a partir de Σ, $C_1, \ldots, C_n$, tal que $C_n = C$.
  Notación: $\Sigma \vdash_r C$.
- Decimos que Σ 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. $\neg P(a,y) \vee \neg P(y,z) \vee \neg P(z,y)$ ⟦Hip.⟧
2. $P(x,f(x)) \vee P(a,x)$ ⟦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. $P(f(x),x) \vee P(a,x)$ ⟦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. $R(x,y) \vee R(y,z)$ $\quad$ $[\![\text{Hip.}]\!]$
2. $\neg R(u, f(u))$ $\quad$ $[\![\text{Hip. (renombramos)}]\!]$
3. $R(f(u), z)$ $\quad$ $[\![\text{Res}(1,2),\ \theta = \{x/u, y/f(u)\}]\!]$
4. $\square$ $\quad$ $[\![$ren. u/v en 2, $\text{Res}(2,3)$, $\theta = \{v/f(u),\ z/f(f(v))\}]\!]$

Adecuación y Completitud

- **Teorema de adecuación**. Sean Σ un conjunto de cláusulas y C una cláusula. Entonces

$$\Sigma \vdash_{ur} C \quad \Longrightarrow \quad \Sigma \models C$$

- Resolución es un caso particular de resolución no restringida, luego si $\Sigma \vdash_r C$ entonces $\Sigma \models C$.
- Tanto resolución como resolución no restringida son incompletas. Pero la completitud de la resolución no restringida **para la refutación** se sigue del teorema de Herbrand y del correspondiente resultado de completitud para la resolución proposicional. Además se tiene,
- **Lema**. Si una cláusula C es una resolvente no restringida de C_1 y C_2, repecto de las subcláusulas D_1 y D_2, entonces C también es una resolvente de C_1 y C_2 con respecto a D_1 y D_2.
- **Teorema de completitud de la refutación**:

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

Ejemplo de refutación

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

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

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

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

- 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:	$\square$	$[R_5, 1 \text{ con } C_3, 1]$

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)

    $$x_j \neq x_0 \vee \neg P(x_1, \ldots, x_j, \ldots x_n) \vee P(x_1, \ldots, x_0, \ldots, x_n)$$

  - Para cada símbolo de función f de L (de aridad n)

    $$x_j \neq x_0 \vee f(x_1, \ldots, x_j, \ldots x_n) = f(x_1, \ldots, x_0, \ldots, x_n)$$

- 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.