Tema 6: Formas normales, unificación, resolución.

(Apuntes elaborados por Juan Manuel Urbano Blanco)

Índice

1 Introducción

En este tema desarrollamos técnicas apropiadas para decidir si un conjunto de fórmulas de un lenguaje de predicados de primer orden $\mathcal{L}$, es insatisfacible ó inconsistente, y por tanto, basándonos en el Teorema 2 de la Sección 7 del Tema 5, podremos saber si una fórmula de $\mathcal{L}$ es o no consecuencia lógica de un subconjunto de $\mathcal{L}$.

2 Formas normales

Dado un conjunto de fórmulas Γ, vamos a aplicarle a cada fórmula en Γ ciertas transformaciones de manera que el conjunto resultante de fórmulas sea insatisfacible si y sólo si Γ también lo es, y con la ventaja adicional de que las transformaciones darán lugar a **fórmulas más "simples"** que las fórmulas de partida.

Para cada fórmula $\alpha \in \Gamma$ comenzamos calculando una forma normal prenexa, y entonces a partir de ésta última, obtenemos una forma normal de Skolem de α. Por último, nos restringiremos sólo a conjuntos formados por sentencias, y a partir de las correspondientes formas normales de Skolem, obtendremos sus formas clausuladas. La idea de todo este proceso es un generalización de la teoría estudiada en el Tema 4.

2.1 Forma normal prenexa

Una fórmula α está en **forma normal prenexa**, si en ella no aparecen cuantificadores o bien es de la forma

$$C_1\ x_1 \dots C_n\ x_n\ \beta,$$

donde cada C_i es un cuantificador $\forall$ ó $\exists$, y β es una fórmula en la cual no hay cuantificadores.

Ejemplo 1. Sea $\mathcal{L}$ un lenguaje de predicados de primer orden tal que

$$\mathrm{Cons}(\mathcal{L}) = \{a\}, \quad \mathrm{Func}(\mathcal{L}) = \{f^1\} \ \text{ y } \ \mathrm{Rel}(\mathcal{L}) = \{P^1, Q^2, R^0\}.$$

Las siguientes fórmulas de $\mathcal{L}$ están en forma normal prenexa:

$$R, \quad Q(x, f(a)), \quad P(x) \to R, \quad \forall x P(x), \quad \forall x \exists y \Big(Q(a,x) \wedge P(f(y))\Big), \quad \exists y \exists x \forall y \neg P(a).$$

Las siguientes no están en forma normal prenexa:

$$\forall x P(x) \to P(f(a)), \quad P(a) \to \exists x P(x), \quad \neg \exists y \Big(P(y) \wedge Q(a,y)\Big), \quad \forall x \neg \exists y Q(x,y).$$

□

Teorema 1. Para toda fórmula α existe otra fórmula α^* en forma normal prenexa semánticamente equivalente a α. □

La demostración se hace aplicando las propiedades de equivalencia lógica estudiadas al final del tema anterior.

Ejemplo 2. Obtengamos una forma normal prenexa para la fórmula siguiente

$$\alpha := P(y) \to \forall x R(x, a).$$

Sabemos que

$$\alpha \equiv \neg P(y) \vee \forall x R(x, a).$$

Como en la primera parte de la fórmula, es decir, en $\neg P(y)$, no hay ocurrencias libres de x, ampliamos el radio de acción del cuantificador $\forall x$, y obtenemos que

$$\neg P(y) \vee \forall x R(x, a) \ \equiv \ \forall x \Big(\neg P(y) \vee R(x, a)\Big).$$

Por consiguiente, la fórmula

$$\forall x \Big(\neg P(y) \vee R(x, a)\Big)$$

es una forma normal prenexa de α.

Nótese que la fórmula siguiente, equivalente a la anterior, es también una forma normal prenexa de α:

$$\forall x \Big(P(y) \to R(x, a)\Big).$$

□

Ejemplo 3. Calculamos una forma normal prenexa para la fórmula siguiente

$$\alpha := \exists x P(x) \to R(a, f(b)).$$

En primer lugar tenemos que

$$\alpha \equiv \neg \exists x P(x) \vee R(a, f(b)).$$

Interiorizamos la negación, y obtenemos que

$$\neg \exists x P(x) \vee R(a, f(b)) \;\equiv\; \forall x \neg P(x) \vee R(a, f(b)).$$

Como en la segunda parte de la fórmula obtenida, es decir, en $R(a, f(b))$, no hay ocurrencias libres de x, ampliamos el radio de acción del cuantificador $\forall x$, y obtenemos que

$$\forall x \neg P(x) \vee R(a, f(b)) \;\equiv\; \forall x \Big(\neg P(x) \vee R(a, f(b)) \Big).$$

Por tanto, la fórmula

$$\forall x \Big(\neg P(x) \vee R(a, f(b)) \Big)$$

es una forma normal prenexa de α.

La fórmula siguiente, equivalente a la anterior, es también una forma normal prenexa de α:

$$\forall x \Big(P(x) \to R(a, f(b)) \Big).$$

□

Ejemplo 4. Son formas normales prenexas de la fórmula

$$\alpha := \forall y P(y) \to \forall x R(x, a),$$

tanto la fórmula

$$\forall x \exists y \Big(\neg P(y) \vee R(x, a) \Big)$$

como la fórmula

$$\exists y \forall x \Big(\neg P(y) \vee R(x, a) \Big).$$

En este ejemplo, el orden de extracción de los cuantificadores es indiferente, pues las dos partes $\forall y P(y)$ y $\forall x R(x, a)$ de α son independientes, en el sentido de que no comparten ninguna variable. Como veremos en otros ejemplos, éste no será el caso. □

Ejemplo 5. La fórmula

$$\alpha := \exists x P(x) \to \forall x R(x, a)$$

es equivalente a

$$\neg \exists x P(x) \vee \forall x R(x, a),$$

la cual a su vez equivale a
$$\forall x \neg P(x) \vee \forall x R(x,a).$$
Los dos cuantificadores universales que aparecen en la fórmula resultante, aunque cuantifican a la misma variable, es posible que tengan significados diferentes, pues la variable x que aparece en la subfórmula $\neg P(x)$ posiblemente no tenga nada que ver con la variable x que aparece en la subfórmula $R(x,a)$.

Recordemos aquí que en general no son equivalentes las fórmulas
$$\forall x \alpha_1 \vee \forall x \alpha_2 \qquad \text{y} \qquad \forall x(\alpha_1 \vee \alpha_2).$$

Así, la fórmula
$$\forall x \neg P(x) \vee \forall x R(x,a),$$
puede que no sea equivalente a la fórmula
$$\forall x\Big(\neg P(x) \vee R(x,a)\Big),$$
con lo cual no podemos asegurar que ésta última sea una forma normal prenexa de α.

Por tanto, por ahora tenemos la fórmula
$$\forall x \neg P(x) \vee \forall x R(x,a).$$

Decidimos comenzar extrayendo el primero de los dos cuantificadores. Como en la subfórmula
$$\forall x R(x,a)$$
no hay ocurrencias libres de x, ampliamos el radio de acción del primer cuantificador $\forall x$, y resulta
$$\forall x \neg P(x) \vee \forall x R(x,a) \;\equiv\; \forall x\Big(\neg P(x) \vee \forall x R(x,a)\Big).$$
Si ahora nos proponemos extraer el cuantificador universal que hay dentro de los paréntesis, no lo podemos hacer tal cual, pues le afectaría a la variable x que aparece en la subfórmula precedente $\neg P(x)$, a la cual realmente no le afectaba.

Por tanto, hemos de renombrar variables. Obtenemos que
$$\forall x R(x,a) \;\equiv\; \forall y R(y,a),$$
con lo cual, resulta
$$\forall x\Big(\neg P(x) \vee \forall x R(x,a)\Big) \;\equiv\; \forall x\Big(\neg P(x) \vee \forall y R(y,a)\Big).$$
Ya sí podemos extraer el cuantificador $\forall y$, pues ahora en la subfórmula $\neg P(x)$ no hay ocurrencias libres de la variable y. Resulta por tanto
$$\forall x\Big(\neg P(x) \vee \forall y R(y,a)\Big) \;\equiv\; \forall x \forall y\Big(\neg P(x) \vee R(y,a)\Big),$$

siendo la fórmula

$$\forall x \forall y \Big(\neg P(x) \vee R(y,a) \Big)$$

una forma normal prenexa de α. □

Ejemplo 6. Si tenemos la fórmula

$$\alpha := \forall x P(x) \to \exists x R(x,a),$$

entonces

$$\alpha \;\equiv\; \neg \forall x P(x) \vee \exists x R(x,a) \;\equiv\; \exists x \neg P(x) \vee \exists x R(x,a) \;\equiv\; \exists x \Big(\neg P(x) \vee R(x,a) \Big).$$

La última equivalencia se basa en el Lema 4 de la Sección 8 del Tema 5. Por tanto una forma normal prenexa de α es

$$\exists x \Big(\neg P(x) \vee R(x,a) \Big).$$

Un camino alternativo hubiese sido renombrar en la fórmula

$$\exists x \neg P(x) \vee \exists x R(x,a)$$

para obtener la fórmula equivalente

$$\exists x \neg P(x) \vee \exists y R(y,a),$$

y de ésta, la fórmula

$$\exists x \exists y \Big(\neg P(x) \vee R(y,a) \Big),$$

la cual es otra forma normal prenexa de α. □

Ejemplo 7. Si tenemos la fórmula

$$\alpha := \neg \exists z \exists x R(x,y) \vee \exists y P(f(y)),$$

comenzamos interiorizando la negación de la subfórmula de la izquierda, y resulta

$$\alpha \;\equiv\; \forall z \forall x \neg R(x,y) \vee \exists y P(f(y)).$$

Por una parte, el cuantificador $\forall z$ es superfluo y lo podemos suprimir, con lo cual obtenemos

$$\alpha \;\equiv\; \forall x \neg R(x,y) \vee \exists y P(f(y)).$$

Como en la subfórmula de la izquierda hay una ocurrencia libre de la variable y, para extraer el cuantificador $\exists y$ primero hemos de renombrar todas las ocurrencias de la variable y en la subfórmula de la derecha. De este modo llegamos a que

$$\alpha \;\equiv\; \forall x \neg R(x,y) \vee \exists y_1 P(f(y_1)),$$

donde y_1 es un nuevo símbolo de variable, distinto de y. Por tanto,

$$\alpha \equiv \exists y_1\Big(\forall x \neg R(x,y) \vee P(f(y_1))\Big) \equiv \exists y_1 \forall x\Big(\neg R(x,y) \vee P(f(y_1))\Big),$$

siendo la fórmula

$$\exists y_1 \forall x\Big(\neg R(x,y) \vee P(f(y_1))\Big)$$

una forma normal prenexa de α.

Como veremos en la sección siguiente, al obtener una forma normal prenexa para una fórmula, es preferible, siempre que se pueda, extraer primero (de modo que queden más a la izquierda) los cuantificadores existenciales. □

Recordemos también que en general, no son equivalentes las fórmulas

$$\exists x \beta_1 \wedge \exists x \beta_2 \qquad \text{y} \qquad \exists x(\beta_1 \wedge \beta_2).$$

Ésto lo tendremos en cuenta en el ejemplo siguiente.

