Lógica de Primer orden
y Demostración Automática de Teoremas

Fco. M. García Olmedo
Universidad de Granada
España

16 de diciembre de 2017

Resumen

Contiene una introducción general a los conceptos fundamentales de lógica de primer orden. Se fundamenta, como aplicación, la demostración automática de teoremas.

Índice

1. Lenguajes de Primer Orden **2**

2. Interpretaciones, satisfacibilidad y verdad **4**

3. Ejercicios sobre Lenguajes e Interpretaciones **8**

4. Forma Normal Prenexa **15**

5. Forma Normal de Skolem **18**

6. Teorema de Herbrand **20**

7. Ejercicios sobre Forma Prenexa **23**

8. Algoritmo de Unificación **25**

9. Resolución **28**

10. Ejemplos **31**

11. Gestión por Saturación **35**

12. Gestión por Saturación con Simplificación **35**

13. Gestión por Preferencia de Cláusulas Simples **36**

14. Exploración. Conceptos Generales **37**

15. Estrategias Lineales **38**

16. Estrategia "Input" **40**

17. Estrategias Ordenadas **42**

18.Estrategias Input Ordenadas y Extracción de Respuestas **44**

19.Ejercicios sobre Resolución **45**

Índice de figuras

1. Lenguajes de Primer Orden

Un *lenguaje de primer orden*, **L**, consta de los siguientes símbolos:

- símbolos de variable: $x_1, \ldots, x_n, \ldots$ y a veces x, y u, v, w (últimas letras minúsculas del alfabeto latino). Todos ellos forman un conjunto infinito numerable representado por V.
- símbolos de constante: $a_1, \ldots, a_n, \ldots$ y a veces a, b y c (primeras letras minúsculas del alfabeto latino). Todos ellos forman un conjunto representado por C, que podría ser vacío.
- símbolos de función: f_k^n (n y k son números naturales y n es no nulo) y a veces f, g y h. Todos ellos forman un conjunto representado por F.
- símbolos de relación: r_k^n (n y k son números naturales no nulos) y a veces r, s, p y q. El conjunto de todos ellos es representado por el símbolo R.
- símbolos lógicos: $\rightarrow$, $\neg$, $\vee$, $\wedge$, $\leftrightarrow$, $\forall$ y $\exists$
- símbolos auxiliares:), (y el signo ,

y llamamos *expresión* de **L** a cada sucesión finita de sus símbolos. En estas notas supondremos, a menos que digamos lo contrario, que los conjuntos de símbolos de constante, función y relación son todos y cada uno de ellos numerables. Un lenguaje que cumpla esto será denominado *lenguaje numerable*.

El número natural n que aparece como superíndice de los símbolos de función o de predicado indica su "ariedad", mientras que el subíndice es únicamente una marca distintiva de símbolos.

Ejemplo 1.1. Representaremos por $\mathbf{L_A}$ al lenguaje para el que $\mathrm{C} = \{c\}$, $\mathrm{F} = \{f^1, g^2, h^2\}$ y $\mathrm{R} = \{r^2\}$ (r^2 está destinada a "encarnar" la igualdad). Para $\mathbf{L_A}$ la sucesión finita de símbolos $axyzfg(r(ah$ es una expresión. Hemos de considerar también como ejemplo a la *expresión vacía*.

Entre las expresiones de **L** destacamos tres géneros: *términos* y *fórmulas atómicas* y *fórmulas*. Los *términos* están definidos por los siguientes items:

1. Son términos los símbolos de constante
2. Son términos los símbolos de variable
3. Si f_k^n es un símbolo de función y $t_1, \ldots, t_n$ son términos, entonces $f_k^n(t_1, \cdots, t_n)$ es un término
4. No hay otros términos aparte de los nombrados en los apartados anteriores.

El conjunto de los términos de **L** es representado por $\mathrm{Term}(\mathbf{L})$.

Las *fórmulas atómicas* de **L** son las expresiones de la forma $r_k^n(t_1, \cdots, t_n)$, donde r_k^n es un símbolo de predicado y $t_1, \ldots, t_n$ son términos. El conjunto de las fórmulas atómicas de **L** es representado por $\mathrm{Atom}(\mathbf{L})$.

Las *fórmulas* de **L** son las expresiones defininidas por las siguientes condiciones:

1. Las fórmulas atómicas son fórmulas

2. Si α es una fórmula, entonces $(\neg\alpha)$ es una fórmula
3. Si α y β son fórmulas, entonces $(\alpha \to \beta)$, $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \leftrightarrow \beta)$ son fórmulas
4. Si α es una fórmula y x es una variable, entonces $\big((\forall x)\alpha\big)$, $\big((\exists x)\alpha\big)$ es una fórmula
5. No hay otras fórmulas aparte de las antes descritas

El conjunto de las fórmulas de $\mathbf{L}$ es representado por $\mathrm{Form}(\mathbf{L})$.