Ejemplo 8. Obtenemos una forma normal prenexa de

$$\alpha := \forall x\Big(R(x,y) \to \forall x R(x,a)\Big) \to \forall x \forall z R(x,b).$$

Aplicamos pasos similares a los que hemos aplicado en los ejemplos anteriores y obtenemos:

$$\alpha \equiv \neg \forall x\Big(\neg R(x,y) \vee \forall x R(x,a)\Big) \vee \forall x R(x,b),$$

$$\alpha \equiv \exists x \neg\Big(\neg R(x,y) \vee \forall x R(x,a)\Big) \vee \forall x R(x,b),$$

$$\alpha \equiv \exists x\Big(\neg\neg R(x,y) \wedge \neg \forall x R(x,a)\Big) \vee \forall x R(x,b),$$

$$\alpha \equiv \exists x\Big(R(x,y) \wedge \exists x \neg R(x,a)\Big) \vee \forall x R(x,b),$$

$$\alpha \equiv \exists x\Big(R(x,y) \wedge \exists x_1 \neg R(x_1,a)\Big) \vee \forall x R(x,b),$$

$$\alpha \equiv \exists x \exists x_1\Big(R(x,y) \wedge \neg R(x_1,a)\Big) \vee \forall x R(x,b),$$

$$\alpha \equiv \exists x \exists x_1\Big(R(x,y) \wedge \neg R(x_1,a)\Big) \vee \forall x_2 R(x_2,b),$$

$$\alpha \equiv \exists x \exists x_1 \forall x_2\Big(\big(R(x,y) \wedge \neg R(x_1,a)\big) \vee R(x_2,b)\Big).$$

Por tanto la fórmula

$$\exists x \exists x_1 \forall x_2\Big(\big(R(x,y) \wedge \neg R(x_1,a)\big) \vee R(x_2,b)\Big)$$

es una forma normal prenexa de α. □

Conviene conocer la propiedad siguiente de cara a nuestros propósitos.

Proposición 1. Dado un conjunto de fórmulas Γ, considérese el conjunto Γ^* formado al escoger una fórmula en forma normal prenexa para cada fórmula de Γ. Entonces Γ es insatisfacible, si y sólo si, Γ^* es insatisfacible. □

2.2 Forma normal de Skolem

Sea α^* una fórmula en forma normal prenexa. Grosso modo, una forma normal de Skolem para α^* será obtenida reemplazando las variables cuantificadas existencialmente en α^* por ciertos términos apropiados y "eliminando" los cuantificadores existenciales que aparecen en α^*.

Dada $\alpha^* \in \text{Form}(\mathcal{L})$ en forma normal prenexa, una forma normal de Skolem de α^* es una fórmula obtenida al ir substituyendo cada variable x_i, cuantificada existencialmente, por un término $f(x_{i_1}, \ldots, x_{i_m})$, donde f es un símbolo de función de aridad m que no aparece en α^* ni ha sido usado hasta el momento para la substitución de otra variable, y los símbolos $x_{i_1}, \ldots, x_{i_m}$ son las variables cuantificadas universalmente que preceden a $\exists x_i$ (es decir, aparacen a su izquierda) en la escritura de α^*; de no haber ninguna, la substitución se hace por un símbolo de constante que no aparezca en α^* ni tampoco haya sido utilizado hasta el momento.

Ejemplo 9.

	F.N.P.	F.N.S.
1.	$P(x)$	$P(x)$
2.	$P(a)$	$P(a)$
3.	$\forall x P(x)$	$\forall x P(x)$
4.	$\exists x P(x)$	$P(a)$
5.	$\exists x Q(a,x)$	$Q(a,b)$
6.	$\exists x \exists y R(a,x,f(y))$	$R(a,b,f(c))$
7.	$\forall x \exists y Q(x,y)$	$\forall x Q(x,f(x))$
8.	$\forall x \exists y Q(f(x),y)$	$\forall x Q(f(x),g(x))$
9.	$\forall x \forall y \exists z R(x,y,z)$	$\forall x \forall y R(x,y,f(x,y))$
10.	$\forall x \forall y \exists z R(f(x),y,z)$	$\forall x \forall y R(f(x),y,g(x,y))$
11.	$\exists x \forall y \exists z R(h(a,x),y,z)$	$\forall y R(h(a,b),y,f(y))$
12.	$\forall x \exists y \forall x_1 \exists y_1 R(h(x,y),y_1,f(x_1))$	$\forall x \forall x_1 R(h(x,g_1(x)),g_2(x,x_1),f(x_1))$

Los reemplazamientos que hemos aplicado son los siguientes:

1. ninguno.
2. ninguno.
3. ninguno.
4. $x := a$.
5. $x := b$.
6. $x := b,\ y := c$.
7. $y := f(x)$.
8. $y := g(x)$.
9. $z := f(x, y)$.
10. $z := g(x, y)$.
11. $x := b,\ z := f(y)$.
12. $y := g_1(x),\ y_1 := g_2(x, x_1)$.

□

Ejemplo 10. Dada la fórmula

$$\alpha^* := \exists x P(x)$$

en forma normal prenexa, una forma normal de Skolem para α^* es

$$\beta := P(a),$$

la cual ha sido obtenida aplicando el reemplazamiento $x := a$.

Consideramos la estructura $\mathcal{E}$ dada por:

- $\mathrm{D} = \mathbb{Z}$,
- $a^{\mathcal{E}} = 5$,
- $P^{\mathcal{E}}(r) = \mathbf{1}$ si y sólo si r es par.

Entonces para cualquier asignación v en $\mathcal{E}$, se cumple $\mathrm{I}^v(\alpha^*) = \mathbf{1}$, mientras que $\mathrm{I}^v(\beta) = \mathbf{0}$. Por tanto α^* y β no son equivalentes.

Sin embargo, los conjuntos de fórmulas $\{\alpha^*\}$ y $\{\beta\}$ son ambos satisfacibles. Que $\{\alpha^*\}$ es satisfacible, lo acabamos de poner de manifiesto con la interpretación anterior. El conjunto $\{\beta\}$ es también satisfacible. Basta considerar la estructura siguiente:

- $\mathrm{D} = \mathbb{Z}$,
- $a^{\mathcal{E}} = 5$,
- $P^{\mathcal{E}}(r) = \mathbf{1}$ si y sólo si r es impar.

□

Vemos con este ejemplo que al “skolemizar” de esta forma, algunas fórmulas ya no son equivalentes a su forma normal de Skolem, es decir, se puede perder la equivalencia semántica, pero el carácter de satisfacibilidad ó insatisfacibilidad del conjunto formado por ellas, que es lo que realmente nos interesa, se conserva.

Proposición 2. Sea Γ^* un conjunto de fórmulas en forma normal prenexa. Calculamos una forma normal de Skolem para cada fórmula en Γ^*, y obtenemos el conjunto Γ^{**}. Entonces Γ^* es insatisfacible, si y sólo si, Γ^{**} es insatisfacible. □

Ya comentamos en un ejemplo previo, que a la hora de obtener una forma normal de Skolem de una fórmula dada, interesa extraer primero los cuantificadores existenciales, siempre que ello sea posible.

Ejemplo 11. Consideramos la fórmula

$$\alpha := \forall x_1 P(x_1) \wedge \forall x_2 P(x_2) \wedge \forall x_3 P(x_3) \wedge \exists x_4 P(x_4) \wedge \exists x_5 P(x_5).$$

Una forma normal prenexa de α es

$$\forall x_1 \forall x_2 \forall x_3 \exists x_4 \exists x_5 \Big(P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge P(x_4) \wedge P(x_5) \Big),$$

que conduce a la forma normal de Skolem

$$\forall x_1 \forall x_2 \forall x_3 \Big(P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge P(f(x_1, x_2, x_3)) \wedge P(g(x_1, x_2, x_3)) \Big),$$

con los reemplazamientos $x_4 := f(x_1, x_2, x_3)$ y $x_5 := g(x_1, x_2, x_3)$.

Claramente es preferible obtener la forma normal prenexa

$$\exists x_4 \exists x_5 \forall x_1 \forall x_2 \forall x_3 \Big(P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge P(x_4) \wedge P(x_5) \Big),$$

a partir de la cual, con los reemplazamientos $x_4 := a$, $x_5 := b$, se obtiene la forma normal de Skolem

$$\forall x_1 \forall x_2 \forall x_3 \Big(P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge P(a) \wedge P(b) \Big),$$

más sencilla que la previamente obtenida. □

Recalcamos que cuando vayamos a transformar varias fórmulas que ya están en forma normal prenexa a forma normal de Skolem, procuraremos que los nuevos símbolos introducidos para cada una de las fórmulas (ya sean de constante o de función) no aparezcan ni hayan sido utilizados en ninguna de las otras fórmulas dadas.

Ejemplo 12. Supongamos el conjunto de fórmulas en forma normal prenexa:

$$\Gamma^* := \Big\{ \forall y \neg Q(a, y),\ \exists x \forall y Q(x, y) \Big\},$$

y sea el conjunto Γ^{**} siguiente formado por una forma normal de Skolem para cada una de las fórmulas de Γ^*, pero donde no hemos tenido en cuenta el criterio anterior acerca de los símbolos nuevos:

$$\Gamma^{**} := \Big\{\forall y \neg Q(a,y),\ \forall y Q(a,y)\Big\}.$$

Es evidente que Γ^{**} es insatisfacible, ya que es imposible que $Q(a,y)$ sea al mismo tiempo verdadero y falso bajo una interpretación cualquiera. Veamos que Γ^* es satisfacible. Definimos una estructura $\mathcal{E}$ como sigue:

$$\mathrm{D}_{\mathcal{E}} = \{1,2,3\}.$$
$$a^{\mathcal{E}} = 3.$$
$$\text{Para } r,s \in \mathrm{D}_{\mathcal{E}},\ Q^{\mathcal{E}}(r,s) = \begin{cases} \mathbf{1} & \text{si } (r,s) \in \{(1,1),(1,2),(1,3)\}, \\ \mathbf{0} & \text{en otro caso.} \end{cases}$$

Ya que las dos fórmulas en Γ son sentencias, para cualquier asignación v en $\mathcal{E}$, se verifica que

$$\mathrm{I}^v\Big(\forall y \neg Q(a,y)\Big) = \mathrm{I}^v\Big(\neg\exists y Q(a,y)\Big) = \mathbf{1},$$

pues no existe ningún elemento $r \in \mathrm{D}_{\mathcal{E}}$ tal que $P^{\mathcal{E}}(3,r) = \mathbf{1}$. Además

$$\mathrm{I}^v\Big(\exists x \forall y Q(x,y)\Big) = \mathbf{1},$$

pues existe el elemento $r = 1 \in \mathrm{D}_{\mathcal{E}}$ el cual está relacionado con cualquier elemento perteneciente a $\mathrm{D}_{\mathcal{E}}$. Por tanto, el conjunto Γ^* es satisfacible. $\square$

2.3 Forma normal clausulada

En esta sección nos restringiremos a fórmulas que sean **sentencias**. Recuérdese que una fórmula es una sentencia si no contiene ocurrencias libres de ninguna variable. Ya comentamos en el Tema 5, al traducir frases a lenguajes de predicados de primer orden, que en las fómulas provenientes del lenguaje hablado no aparecen ocurrencias libres de variables. Tal y como veremos en los ejercicios, la restricción que hemos impuesto a las fórmulas no afecta a nuestro objetivo.

Así pues vamos a partir de sentencias que están ya en forma normal de Skolem, digamos

$$\alpha := \forall x_1 \cdots \forall x_n \beta,$$

donde no aparece ningún cuantificador en β.

Obtenemos una forma normal clausulada de α simplemente calculando una forma clausulada de β de manera análoga a como vimos en el Tema 4 para la Lógica de proposiciones.

Si tenemos una cláusula C y L es un literal que aparece en C varias veces, reduciremos todas las ocurrencias del literal L en C a una sóla. De este modo obtenemos otra cláusula más simple que es equivalente a C.

Por ejemplo, la cláusula

$$R(x, f(y)) \vee \neg S(x, a) \vee R(x, f(y)) \vee R(y, f(y))$$

la reducimos a

$$R(x, f(y)) \vee \neg S(x, a) \vee R(y, f(y)).$$

Si $C_1 \wedge \cdots \wedge C_m$ es una fórmula equivalente a β, siendo cada C_i una disyunción de (uno o más) literales, decimos que la fórmula

$$\forall x_1 \cdots \forall x_n \Big(C_1 \wedge \cdots \wedge C_m \Big)$$

es una **forma normal clausulada** de α. También decimos que las fórmulas

$$\underbrace{\forall x_1 \cdots \forall x_n C_1}, \ldots, \underbrace{\forall x_1 \cdots \forall x_n C_m}$$

son las cláusulas de α.

Nótese que por las equivalencias estudiadas en el Tema 5, se cumple que

$$\forall x_1 \cdots \forall x_n \Big(C_1 \wedge \cdots \wedge C_m \Big) \;\equiv\; \Big(\forall x_1 \cdots \forall x_n C_1 \Big) \wedge \cdots \wedge \Big(\forall x_1 \cdots \forall x_n C_m \Big).$$

De hecho, de ahora en adelante al escribir una forma normal clausulada de una fórmula dada (y en particular al escribir las cláusulas que la componen) omitiremos los cuantificadores universales precedentes, sin olvidarnos de que todas las variables estarán cuantificadas universalmente. Por este motivo nos hemos restringido a sentencias, para que no haya ambigüedad. Así pues, si escribimos la cláusula

$$R(x, a) \vee \neg P(z)$$

realmente nos estamos refiriendo a

$$\forall x \forall z \Big(R(x, a) \vee \neg P(z) \Big).$$

Siguiendo con la nomenclatura anterior, diremos también que el conjunto $\{C_1, \ldots, C_m\}$ es una forma normal clausulada de α.

Ejemplo 13. Si tenemos la fórmula ya en forma normal clausulada,

$$\alpha \;:=\; \forall x \forall y \Big(\big(Q(x, a) \vee \neg P(a) \big) \wedge \big(\neg Q(a, y) \vee R \big) \Big),$$

entonces

$$\forall x \forall y \Big(Q(x, a) \vee \neg P(a) \Big) \qquad \text{y} \qquad \forall x \forall y \Big(\neg Q(a, y) \vee R \Big)$$

son las cláusulas de α.

También diremos que el conjunto de fórmulas

$$\Big\{Q(x,a) \vee \neg P(a),\ \neg Q(a,y) \vee R\Big\}$$

es una forma normal clausulada de α. □

Ejemplo 14. En el Ejemplo 8 vimos que una forma normal prenexa de la fórmula

$$\forall x\Big(R(x,y) \to \forall x R(x,a)\Big) \to \forall x \forall z R(x,b),$$

es

$$\exists x \exists x_1 \forall x_2\Big(\big(R(x,y) \wedge \neg R(x_1,a)\big) \vee R(x_2,b)\Big).$$

Por tanto, una forma normal prenexa de la fórmula

$$\beta := \forall y\Big(\forall x\Big(R(x,y) \to \forall x R(x,a)\Big) \to \forall x \forall z R(x,b)\Big),$$

es

$$\forall y \exists x \exists x_1 \forall x_2\Big(\Big(R(x,y) \wedge \neg R(x_1,a)\Big) \vee R(x_2,b)\Big).$$

Posiblemente haya otras formas normales prenexas para β más favorables que la que hemos obtenido, aunque ello no nos importa en este momento.

Así, una forma normal de Skolem para β es

$$\forall y \forall x_2\Big(\Big(R(f(y),y) \wedge \neg R(g(y),a)\Big) \vee R(x_2,b)\Big),$$

donde hemos aplicado los reemplazamientos $x := f(y)$, $x_1 := g(y)$.

Para obtener una forma normal clausulada, aplicamos una propiedad distributiva dentro del paréntesis grande. Resulta

$$\forall y \forall x_2\Big(\Big(R(f(y),y) \vee R(x_2,b)\Big) \wedge \Big(\neg R(g(y),a)\big) \vee R(x_2,b)\Big)\Big),$$

que es una forma normal clausulada de β. De acuerdo con las definiciones anteriores, las cláusulas de β son

$$\forall y \forall x_2\Big(R(f(y),y) \vee R(x_2,b)\Big),\ \forall y \forall x_2\Big(\neg R(g(y),a)\big) \vee R(x_2,b)\Big).$$

También decimos que el conjunto

$$\Big\{R(f(y),y) \vee R(x_2,b),\ \neg R(g(y),a)\big) \vee R(x_2,b)\Big\}$$

es una forma normal clausulada de β. □

Se verifica el siguiente teorema.

Teorema 2. Dado un conjunto de sentencias Γ, sea el conjunto Γ' resultante de considerar para cada fórmula de Γ una forma normal clausulada, y sea Γ'' el conjunto formado por todas las cláusulas que aparecen en las fórmulas de Γ'. Entonces Γ es insatisfacible, si y sólo si, Γ'' es insatisfacible.

Así, probar la insatisfacibilidad ó inconsistencia de un conjunto de sentencias, es equivalente a probar la insconsistencia de un conjunto de cláusulas, que son fórmulas más sencillas. Como se puede apreciar, es la misma idea que ya vimos en el Tema 4, pero ahora en el contexto de la Lógica de predicados.

Ejemplo 15. Dado el conjunto de sentencias

$$\Gamma := \Big\{\forall x(P(x) \to Q(x)),\ \exists y(P(y) \wedge P(a)),\ \neg\exists z Q(z)\Big\},$$

obtenemos una forma normal clausulada para cada una de ellas:

$\forall x(P(x) \to Q(x))$	$\forall x(\neg P(x) \vee Q(x))$
$\exists y(P(y) \wedge P(a))$	$P(b) \wedge P(a)$
$\neg\exists z Q(z)$	$\forall z \neg Q(z)$

Entonces, siguiendo la notación del teorema anterior, tenemos

$$\Gamma' := \Big\{\forall x(\neg P(x) \vee Q(x)),\ P(b) \wedge P(a),\ \forall z \neg Q(z)\Big\}.$$

Al descomponer cada una de sus fórmulas en cláusulas y omitir cuantificadores, resulta

$$\Gamma'' := \Big\{\neg P(x) \vee Q(x),\ P(b),\ P(a),\ \neg Q(z)\Big\}.$$

Según el teorema anterior, el conjunto de sentencias Γ es insatisfacible, si y sólo si, el conjunto de cláusulas Γ'' es insatisfacible. Aplicando los algoritmos que veremos en la última sección del tema, obtendremos fácilmente que el conjunto de cláusulas Γ'' es insatisfacible. □

3 Unificación

Recordemos que un **literal** en un lenguaje de predicados de primer orden puede ser una fórmula atómica o bien la negación de una fórmula atómica.

Dados dos literales, el proceso de reemplazar determinados símbolos de variable en dichos literales de forma que ambos resulten idénticos, se denomina **unificación**. Por ejemplo, si tenemos los literales

$$R(x, f(f(a))) \quad \text{y} \quad R(y, f(z)),$$

ambos se unifican aplicando el reemplazamiento

$$\begin{cases} x := y \\ z := f(a) \end{cases}$$

para obtener el literal

$$R(y, f(f(a))).$$

Por supuesto hay otros reemplazamientos que también unifican a dichos literales, como por ejemplo,

$$\begin{cases} x := g(h(z_1)) \\ y := g(h(z_1)) \\ z := f(a) \end{cases}$$

para obtener

$$R(g(h(z_1)), f(f(a))),$$

ó bien

$$\begin{cases} x := b \\ y := b \\ z := f(a) \end{cases}$$

para obtener

$$R(b, f(f(a))).$$

Recalcamos que se trata de un proceso de reemplazamiento de unas letras por otras secuencias de letras, es decir, es un proceso puramente sintáctico, en el que no se evalúa nada. A continuación damos las definiciones precisas.

Una **substitución** en un lenguaje de predicados $\mathcal{L}$ viene dada por una aplicación

$$\sigma : \mathrm{Var}(\mathcal{L}) \to \mathrm{Term}(\mathcal{L})$$

la cual se extiende a todo término (de forma parecida a como ocurría con las asignaciones en el Tema 5). Esta nueva aplicación del conjunto $\mathrm{Term}(\mathcal{L})$ en $\mathrm{Term}(\mathcal{L})$ también la denotaremos por σ. Nótese que si c es un símbolo de constante, entonces $\sigma(c) = c$.

Dada la substitución σ que verifica

$$\begin{cases} \sigma(x_1) & = & t_1, \\ & \vdots & \\ \sigma(x_n) & = & t_n, \\ \sigma(y) & = & y, \text{ si } y \in \mathrm{Var}(\mathcal{L}) \setminus \{x_1, \ldots, x_n\}, \end{cases}$$

donde como es usual x_i son símbolos de variable del lenguaje $\mathcal{L}$ y t_j son términos de $\mathcal{L}$, escribiremos

$$\sigma = (x_1|t_1, \ldots, x_n|t_n).$$

Así, al dar una substitución σ, asumiremos que los símbolos de variable que no aparecen como una parte izquierda, se quedan fijos.

Denotaremos a la substitución identidad como $\psi = ()$.