Ejemplo 1.2. Para el lenguaje $\mathbf{L}$ del ejemplo 1.1 mostramos a modo de ejemplo los siguientes términos: c, x, $f(x)$, $g(x,c)$, $g\big(f(c),c\big)$ y $h\big(g(x,c),f(f(c))\big)$. Como fórmulas atómicas podemos mostrar: $r(x,c)$, $r(x,y)$, $r\big(g(x,c),h\big(g(x,c),f(f(c))\big)\big)$. Como fórmulas podemos poner de ejemplo a: $(r(x,y) \to r(c,f(c)))$, $\big(\neg r(x,y)\big)$, $\big((\forall x)(r(x,y) \to r(c,f(c)))\big)$, $\big((\forall y)((\forall x)(r(x,c) \to r(y,f(c))))\big)$ y, por qué no, $\big(\neg((\forall x)r(c,f(c)))\big)$.

Se puede demostrar que cualquier fórmula (resp. fórmula atómica, término) no puede ser escrita más que en una forma. Ello nos permitirá realizar razonamientos por *inducción* sobre fórmulas.

En la fórmula $\big((\forall x)\alpha\big)$ (resp. $\big((\exists x)\alpha\big)$), la fórmula α es denominada *radio de acción* o *ámbito* de $\forall x$ (resp. $\exists x$). Una ocurrencia de un símbolo de variable x en la fórmula α es *ligada* si en la escritura de α dicha ocurrencia está inmediatamente precedida de un cuantificador $\forall$ o de un cuantificador $\exists$ o bien la ocurrencia tiene lugar en el radio de acción de un cuantificador "$\forall x$" o "$\exists x$". La ocurrencia del símbolo de variable es *libre* cuando no es ligada. Un símbolo de variable es libre (resp. ligado) en una fórmula α cuando tiene una ocurrencia libre (resp. ligada) en α. Observar que un símbolo de variable dado puede ser simultáneamente libre y ligado en una fórmula. En lo sucesivo, si φ es una fórmula y $v_{i_1},\dots,v_{i_k} \in \mathrm{V}$, la expresión $\varphi(v_{i_1},\dots,v_{i_k})$ es una representación de φ e indicará que algunas de las variables $v_{i_1},\dots,v_{i_k}$ son variables libres en φ. Esto no significa que φ contenga a estas variables como variables libres, ni significa que φ no contenga otras variables libres.

Llamamos *sentencia* a toda aquella fórmula en la que cada ocurrencia de cada una de sus variables es ligada. El conjunto de las sentencias del lenguaje será representado por $\mathrm{Sent}(\mathbf{L})$.

Ejemplo 1.3. Sea φ la fórmula $\forall x\big(r(g(x,a),y)\big) \to r\big(g(x,f(y)),f(g(x,y))\big)$, el radio de acción del único $\forall x$ que aparece es $r(g(x,a),y)$ que llamaremos α. En φ es ligada la única ocurrencia de x que hay en α, y ello por estar en el radio de acción de un cuantificador $\forall x$; sin embargo, la ocurrencia de y en α no es ligada en φ pues aún estando en el radio de acción de un cuantificador no lo está en el radio de acción de un cuantificador $\forall y$. Observar que ninguna ocurrencia de las variables en α son ligadas en α; pues en α no ocurre cuantificador alguno.

Sea ψ la fórmula $r(g(x,f(y)),f(g(x,y)))$. En ella no hay ninguna ocurrencia de variables que sea ligada, dada la ausencia de cuantificadores, y lo mismo podemos decir de las ocurrencias de variables en ψ como subfórmula de φ.

Ejemplo 1.4. Sea ϕ la fórmula $\forall y\big(\forall x(r(g(x,a),y)) \to r\big(g(x,f(y)),f(g(x,y))\big)\big)$. En ella no hay ninguna ocurrencia de y que sea libre. ϕ no es una sentencia, aunque sí lo son $\forall x\forall y\big(\forall x(r(g(x,a),y)) \to r\big(g(x,f(y)),f(g(x,y))\big)\big)$, $\forall x(r(g(x,a),x)) \to \forall x\forall y\big(r(g(x,f(y)),f(g(x,y)))\big)$ y $q\big(g(f(a)),f(f(a))\big)$. En esta última fórmula no hay ninguna ocurrencia de variables, lo cual hace que sea sentencia; en las dos anteriores sí las hay, si bien ninguna es libre.

Ejemplo 1.5. En la sentencia $\forall x\big(q(x) \vee \neg\exists x r(a,x)\big)$ puede pensarse que la ocurrencia de x en $r(a,x)$ está "dóblemente ligada" cuando realmente no es así. Ha sido convenido, con bastante fundamento, que si una ocurrencia de un símbolo de variable está en el radio de acción de varios cuantificadores que pueden ligarla, es el más interno el que la liga mientras que el resto no tienen sobre ella el más mínimo efecto.