Ejemplo 16. Si tenemos un lenguaje de predicados $\mathcal{L}$ tal que

$$\mathrm{Cons}(\mathcal{L}) = \{a\}, \quad \mathrm{Func}(\mathcal{L}) = \{f^1\} \;\; \text{y} \;\; \mathrm{Rel}(\mathcal{L}) = \{P^1, Q^2, R^0\},$$

entonces la aplicación $\sigma : \mathrm{Var}(\mathcal{L}) \to \mathrm{Term}(\mathcal{L})$ determinada por las condiciones

$$\sigma(x) = f(f(a)), \quad \sigma(y) = f(y)$$

es un substitución en $\mathcal{L}$ que denotamos como

$$\sigma = \big(x|f(f(a)),\ y|f(y)\big).$$

A partir de esta información calculamos

$$\begin{aligned} &\sigma\Big(f(x)\Big) = f(f(f(a))), \\ &\sigma(a) = a, \\ &\sigma(f(y)) = f(f(y)). \end{aligned}$$

Además tenemos que

$$\sigma(z) = z, \quad \sigma(x_1) = x_1,$$

con lo cual

$$\begin{aligned} &\sigma(f(z)) = f(z), \\ &\sigma(f(f(x_1))) = f(f(x_1)). \end{aligned}$$

□

Dado un literal L, ya sea de la forma $P(t_1, \ldots, t_n)$ o bien $\neg P(t_1, \ldots, t_n)$, y una substitución σ, denotamos por $\sigma(L)$ el literal que resulta de reemplazar cada símbolo de variable x_i que aparece en L por el valor $\sigma(x_i)$. Es decir, si

$$L = P(t_1, \ldots, t_n),$$

entonces

$$\sigma(L) = P(\sigma(t_1), \ldots, \sigma(t_n)),$$

y si

$$L = \neg P(t_1, \ldots, t_n),$$

entonces

$$\sigma(L) = \neg P(\sigma(t_1), \ldots, \sigma(t_n)).$$

Ejemplo 17. Si tenemos la substitución $\sigma = \big(x|f(x,y),\ y|a\big)$, y el literal

$$L := R(x, g(y), z),$$

entonces

$$\sigma(L) = R(f(x,y), g(a), z),$$

mientras que si

$$L := \neg R(x, g(y), z),$$

entonces

$$\sigma(L) = \neg R(f(x,y), g(a), z).$$

$\square$

Ejemplo 18. Sean la substitución

$$\sigma = \big(x|f(a,b),\ y|z\big)$$

y el literal

$$L := P(x, g(b), y, g(z)).$$

Entonces

$$\sigma(L) = P\Big(f(a,b), g(b), z, g(z)\Big).$$

$\square$

Ejemplo 19. Si $\sigma = \big(x|y,\ y|x\big)$ y $L := P(x, g(b), y, g(z))$, entonces

$$\sigma(L) = P\Big(y, g(b), x, g(z)\Big)$$

y

$$\sigma^2(L) = \sigma\Big(\sigma(L)\Big) = L.$$

$\square$

Una substitución σ es un **unificador** para un conjunto de literales $\Gamma = \{L_1, \ldots, L_n\}$, si

$$\sigma(L_1) = \cdots = \sigma(L_n).$$

Un conjunto de literales Γ se dice **unificable**, si existe al menos un unificador para Γ.

Ejemplo 20. El conjunto de literales

$$\Big\{P(x),\ P(a),\ P(y)\Big\}$$

es unificable tomando la substitución

$$\sigma = \big(x|a,\ y|a\big),$$

pues
$$\begin{aligned}
&\sigma(P(x)) = P(\sigma(x)) = P(a),\\
&\sigma(P(a)) = P(a),\\
&\sigma(P(y)) = P(\sigma(y)) = P(a).
\end{aligned}$$
$\square$

Ejemplo 21. El conjunto de literales
$$\Big\{P(x,z),\ \ P(f(y),y)\Big\}$$
es unificable, pues si tomamos la substitución
$$\sigma = \big(x|f(y),\ \ z|y\big)$$
resulta
$$\sigma(P(x,z)) = P(\sigma(x),\sigma(z)) = P(f(y),y)$$
y
$$\sigma(P(f(y),y)) = P(f(y),y),$$
pues se supone que $\sigma(y) = y$. $\square$

Ejemplo 22. El conjunto de literales
$$\Big\{P(x,a),\ \ P(b,x)\Big\}$$
no es unificable, pues no existe ninguna substitución σ tal que $\sigma(x) = b$ y $\sigma(x) = a$, ya que a y b son símbolos de constante distintos. $\square$

Una vez que tenemos un unificador para un conjunto de literales Γ, en general, podremos construir a partir de él infinitos unificadores para Γ. De entre todos los posibles unificadores para Γ, habrá unos que son más simples que los demás.

Un unificador σ para un conjunto de literales Γ se dice que es un **unificador principal** o **de máxima generalidad** para Γ, si para cualquier otro unificador ρ para Γ existe una substitución μ de forma que $\rho = \mu \circ \sigma$, es decir, ρ se obtiene componiendo μ con σ.

Ejemplo 23. El conjunto de literales
$$\Gamma := \Big\{P(a),\ \ P(x)\Big\}$$
es unificable mediante el unificador $\sigma = \big(x|a\big)$, el cual es un unificador principal para Γ. $\square$

Ejemplo 24. Dado el conjunto de literales
$$\Gamma := \Big\{P(x), P(y)\Big\},$$
y las substituciones siguientes,

1. $\sigma_1 = (y|x)$,
2. $\sigma_2 = (x|z, y|z)$,
3. $\sigma_3 = (x|f(y), y|f(y))$,
4. $\sigma_4 = (x|a, y|a)$,

todas ellas son unificadores para Γ, de los cuales sólo σ_1 es unificador principal para Γ. $\square$

3.1 Cálculo de un unificador principal mediante la resolución de sistemas de ecuaciones en términos

Dados los literales L_1 y L_2 de un lenguaje de predicados $\mathcal{L}$, para que puedan ser unificables es necesario que ambos vayan negados o bien ambos sin negar, y que además empiecen por el mismo símbolo de relación. Esto es, L_1 y L_2 deben ser,

$$L_1 = R(t_1, \ldots, t_n), \quad L_2 = R(t'_1, \ldots, t'_n),$$

o bien

$$L_1 = \neg R(t_1, \ldots, t_n), \quad L_2 = \neg R(t'_1, \ldots, t'_n).$$

Queda claro pues que L_1 y L_2 son unificables si y sólo si existe una substitución σ en $\mathcal{L}$ tal que $\sigma(t_i) = \sigma(t'_i)$ para todo $i = 1, \ldots, n$. Esta idea da lugar a la siguiente definición y proporciona un método para resolver el problema de la unificación.

Una **ecuación** en un lenguaje de predicados $\mathcal{L}$ es una pareja de términos de $\mathcal{L}$, es decir, (t_1, t_2) con $t_1, t_2 \in \mathrm{Term}(\mathcal{L})$. La ecuación (t_1, t_2) la escribiremos como

$$t_1 = t_2.$$

Una **solución** de una ecuación (t_1, t_2) es una substitución σ en $\mathcal{L}$ que cumple

$$\sigma(t_1) = \sigma(t_2).$$

Ejemplo 25. Cada una de las substituciones

$$\sigma_1 = \big(x|f(x),\ y|f(x)\big), \qquad \sigma_2 = \big(x|y\big), \qquad \sigma_3 := \big(y|x\big)$$

es una solución de la ecuación $x = y$.

La ecuación

$$f(x) = g(x, y)$$

no tiene solución. Tampoco tiene solución la ecuación

$$a = f(x).$$

$\square$

Un **sistema de ecuaciones** E es un conjunto finito de ecuaciones.

Ejemplo 26. El sistema $E := \{x = y,\ y = a,\ z = b\}$ consta de tres ecuaciones. □

Una **solución** de un sistema de ecuaciones es una substitución σ que es solución de cada una de las ecuaciones del sistema. Un sistema de ecuaciones en términos se dice compatible, si posee al menos una solución. En caso contrario, se denomina incompatible.

Dos **sistemas** de ecuaciones son **equivalentes** si tienen las mismas soluciones.

Ejemplo 27. La substitución $\sigma = \big(x|a,\ y|a,\ z|b\big)$ es solución del sistema

$$E := \{x = y,\ y = a,\ z = b\}.$$

□

Un sistema $E := \{e_1, e_2, \ldots, e_n\}$ está en **forma resuelta**, si cada ecuación e_i es de la forma

$$x_i = t_i,$$

donde

- como es usual los símbolos x_i son símbolos de variable,
- $x_i \neq x_j$ para $i \neq j$,
- los símbolos t_i son términos de $\mathcal{L}$, y además
- ninguna de las variables que aparecen en los términos t_i pertenece al conjunto

$$\Big\{x_1, x_2, \ldots, x_n\Big\}.$$

Si el sistema

$$E := \Big\{x_1 = t_1,\ x_2 = t_2, \ldots,\ x_n = t_n\Big\}$$

está en forma resuelta (donde como es habitual $x_1, x_2, \ldots, x_n$ son símbolos de variable y $t_1, t_2, \ldots, t_n$ denotan términos), definimos la substitución

$$\sigma_E = \Big(x_1|t_1,\ x_2|t_2, \ldots, x_n|t_n\Big).$$

Teorema 3. Si tenemos un conjunto de literales Γ, a partir del cual obtenemos un sistema de ecuaciones en términos E el cual está en forma resuelta, entonces Γ es unificable y la substitución σ_E es un unificador principal para Γ. □

A continuación damos una serie de propiedades que nos van a permitir transformar un sistema de ecuaciones en otro equivalente a él, de forma que, tras un número finito de transformaciones lleguemos a un sistema en forma resuelta o a un sistema incompatible.

Lema 1. Los sistemas $E \cup \{t = t\}$ y E son equivalentes. $\square$

Lema 2. Los sistemas $E \cup \{t_1 = t_2\}$ y $E \cup \{t_2 = t_1\}$ son equivalentes. $\square$

Lema 3. Los sistemas $E \cup \{f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)\}$ y $E \cup \{t_1 = u_1, \ldots, t_n = u_n\}$ son equivalentes. $\square$

Lema 4. Dado un símbolo de variable x y un término t tales que x no aparece en t, entonces los sistemas $E \cup \{x = t\}$ y $\sigma(E) \cup \{x = t\}$, con $\sigma = (x|t)$, son equivalentes. ($\sigma(E)$ denota el sistema obtenido al aplicar σ a cada una de las ecuaciones en E.) $\square$

Lema 5. El sistema $E \cup \{f(t_1, \ldots, t_n) = g(u_1, \ldots, u_m)\}$, con $f \neq g$, es incompatible. $\square$

Lema 6. El sistema $E \cup \{f(t_1, \ldots, t_n) = a\}$, con a un símbolo de constante, es incompatible. $\square$

Lema 7. Si x es un símbolo de variable, t es un término tal que $t \neq x$, de forma que x aparece en t, entonces el sistema $E \cup \{x = t\}$ no tiene solución. $\square$

Teorema 4. Para todo sistema compatible existe un sistema equivalente que está en forma resuelta. $\square$

Ejemplo 28. Determinar si el conjunto de literales

$$\Gamma := \Big\{ P(a, x, f(g(y))),\ P(z, f(z),\ f(u)) \Big\}$$

es unificable o no lo es. Caso de serlo, encontrar un unificador principal para él.

El sistema de ecuaciones asociado a este problema es

$$\left.\begin{array}{l} a = z \\ x = f(z) \\ f(g(y)) = f(u) \end{array}\right\}$$

que es equivalente a

$$\left.\begin{array}{l} z = a \\ x = f(z) \\ f(g(y)) = f(u) \end{array}\right\}$$

el cual equivale a

$$\left.\begin{array}{l} z = a \\ x = f(a) \\ f(g(y)) = f(u) \end{array}\right\}$$

equivalente a

$$\left.\begin{array}{l} z = a \\ x = f(a) \\ g(y) = u \end{array}\right\}$$

y por último equivalente a

$$\left.\begin{array}{l} z = a \\ x = f(a) \\ u = g(y) \end{array}\right\}$$

que está en forma resuelta. Por tanto Γ es unificable y un unificador principal para Γ es

$$\sigma_E = \Big(z|a,\ x|f(a),\ u|g(y)\Big).$$

$\square$

Ejemplo 29. Determinar si el conjunto de literales

$$\Gamma := \Big\{P(x,y),\ P(f(z),x),\ P(u,f(x))\Big\}$$

es unificable o no, y en caso de serlo encontrar un unificador principal para él.

Un sistema de ecuaciones asociado a este problema es:

$$\left.\begin{array}{l} x = f(z) \\ y = x \\ x = u \\ y = f(x) \end{array}\right\}$$

Este sistema es equivalente a

$$\left.\begin{array}{l} x = f(z) \\ y = f(z) \\ f(z) = u \\ y = f(f(z)) \end{array}\right\}$$

que es equivalente a

$$\left.\begin{array}{l} x = f(z) \\ y = f(z) \\ u = f(z) \\ y = f(f(z)) \end{array}\right\}$$

equivalente a

$$\left.\begin{array}{l} x = f(z) \\ y = f(z) \\ u = f(z) \\ f(z) = f(f(z)) \end{array}\right\}$$

que por último es equivalente a

$$\left.\begin{array}{l} x = f(z) \\ y = f(z) \\ u = f(z) \\ z = f(z) \end{array}\right\}$$

el cual al contener la ecuación $z = f(z)$, no tiene solución. Por tanto el conjunto Γ no es unificable. $\square$

4 Resolución

Hemos visto ya anteriormente la siguiente propiedad, llamada *Modus ponendo ponens*

$$\Big\{\alpha,\ \alpha \to \beta\Big\} \models \beta,$$

donde α y β son dos fórmulas cualesquiera de un lenguaje de predicados $\mathcal{L}$.

Ésta propiedad la podemos expresar de modo equivalente como

$$\Big\{\alpha,\ \neg\alpha\vee\beta\Big\} \models \beta,$$

y nos dice que siempre que tengamos en nuestro conjunto de fórmulas dos fórmulas α y $\neg\alpha\vee\beta$, a partir de ellas podemos deducir la fórmula β.

Así pues, tenemos una *regla de inferencia ó de deducción*, la cual a partir de dos fórmulas apropiadas nos permite obtener otra que es consecuencia lógica de las dos anteriores

Esta regla de deducción se puede generalizar de la manera siguiente.

Lema 8. Dadas las fórmulas

$$\alpha \vee \beta_1 \vee \cdots \vee \beta_m$$

y

$$\neg\alpha \vee \beta_1' \vee \cdots \vee \beta_n',$$

se verifica que

$$\Big\{\alpha \vee \beta_1 \vee \cdots \vee \beta_m,\ \neg\alpha \vee \beta_1' \vee \cdots \vee \beta_n'\Big\} \models \beta_1 \vee \cdots \vee \beta_m \vee \beta_1' \vee \cdots \vee \beta_n'.$$

□

Recordemos que un literal de $\mathcal{L}$ es una fórmula atómica ó la negación de una fórmula atómica. Además, si L es un literal, entonces su literal complementario se define como

$$L^c = \begin{cases} L' & \text{si } L = \neg L', \\ \neg L' & \text{si } L = L', \end{cases}$$

donde L' denota una fórmula atómica.

De forma parecida a lo que hicimos en el Tema 4 cuando estudiamos el Algoritmo de Davis-Putnam, aquí también trabajaremos con cláusulas, motivo por el cual damos la siguiente versión del Lema 8.

Lema 9. Dadas las cláusulas $L \vee L_1 \vee \cdots \vee L_m$ y $L^c \vee L_1' \vee \cdots \vee L_n'$, se verifica que

$$\Big\{L \vee L_1 \vee \cdots \vee L_m,\ L^c \vee L_1' \vee \cdots \vee L_n'\Big\} \models L_1 \vee \cdots \vee L_m \vee L_1' \vee \cdots \vee L_n'.$$

□

Si C es una cláusula y L es un literal que aparece en C, notamos por $C - L$ a la cláusula que tiene todos los literales de C excepto L. En el caso en que $C = L$, a $C - L$ lo representamos por $\square$, y nos referimos a ella como la **cláusula vacía**. Recordemos que todo conjunto de cláusulas que contenga a la cláusula vacía se considera insatisfacible ó inconsistente.

Sean C_1 y C_2 dos cláusulas *sin variables comunes* y sean L_1 y L_2 dos literales de C_1 y C_2, respectivamente. Si L_1 y L_2^c tienen un unificador principal, σ, entonces la cláusula

$$\Big(\sigma(C_1) - \sigma(L_1)\Big) \ \vee \ \Big(\sigma(C_2) - \sigma(L_2)\Big)$$

es una **resolvente binaria** de C_1 y C_2. En tal caso diremos que L_1 y L_2 son los literales sobre los que se resuelve.

Como se supone que C_1 y C_2 son sentencias, donde sus variables aparecen cuantificadas universalmente, si es necesario, siempre podemos renombrar variables. Por ello podemos suponer que las cláusulas consideradas no tienen variables en común.

Lema 10. Si C es una resolvente binaria de las cláusulas C_1 y C_2, entonces

$$\Big\{C_1, C_2\Big\} \ \models \ C.$$

$\square$

Ejemplo 30. Sean las cláusulas

$$\begin{aligned} C_1 \equiv &\ \ R(f(x), b) \vee S(x, y) \vee P(a), \\ C_2 \equiv &\ \ \neg R(f(z), b) \vee \neg P(f(z)), \end{aligned}$$

es decir,

$$\begin{aligned} C_1 \equiv &\ \ \forall x \forall y \Big(R(f(x), b) \vee S(x, y) \vee P(a)\Big), \\ C_2 \equiv &\ \ \forall z \Big(\neg R(f(z), b) \vee \neg P(f(z))\Big). \end{aligned}$$

Observamos que las dos cláusulas no tienen variables en común.

Consideramos el literal $L_1 := R(f(x), b)$ de C_1 y el literal $L_2 := \neg R(f(z), b)$ de C_2. Un unificador principal para los literales L_1 y L_2^c, es decir, para el conjunto de literales

$$\Big\{R(f(x), b),\ R(f(z), b)\Big\},$$

es

$$\sigma = (z|x).$$

(También hubiese sido correcto utilizar el unificador principal para L_1 y L_2^c dado por $\sigma' = (x|z)$.)

Tenemos por una parte que

$$\sigma(C_1) \;=\; C_1 \;=\; R(f(x),b) \vee S(x,y) \vee P(a) \qquad \text{y} \qquad \sigma(L_1) \;=\; L_1 \;=\; R(f(x),b),$$

y por otra que

$$\sigma(C_2) \;=\; \neg R(f(x),b) \vee \neg P(f(x)) \qquad \text{y} \qquad \sigma(L_2) \;=\; \neg R(f(x),b).$$

De aquí,

$$\sigma(C_1) - \sigma(L_1) \;=\; S(x,y) \vee P(a),$$

y

$$\sigma(C_2) - \sigma(L_2) \;=\; \neg P(f(x)).$$

Por tanto obtenemos la resolvente binaria siguiente para C_1 y C_2:

$$S(x,y) \vee P(a) \vee \neg P(f(x)).$$

Nótese que no es posible resolver sobre los literales $P(a)$ y $\neg P(f(z))$ respectivos de C_1 y C_2, al no ser los literales $P(a)$ y $P(f(z))$ unificables. $\square$

Ejemplo 31. Sean las cláusulas

$$\begin{aligned} C_1 \equiv &\;\; R(f(x),b) \vee S(x,y) \vee P(a), \\ C_2 \equiv &\;\; \neg R(f(x),b) \vee \neg P(f(z)). \end{aligned}$$

Vemos que ambas cláusulas comparten variables, aunque en este ejemplo particular es posible calcular una resolvente binaria sobre los literales $L_1 := R(f(x),b)$ de C_1 y $L_2 := \neg R(f(x),b)$ de C_2. En este caso utilizamos el unificador principal identidad que transforma cada variable en ella misma. La resolvente binaria que obtenemos es

$$S(x,y) \vee P(a) \vee \neg P(f(z)).$$

$\square$

En el ejemplo siguiente se muestra por qué hemos de procurar que cláusulas distintas no compartan variables.

Ejemplo 32. Sean las cláusulas

$$\begin{aligned} C_1 \equiv &\;\; R(x,b) \vee Q(x,a), \\ C_2 \equiv &\;\; \neg R(a,x) \vee P(x). \end{aligned}$$

En el literal $R(x,b)$ de C_1 y en el literal $\neg R(a,x)$ de C_2 aparece x, y ello hace que no podamos encontrar un unificador principal para el conjunto de literales

$$\Big\{ R(x,b),\; R(a,x) \Big\},$$

pues el sistema de ecuaciones en términos

$$\begin{cases} x = a \\ b = x \end{cases}$$

es incompatible. Por consiguiente, de esta forma no podemos obtener una resolvente binaria para las cláusulas C_1 y C_2.

Si renombramos en C_2 cambiando el símbolo de variable x por z, resulta

$$\begin{aligned} C_1 \equiv &\quad R(x,b) \vee Q(x,a), \\ C_2 \equiv &\quad \neg R(a,z) \vee P(z). \end{aligned}$$

Tomando $L_1 := R(x,b), L_2 := \neg R(a,z)$ y $\sigma = (x|a, z|b)$, obtenemos la resolvente binaria

$$Q(a,a) \vee P(b).$$

$\square$

Es razonable pensar que si un conjunto de cláusulas Γ es insatisfacible, entonces de alguna forma ha de ser posible obtener la cláusula vacía calculando resolventes binarias a partir de cláusulas de Γ ó de otras resolventes binarias previamente calculadas.

Ejemplo 33. Sea el conjunto de cláusulas

$$\Gamma := \Big\{\neg P(x) \vee Q(x),\ \neg Q(y) \vee R(y),\ P(a),\ \neg R(a)\Big\}.$$

Intuitivamente vemos que Γ es insatisfacible, pues si escribimos las fórmulas de Γ como

$$\begin{aligned} &\forall x(P(x) \to Q(x)), \\ &\forall y(Q(y) \to R(y)), \\ &P(a), \\ &\neg R(a), \end{aligned}$$

observamos que a partir de las tres primeras se obtiene la conclusión $R(a)$, la cual es "incompatible" con la cuarta fórmula $\neg R(a)$. Dicho de otro modo, si Γ fuese satisfacible, entonces por el Lema 9 obtendríamos que el conjunto de cláusulas $\{R(a),\ \neg R(a)\}$ sería satisfacible, cosa que no es cierta.

Ahora vamos a obtener la cláusula vacía calculando resolventes binarias a partir de Γ. Para ello llamamos

$$\begin{aligned} C_1 &\equiv \neg P(x) \vee Q(x), \\ C_2 &\equiv \neg Q(y) \vee R(y), \\ C_3 &\equiv P(a), \\ C_4 &\equiv \neg R(a). \end{aligned}$$

De C_1 y C_3, con $\sigma_1 = (x|a)$, obtenemos la resolvente binaria

$$C_5 \equiv Q(a).$$

A partir de ésta y C_2, con $\sigma_2 = (y|a)$, obtenemos la resolvente binaria

$$C_6 \equiv R(a).$$

Por último, de C_4 y C_6, con $\sigma_3 = ()$, obtenemos

$$C_7 \equiv \square.$$

$\square$

Sin embargo, con la definición de resolvente que tenemos hasta este momento, existen conjuntos de cláusulas insatisfacibles a partir de los cuales no es posible obtener la cláusula vacía mediante el cálculo de resolventes binarias sucesivas.

Ejemplo 34. El conjunto de cláusulas

$$\Gamma := \Big\{ P(x) \vee P(y),\ \neg P(z) \vee \neg P(u) \Big\}$$

claramente es inconsistente pues $\Gamma \models P(a)$ y $\Gamma \models \neg P(a)$.

Sin embargo es imposible obtener la cláusula vacía mediante el cálculo de resolventes binarias sucesivas, pues según las definiciones, para calcular una resolvente binaria a partir de dos cláusulas, tomamos un literal en cada una de ellas y calculamos un unificador principal que aplicamos a ambas cláusulas. Finalmente los transformados de los literales seleccionados son suprimidos.

En nuestro caso, cualquier resolvente binaria que obtengamos usando cláusulas de Γ ó resolventes binarias previamente calculadas, consta siempre de dos literales. Vemos así que nunca obtendremos la cláusula vacía. Por ello necesitamos ampliar el concepto de resolvente. $\square$

Si dos o más literales de una cláusula C tienen un unificador principal σ, entonces se dice que $\sigma(C)$ es un **factor** de la cláusula C. También se suele decir que $\sigma(C)$ se obtiene a partir de C aplicando la regla de disminución. Si $\sigma(C)$ tiene un sólo literal, diremos que $\sigma(C)$ es un **factor unitario** de C.

Ejemplo 35. Sea la cláusula

$$C_1 \equiv P(x) \vee P(a) \vee Q(f(x), y)$$

y la substitución

$$\sigma_1 = (x|a).$$

Entonces la cláusula
$$\sigma_1(C_1) \equiv P(a) \vee Q(f(a), y)$$
es un factor de C_1. Así hemos unificado los literales $P(x)$ y $P(a)$ que aparecen en C_1 para obtener la cláusula $\sigma_1(C_1)$.

A partir de la cláusula
$$C_2 \equiv P(f(x)) \vee \neg Q(y, a) \vee P(y) \vee P(z)$$
podemos obtener varios factores según nos interese. Por ejemplo, una posibilidad consiste en unificar los literales $P(f(x))$ y $P(y)$ mediante la substitución
$$\sigma_2 = (y|f(x)),$$
para obtener el factor
$$\sigma_2(C_2) = P(f(x)) \vee \neg Q(f(x), a) \vee P(z).$$
Otra posibilidad consiste en unificar los literales $P(f(x))$, $P(y)$ y $P(z)$ mediante la substitución
$$\sigma_3 = (y|f(x), z|f(x)).$$
Resulta entonces la cláusula
$$\sigma_3(C_2) = P(f(x)) \vee \neg Q(f(x), a).$$
□

Lema 11. Si $\sigma(C)$ es un factor de una cláusula C, entonces $C \models \sigma(C)$. □

Ya estamos en condiciones de dar el concepto de resolvente que vamos a adoptar.

Una **resolvente** de dos cláusulas C_1 y C_2 es una de las siguientes resolventes binarias:

1. Una resolvente binaria de C_1 y C_2.
2. Una resolvente binaria de C_1 y un factor de C_2.
3. Una resolvente binaria de C_2 y un factor de C_1.
4. Una resolvente binaria de un factor de C_1 y un factor de C_2.

Ejemplo 36. Obtenemos todas las resolventes posibles para las cláusulas
$$\left.\begin{array}{rl} C_1 \equiv & P(x) \vee P(f(y)) \vee R(x, y) \\ C_2 \equiv & \neg P(f(g(a))) \vee Q(z) \end{array}\right\}.$$

1. Resolvente binaria de C_1 y C_2 sobre los literales respectivos $P(x)$ y $\neg P(f(g(a)))$. Usamos el unificador principal $\sigma = (x|f(g(a)))$ y obtenemos

$$P(f(y)) \vee R(f(g(a)), y) \vee Q(z).$$

2. Resolvente binaria de C_1 y C_2 sobre los literales respectivos $P(f(y))$ y $\neg P(f(g(a)))$. Usamos el unificador principal $\sigma = (y|g(a))$ y obtenemos

$$P(x) \vee R(x, g(a)) \vee Q(z).$$

3. Resolvente binaria de un factor de C_1 con la cláusula C_2. Previamente unificamos los literales

$$L_1 := P(x),\ L_2 := P(f(y))$$

de C_1 usando el unificador principal

$$\sigma = (x|f(y))$$

y obtenemos el factor de C_1 dado por

$$\sigma(C_1) \equiv P(f(y)) \vee R(f(y), y).$$

Finalmente calculamos la resolvente binaria de $\sigma(C_1)$ y C_2. Para ello usamos el unificador principal

$$\delta = (y|g(a))$$

y obtenemos

$$R(f(g(a)), g(a)) \vee Q(z),$$

que es una resolvente para C_1 y C_2.

Otra forma de obtener este mismo resultado consiste en calcular un unificador principal σ' para el conjunto de literales

$$\Big\{P(x),\ P(f(y)),\ P(f(g(a)))\Big\}.$$

Resulta

$$\sigma' = (x|f(g(a)), y|g(a)),$$

de donde

$$\sigma'(C_1) \equiv P(f(g(a))) \vee R(f(g(a)), g(a)),$$

$$\sigma'(C_2) \equiv \neg P(f(g(a))) \vee Q(z),$$

y de aquí la resolvente

$$R(f(g(a)), g(a)) \vee Q(z).$$

□

La siguiente propiedad es la generalización del Lema 10.

Lema 12. Si C es una resolvente de las cláusulas C_1 y C_2, entonces

$$\Big\{C_1, C_2\Big\} \;\models\; C.$$

□

A continuación damos la definición formal de **deducción de una cláusula a partir de un conjunto de cláusulas**, la cual ya la hemos comentado intuitivamente en algunos ejemplos anteriores.

> Dado un conjunto de cláusulas, $\Gamma \cup \{C\}$, una deducción de C a partir de Γ es una secuencia finita de cláusulas $C_1, C_2, \ldots, C_m$ tal que $C_m = C$ y para todo $i < m$, se cumple que C_i es un elemento de Γ ó bien es una resolvente de dos cláusulas C_j y C_k, con j y k ambos menores que i.

De acuerdo con la definición anterior, a las cláusulas pertenecientes a Γ se les llama hipótesis.

En la práctica, cuando escribamos una deducción, numeraremos y describiremos las fórmulas que la componen, ya sean hipótesis ó resolventes, debiendo en este último caso indicar las cláusulas intervinientes y el unificador principal utilizado.

Ejemplo 37. Consideramos el conjunto de cláusulas siguiente:

$$\Gamma := \Big\{\neg P(x) \vee Q(x),\ \neg Q(y) \vee R(y),\ P(a),\ \neg R(a)\Big\}.$$

La secuencia siguiente es una deducción de la cláusula $R(a)$ a partir del conjunto Γ:

1. $\neg P(x) \vee Q(x)$, hipótesis.
2. $P(a)$, hipótesis.
3. $Q(a)$, resolvente de (1.) y (2.); $\sigma_1 = (x|a)$.
4. $\neg Q(y) \vee R(y)$, hipótesis.
5. $R(a)$, resolvente de (3.) y (4.); $\sigma_2 = (y|a)$.

□

Ejemplo 38. Dado el conjunto de cláusulas

$$\Gamma := \Big\{\neg P(x) \vee Q(x),\ P(a)\Big\},$$

la secuencia siguiente es una deducción (muy poco interesante) de la cláusula $P(a)$ a partir del conjunto Γ:

1. $\neg P(x) \vee Q(x)$, hipótesis.
2. $P(a)$, hipótesis.
3. $P(a)$, hipótesis.

$\square$

Un **refutación para un conjunto de cláusulas** Γ es una deducción de la cláusula vacía a partir de Γ.

Un conjunto de cláusulas Γ se denomina **refutable**, si existe alguna refutación para Γ.

Ejemplo 39. Consideramos de nuevo el conjunto de cláusulas

$$\Gamma := \Big\{\neg P(x) \vee Q(x),\ \neg Q(y) \vee R(y),\ P(a),\ \neg R(a)\Big\}.$$

La secuencia siguiente es una refutación para Γ:

1. $\neg P(x) \vee Q(x)$, hipótesis.
2. $P(a)$, hipótesis.
3. $Q(a)$, resolvente de (1.) y (2.); $\sigma_1 = (x|a)$.
4. $\neg Q(y) \vee R(y)$, hipótesis.
5. $R(a)$, resolvente de (3.) y (4.); $\sigma_2 = (y|a)$.
6. $\neg R(a)$, hipótesis.
7. $\square$, resolvente de (5.) y (6.); $\sigma_3 = ()$.

Por tanto, Γ es refutable. $\square$

Ejemplo 40. El conjunto de cláusulas siguiente no es refutable:

$$\Gamma := \Big\{\neg P(x) \vee Q(x),\ P(a),\ \neg Q(b)\Big\}.$$

Si intentamos encontrar una refutación para Γ y comenzamos con la secuencia

1. $\neg P(x) \vee Q(x)$, hipótesis.
2. $P(a)$, hipótesis.
3. $Q(a)$, resolvente de (1.) y (2.); $\sigma_1 = (x|a)$.
4. $\neg Q(b)$, hipótesis.
5. $\neg P(b)$, resolvente de (1.) y (4.); $\sigma_2 = (x|b)$.

vemos que ya no podemos obtener nuevas resolventes. Como no hemos obtenido $\square$, el conjunto no es refutable. $\square$

El Principio de resolución es una regla de inferencia ó de deducción que consiste en generar resolventes a partir de un conjunto de cláusulas. Así, para demostrar la inconsistencia de un conjunto de cláusulas intentaremos encontrar una deducción de $\square$ a partir del conjunto dado. Esta idea se sustenta en el siguiente resultado.

Teorema 5. (**Completitud del Principio de resolución**). Un conjunto de cláusulas Γ es inconsistente, si y sólo si, Γ es refutable, es decir, existe una deducción de $\square$ a partir de Γ. $\square$

Ejemplo 41. En el Ejemplo 34 hemos estudiado el conjunto de cláusulas

$$\Gamma := \Big\{P(x) \vee P(y),\ \neg P(z) \vee \neg P(u)\Big\},$$

para el cual hemos argumentado que es inconsistente. Ahora, basándonos en el Teorema de Completitud del principio de resolución, obtenemos esta misma conclusión. Para ello basta considerar la siguiente refutación para Γ:

1. $P(x)$, factor de la hipótesis $P(x) \vee P(y)$; $\sigma_1 = (y|x)$.
2. $\neg P(z)$, factor de la hipótesis $\neg P(z) \vee \neg P(u)$; $\sigma_2 = (u|z)$.
3. $\square$, resolvente de (1.) y (2.); $\sigma_3 = (z|x)$.

$\square$

Ejemplo 42. Consideramos el conjunto de cláusulas

$$\Gamma := \Big\{R(a,x) \vee \neg P(x),\ \neg R(y,y) \vee P(y)\Big\},$$

y llamamos

$$C_1 \equiv R(a,x) \vee \neg P(x),$$

$$C_2 \equiv \neg R(y,y) \vee P(y).$$

De C_1 y C_2, resolvemos sobre los literales $R(a,x)$ y $\neg R(y,y)$, usando $\sigma_1 = (x|a,\ y|a)$, y obtenemos

$$C_3 \equiv \neg P(a) \vee P(a).$$

De C_1 y C_2, resolvemos sobre los literales $\neg P(x)$ y $P(y)$, usando $\sigma_2 = (y|x)$, y obtenemos

$$C_4 \equiv R(a,x) \vee \neg R(x,x).$$

Nótese que $\square$ no se puede obtener como resolvente de C_1 y C_2.

De hecho, Γ es satisfacible. Para verlo, consideramos la estructura $\mathcal{E}$ tal que:

$$\begin{aligned}
&\mathrm{D}_{\mathcal{E}} := \{1,2\} \subseteq \mathbb{N},\\
&a^{\mathcal{E}} := 1,\\
&P^{\mathcal{E}}(r) = \mathbf{1} \text{ si y sólo si } r = 2,\\
&R^{\mathcal{E}}(r,s) = \mathbf{1} \text{ si y sólo si } (r,s) \in \{(1,2),(2,2)\}.
\end{aligned}$$

Obsérvese que

$$C_1 \equiv \forall x\Big(P(x) \to R(a,x)\Big),$$

$$C_2 \equiv \forall y\Big(R(y,y) \to P(y)\Big).$$

Entonces, para cualquier asignación v en $\mathcal{E}$ tenemos que

$$\mathrm{I}^v\left(C_1\right) = \mathrm{I}^v\left(C_2\right) = \mathbf{1}.$$

$\square$

Ejemplo 43. Consideramos de nuevo el conjunto de cláusulas del Ejemplo 40 el cual hemos visto que no es refutable:

$$\Gamma := \Big\{\neg P(x) \vee Q(x),\ P(a),\ \neg Q(b)\Big\}.$$

Por el Teorema 5, Γ es consistente. $\square$

Ejemplo 44. Sea el conjunto de cláusulas

$$\Gamma := \Big\{R(a,x) \vee P(x),\ \neg R(f(y),y) \vee P(y),\ \neg P(a)\Big\}.$$

Llamamos

$$\begin{aligned}
C_1 &\equiv R(a,x) \vee P(x),\\
C_2 &\equiv \neg R(f(y),y) \vee P(y),\\
C_3 &\equiv \neg P(a).
\end{aligned}$$

Nótese que cláusulas distintas no tienen variables en común.

De C_1 y C_2, no podemos obtener ninguna resolvente puesto que el sistema siguiente de ecuaciones en términos,

$$\begin{cases} a = f(y)\\ x = y \end{cases}$$

es incompatible.

De C_1 y C_3, resolvemos sobre los literales $P(x)$ y $\neg P(a)$, usando $\sigma_1 = (x|a)$, y obtenemos

$$C_4 \equiv R(a,a).$$

De C_2 y C_3, resolvemos sobre los literales $P(y)$ y $\neg P(a)$, usando $\sigma_2 = (y|a)$, y obtenemos

$$C_5 \equiv \neg R(f(a), a).$$

En este punto del proceso, observamos que C_4 no se puede resolver con ninguna de las cláusulas C_1, C_2, C_3, C_5. Tampoco es posible resolver C_5 con ninguna de las cláusulas C_1, C_2, C_3, C_4.

Así pues, no es posible generar nuevas resolventes. Como $\Box$ no pertenece a Γ ni tampoco ha sido obtenida como resolvente, basándonos en el Teorema de Completitud del principio de resolución, concluimos que Γ es satisfacible ó consistente. $\Box$

En el Tema 4 estudiamos el Algoritmo de Davis-Putnam que permite decidir si un conjunto de cláusulas de la Lógica proposicional es o no insatisfacible. Como la Lógica de proposiciones se generaliza mediante la Lógica de predicados, es posible utilizar el Principio de resolución como método alternativo para decidir la insatisfacibilidad de un conjunto de cláusulas de la Lógica proposicional. Ahora las substituciones utilizadas son todas iguales a la substitución identidad $\sigma = ()$.

Ejemplo 45. Comprobemos mediante el Principio de resolución que, tal y como vimos en el Tema 4, el conjunto de cláusulas siguiente es insatisfacible,

$$\Sigma = \Big\{ P \vee Q,\ P \vee \neg Q,\ \neg P \vee R,\ \neg P \vee \neg R \Big\}.$$

Para ello basta construir la siguiente refutación para Σ:

1. $P \vee Q$, hipótesis.
2. $P \vee \neg Q$, hipótesis.
3. P, resolvente de (1.) y (2.).
4. $\neg P \vee R$, hipótesis.
5. $\neg P \vee \neg R$, hipótesis.
6. $\neg P$, resolvente de (4.) y (5.).
7. $\Box$, resolvente de (3.) y (6.).

Por el Principio de resolución, Σ es insatisfacible. $\Box$

Ejemplo 46. En el Tema 4 vimos mediante el Algoritmo de Davis-Putnam que el conjunto siguiente de cláusulas es satisfacible,

$$\Sigma = \Big\{ P \vee \neg Q,\ \neg P \vee R,\ \neg P \vee \neg R \Big\}.$$

Ahora obtenemos esta misma conclusión aplicando el Principio de resolución. Para ello llamamos

$$C_1 := P \vee \neg Q, \quad C_2 := \neg P \vee R, \quad C_3 := \neg P \vee \neg R.$$

Primero calculamos todas las resolventes posibles entre ellas.

- $C_4 := \neg Q \vee R$, resolvente de C_1 y C_2.
- $C_5 := \neg Q \vee \neg R$, resolvente de C_1 y C_3.
- $C_6 := \neg P$, resolvente de C_2 y C_3.

Ahora calculamos todas las resolventes posibles entre C_4 y alguna cláusula del conjunto $\{C_1, C_2, C_3, C_5, C_6\}$.

- $C_7 := \neg Q \vee \neg P$, resolvente de C_4 y C_3.
- $C_8 := \neg Q$, resolvente de C_4 y C_5.

Ahora todas las resolventes de C_5 con alguna cláusula del conjunto $\{C_1, C_2, C_3, C_6\}$. Sólo se obtiene C_7 como resolvente de C_5 con C_2.

A continuación obtenemos todas las resolventes de C_6 con alguna cláusula del conjunto $\{C_1, C_2, C_3\}$. Sólo se obtiene C_8 como resolvente de C_6 con C_1.

Por tanto, hasta este momento hemos calculado todas las resolventes posibles entre dos cláusulas del conjunto $\{C_1, C_2, C_3, C_4, C_5, C_6\}$. A continuación intentamos obtener una resolvente de C_7 con cada una de las cláusulas del conjunto $\{C_1, C_2, C_3, C_4, C_5, C_6\}$, y análogamente para C_8. También hemos de intentar resolver C_7 con C_8.

De todas éstas posibilidades, sólo se puede calcular la resolvente de C_7 con C_1, que produce de nuevo C_8.

Por tanto no se pueden generar nuevas resolventes. Como no hemos obtenido la cláusula vacía, por el Principio de resolución, el conjunto Σ es satisfacible. $\square$

5 Estrategias de resolución

Si partimos de un conjunto de cláusulas Γ y queremos estudiar si es satisfacible ó insatisfacible, necesitamos una **estrategia** que nos vaya guiando en el proceso de obtención de resolventes.

Como un primer paso, suprimiremos las cláusulas tautológicas que aparezcan en el conjunto inicial, así como aquellas que se vayan obteniendo.

La estrategia más simple para generar resolventes, es la llamada **estrategia de saturación**. Su descripción es la siguiente:

1. $\Gamma_0 := \Gamma$.
2. Para cada $i \geq 0$, calcular el conjunto Δ_i formado por todas las resolventes obtenibles a partir de dos cláusulas cualesquiera $C_{i,j}, C_{i,k} \in \Gamma_i$, distintas. Definir $\Gamma_{i+1} := \Gamma_i \cup \Delta_i$.
3. Si $\square \in \Gamma_{i+1}$, entonces se detiene el proceso y se concluye que Γ es insatisfacible.

4. Si $\Gamma_{i+1} = \Gamma_i$, entonces se detiene el proceso y se concluye que Γ es satisfacible.

5. Hacer $i := i + 1$, y volver al paso (2.).

$\square$

La estrategia de saturación suele ser muy ineficiente, pues es posible que obtengamos una misma cláusula varias veces además de muchas cláusulas inútiles de cara a la obtención de la cláusula vacía.

Esta estrategia la hemos usado ya en los Ejemplos 40 y 46.

Otra posibilidad que puede presentarse en la práctica, es que sea posible generar una infinidad de resolventes distintas. En este caso, si no se ha obtenido $\square$, el proceso debería detenerse en algún momento, y concluir que muy probablemente Γ sea satisfacible.

Ello nos lleva a que hemos de poner algún tipo de cota ó límite sobre el máximo número de resolventes a generar.

Ejemplo 47. Sea el conjunto de cláusulas

$$\Gamma := \Big\{\neg P(x) \vee P(f(x)),\ P(a)\Big\}.$$

A partir de Γ resulta la sucesión de resolventes

$$P(f(a)),\ P(f(f(a))),\ P(f(f(f(a)))),\ P(f(f(f(f(a))))), \ldots.$$

Se puede comprobar que Γ es satisfacible, con lo cual nunca obtendremos $\square$. Por tanto, tras generar un número suficientemente grande de resolventes, concluímos que muy probablemente Γ es satisfacible. $\square$

Entre otras ideas que se pueden utilizar para generar resolventes, mencionamos las siguientes:

1. *Elección de la cláusula más corta.* En cada momento del proceso, de entre todas las resolventes posibles que se pueden obtener, se calcula una cuyo número de literales sea el menor posible.

2. *Preferencia de cláusulas unitarias.* En cada momento del proceso, se intenta que la resolvente obtenida sea tal que alguna de las cláusulas intervinientes en su obtención conste de un sólo literal.

3. *Técnica del conjunto soporte.* Si estamos intentando dar respuesta al problema $\Omega \models \alpha$, y para ello queremos saber si el conjunto de fórmulas $\Omega \cup \{\neg\alpha\}$ es insatisfacible, obtenemos un conjunto de cláusulas Γ a partir de las fórmulas en $\Omega \cup \{\neg\alpha\}$. Supongamos que Δ es el subconjunto de Γ formado por aquellas cláusulas que proceden de la fórmula $\neg\alpha$. Como Ω se supone satisfacible, ya que de lo contrario sería inmediato que $\Omega \models \alpha$, la idea de esta estrategia consiste en que toda resolvente obtenida se base en alguna cláusula que proceda, aunque sea de forma indirecta, del subconjunto Δ.

Estas ideas también pueden ser aplicadas de forma combinada.

Establecemos en este punto la metodología que se deduce de todas las ideas expuestas en este tema:

Para probar que un conjunto de fórmulas Ω *implica semánticamente a una fórmula* α, *tratamos de ver que el conjunto de fórmulas* $\Omega\cup\{\neg\alpha\}$ *es inconsistente. Para ello, obtenemos un conjunto de formas clausuladas correspondientes, y a partir de éste, un conjunto de cláusulas* Γ. *Entonces aplicamos el Principio de resolución a* Γ *tratando de obtener una refutación para él.*

Este método para probar una implicación semántica se denomina **refutación por resolución**.

Ejemplo 48. Queremos demostrar la siguiente implicación semántica mediante una refutación por resolución:

$$\Big\{\forall x(R(x) \to R(f(x))),\ R(a),\ \forall xP(x) \vee \forall xQ(x)\Big\} \models \forall x\big(P(x) \vee Q(x)\big).$$

Para ello consideramos el conjunto

$$\Sigma = \Big\{\forall x(R(x) \to R(f(x))),\ R(a),\ \forall xP(x) \vee \forall xQ(x),\ \neg\forall x\big(P(x) \vee Q(x)\big)\Big\},$$

el cual queremos ver que es inconsistente. A partir de éste obtenemos un conjunto de formas clausuladas correspondientes:

$$\Sigma'' = \Big\{\forall x(\neg R(x) \vee R(f(x))),\ R(a),\ \forall y\forall z(P(y) \vee Q(z)),\ \neg P(a) \wedge \neg Q(a)\Big\}.$$

Finalmente llegamos al conjunto de cláusulas

$$\Gamma = \Big\{\neg R(x) \vee R(f(x)),\ R(a),\ P(y) \vee Q(z),\ \neg P(a),\ \neg Q(a)\Big\}.$$

Si aplicamos la estrategia de elección de la cláusula más corta o bien la de preferencia de cláusulas unitarias, podríamos calcular, entre otras posibilidades, una resolvente a partir de las cláusulas

$$\neg R(x) \vee R(f(x)) \qquad \text{y} \qquad R(a).$$

Obtenemos la cláusula $R(f(a))$, a partir de la cual podríamos generar también las resolventes

$$R(f(f(a))),\ R(f(f(f(a)))),\ R(f(f(f(f(a))))), \ldots,$$

sin llegar nunca a la cláusula vacía (Véase el Ejemplo 47).

Es preferible aplicar la Técnica del conjunto soporte, y hacer intervenir en nuestros cálculos de resolventes siempre alguna cláusula que proceda directa ó indirectamente de las cláusulas obtenidas al negar la fórmula $\forall x\big(P(x) \vee Q(x)\big)$ que queremos demostrar.

Así pues, calculamos una resolvente a partir de las cláusulas

$$P(y) \vee Q(z) \qquad \text{y} \qquad \neg P(a),$$

y obtenemos la cláusula $Q(a)$, la cual sólo se puede resolver con la cláusula $\neg Q(a)$. Llegamos de este modo a $\square$, probando así la inconsistencia de Γ y por tanto la implicación semántica propuesta. $\square$

Cuando partimos de un conjunto de cláusulas Γ, y comenzamos a generar resolventes, estamos generando un árbol, llamado el **árbol de las deducciones**. La forma en la que vamos obteniendo resolventes determina la forma en la que vamos recorriendo dicho árbol. Por ejemplo, la estrategia de saturación estudiada anteriormente, da lugar a lo que se denomina **exploración del árbol primero en anchura**. Otra posibilidad, es la llamada **exploración del árbol primero en profundidad**, que se produce cuando al obtener la resolvente $C_{i,j+1}$ se utiliza $C_{i,j}$, y así para todo $j \geq 1$. Si no hemos obtenido la cláusula vacía, podemos retroceder hasta otro nodo más arriba, en el que aplicamos de nuevo la misma idea. En este caso se habla de **exploración del árbol primero en profundidad con retroceso**.

Al margen de estas cuestiones algorítmicas, otra alternativa consiste en restringir el concepto de deducción.

> Una **deducción lineal-input** de una cláusula C a partir de un conjunto de cláusulas Γ, es una secuencia finita de cláusulas $C_1, C_2, \ldots, C_m$, tal que $C_m = C$ y para todo i, con $2 \leq i \leq m$, se cumple que C_i es una resolvente calculada a partir de C_{i-1} y otra cláusula perteneciente a Γ.

Nótese que la propia definición de deducción lineal-input conlleva que el árbol de las deducciones sea recorrido primero en profundidad.

Consideramos de nuevo el Ejemplo 37 en el cual, sin darnos cuenta, hicimos una deducción lineal-input de la cláusula $R(a)$ a partir del conjunto Γ dado.

Tenemos el conjunto de cláusulas

$$\Gamma := \Big\{ \neg P(x) \vee Q(x),\ \neg Q(y) \vee R(y),\ P(a),\ \neg R(a) \Big\},$$

y la secuencia siguiente de cláusulas:

1. $\neg P(x) \vee Q(x)$, hipótesis.
2. $P(a)$, hipótesis.
3. $Q(a)$, resolvente de (1.) y (2.); $\sigma_1 = (x|a)$.
4. $\neg Q(y) \vee R(y)$, hipótesis.

5. $R(a)$, resolvente de (3.) y (4.); $\sigma_2 = (y|a)$.

Una **refutación lineal-input** de un conjunto de cláusulas Γ, es una deducción lineal-input de la cláusula vacía a partir de Γ.

Veamos de nuevo el Ejemplo 39 en el que se lleva a cabo una refutación lineal-input para el conjunto Γ dado.

Teníamos

$$\Gamma := \Big\{\neg P(x)\vee Q(x),\ \neg Q(y)\vee R(y),\ P(a),\ \neg R(a)\Big\},$$

y la secuencia siguiente de cláusulas que finaliza en $\square$:

1. $\neg P(x)\vee Q(x)$, hipótesis.
2. $P(a)$, hipótesis.
3. $Q(a)$, resolvente de (1.) y (2.); $\sigma_1 = (x|a)$.
4. $\neg Q(y)\vee R(y)$, hipótesis.
5. $R(a)$, resolvente de (3.) y (4.); $\sigma_2 = (y|a)$.
6. $\neg R(a)$, hipótesis.
7. $\square$, resolvente de (5.) y (6.); $\sigma_3 = ()$.

6 Conjuntos de Horn

Un literal de la forma $R(t_1, \ldots, t_n)$ se dice que es un **literal positivo**, mientras que un literal de la forma $\neg R(t_1, \ldots, t_n)$ se denomina un **literal negativo**.

Un **cláusula negativa** es aquella en la que todos sus literales son negativos.

Un **cláusula de Horn** es la que tiene exactamente un literal positivo, y todos los literales restantes, si los hay, son negativos.

Un **conjunto de Horn** es aquel de la forma $\{C_0\} \cup \Delta$, donde C_0 es una cláusula negativa y Δ es un conjunto de cláusulas de Horn.

Ejemplo 49. Con las notaciones usuales, las siguientes cláusulas son negativas:

$$\neg R(f(x,y),z),\quad \neg Q(a),\quad \neg R(f(x,y),z)\vee\neg P(b),\quad \neg R(y,z)\vee\neg P(b)\vee\neg R(a,g(g(y))).$$

Las siguientes son cláusulas de Horn:

$$R(x,g(y)),\quad P(x)\vee\neg Q(h(x)),\quad R(x,a)\vee\neg P(y)\vee\neg Q(b)\vee\neg R(z,g(z)).$$

Ninguna de las cláusulas siguientes es de Horn:

$$P(a) \vee Q(y),\quad \neg R(a,y) \vee \neg Q(g(z)).$$

Todos los conjuntos siguientes son de Horn:

- $\Big\{\neg Q(x) \vee \neg P(a)\Big\}$.
- $\Big\{\neg Q(x) \vee \neg P(a),\ Q(y) \vee \neg R(y, h(x))\Big\}$.
- $\Big\{\neg Q(x) \vee \neg P(a),\ Q(z) \vee \neg R(y, h(x)),\ R(a, b) \vee \neg P(z) \vee \neg Q(f(x, y))\Big\}$.

El conjunto $\Big\{\neg Q(x) \vee \neg P(a),\ Q(a) \vee \neg R(y, h(x)) \vee P(z)\Big\}$ no es de Horn. □

Observamos que una resolvente obtenida a partir de una cláusula negativa y una cláusula de Horn, es de nuevo una cláusula negativa.

El siguiente resultado establece la importancia de los conjuntos de Horn.

Proposición 3. Un conjunto de Horn $\{C_0\} \cup \Delta$ es insatisfacible, si y sólo si, existe una refutación lineal-input para $\{C_0\} \cup \Delta$ que comienza en la cláusula C_0. □

En el Ejemplo 39 ya vimos un conjunto de Horn y una refutación lineal-input para él.

Hay conjuntos de cláusulas que no son de Horn para los que existen refutaciones lineales-input, como por ejemplo,

$$\Gamma = \Big\{P \vee Q,\ \neg P,\ \neg Q\Big\}.$$

Una refutación lineal-input para Γ es:

1. $P \vee Q$, hipótesis.
2. $\neg P$, hipótesis.
3. Q, resolvente de (1.) y (2.).
4. $\neg Q$, hipótesis.
5. $\square$, resolvente de (3.) y (4.).

Los conceptos estudiados en este tema forman la base del sistema PROLOG, que es un lenguaje de programación ideado a principios de los años 70 en la Universidad de Aix-Marseille I (en Marsella, Francia) por Alain Colmerauer y Philippe Roussel. El nombre PROLOG proviene del francés *PROgrammation en LOGique*.

PROLOG trabaja con conjuntos de Horn con los que trata de obtener refutaciones lineales-input, también llamadas refutaciones lineales-ordenadas. Para ello aplica una estrategia de búsqueda denominada *búsqueda primero en profundidad con retroceso*.