Ejercicio 1.1. Entender las siguientes fórmulas y decir de cada ocurrencia de variable, si es libre o ligada:

1. $\forall z\big(\forall x r(x,y) \to r(z,a)\big)$
2. $\forall y r(z,y) \to \forall z r(z,y)$
3. $\forall y\exists x s(x,y,g(x,y)) \vee \neg\forall x r(y,f(x))$

2. Interpretaciones, satisfacibilidad y verdad

Sea $\mathbf{L}$ un lenguaje de primer orden. Una *estructura* $\mathbf{A}$ para $\mathbf{L}$ consta de:

1. Un conjunto no vacío A, denominado *dominio* o *universo* de la estructura.
2. Por cada símbolo de constante a_i de $\mathbf{L}$, un elemento fijo $(a_i)^{\mathbf{A}}$ de A.
3. Por cada símbolo de función f_j^n de $\mathbf{L}$, una operación n-aria $(f_j^n)^{\mathbf{A}}$ en A (es decir, una función de A^n en A).
4. Por cada símbolo de predicado r_k^n de $\mathbf{L}$, una relación n-aria $(r_k^n)^{\mathbf{A}}$ en A (es decir, un subconjunto de A^n).

Ejemplo 2.1. Consideremos el lenguaje $\mathbf{L_A}$ y para él, a modo de ejemplo, la estructura $\mathbf{N}$ que consta de:

1. como universo de la estructura, los números naturales
2. $(c)^{\mathbf{N}} = 0$
3. $(f)^{\mathbf{N}}$, definida por $(f)^{\mathbf{N}}(m) = m'$
4. $(g)^{\mathbf{N}}$, definida por $(g)^{\mathbf{N}}(m,n) = m + n$
5. $(h)^{\mathbf{N}}$, definida por $(h)^{\mathbf{N}}(m,n) = mn$
6. $(r)^{\mathbf{N}}$, coincidente con la igualdad

Sea $\mathbf{A}$ una estructura para un lenguaje $\mathbf{L}$ y sea A el dominio de $\mathbf{A}$. Una *asignación* (de V en $\mathbf{A}$) es cualquier aplicación $s\colon \mathrm{V} \longrightarrow A$. A partir de una asignación s definimos una aplicación $\overline{s}\colon \mathrm{Term}(\mathbf{L}) \longrightarrow A$ como sigue:

1. si $t = a_j$, entonces $\overline{s}(t) = (a_j)^{\mathbf{A}}$
2. si $t = x_j$, entonces $\overline{s}(t) = s(x_j)$
3. si f_k^n es un símbolo de función, $(f_j^n)^{\mathbf{A}}$ es su correspondiente operación en A y $t_1, \ldots, t_n$ son términos, entonces $\overline{s}(f_k^n(t_1 \cdots t_n)) = (f_j^n)^{\mathbf{A}}(\overline{s}(t_1), \ldots, \overline{s}(t_n))$

Al par formado por una estructura para un lenguaje $\mathbf{L}$ y una asignación de V en $\mathbf{A}$ le llamamos *interpretación* de $\mathbf{L}$.

Definición 2.1. Sea $\langle \mathbf{A}, s\rangle$ una interpretación del lenguaje $\mathbf{L}$, definimos $I^s_{\mathbf{A}}$ sobre las fórmulas tomando valores en $\{0,1\}$ como sigue:

1. Para cualesquiera términos $t_1, \ldots, t_n$ de $\mathbf{L}$ y símbolo de relación r_k^n, $I^s_{\mathbf{A}}(r_k^n(t_1, \ldots, t_n)) = 1$ sii, por definición, $\langle \overline{s}(t_1), \ldots, \overline{s}(t_n)\rangle \in (r_k^n)^{\mathbf{A}}$.
2. Para cualesquiera fórmulas α y β de $\mathbf{L}$,
   a) $I^s_{\mathbf{A}}(\neg\alpha) = I^s_{\mathbf{A}}(\alpha) + 1$
   b) $I^s_{\mathbf{A}}(\alpha \to \beta) = I^s_{\mathbf{A}}(\alpha) I^s_{\mathbf{A}}(\beta) + I^s_{\mathbf{A}}(\alpha) + 1$
   c) $I^s_{\mathbf{A}}(\alpha \vee \beta) = I^s_{\mathbf{A}}(\alpha) I^s_{\mathbf{A}}(\beta) + I^s_{\mathbf{A}}(\alpha) + I^s_{\mathbf{A}}(\beta)$
   d) $I^s_{\mathbf{A}}(\alpha \wedge \beta) = I^s_{\mathbf{A}}(\alpha) I^s_{\mathbf{A}}(\beta)$
   e) $I^s_{\mathbf{A}}(\alpha \leftrightarrow \beta) = I^s_{\mathbf{A}}(\alpha) + I^s_{\mathbf{A}}(\beta) + 1$
3. Para fórmula α y símbolo de variable x del lenguaje:
   a) $I^s_{\mathbf{A}}(\forall x\alpha) = 1$ sii, por definición, para todo $d \in A$, $I^{s(x|d)}_{\mathbf{A}}(\alpha) = 1$

b) $I^s_{\mathbf{A}}(\exists x\alpha) = 1$ sii, por definición, existe $d \in A$ tal que $I^{s(x|d)}_{\mathbf{A}}(\alpha) = 1$

donde

$$s(x|d)(y) = \begin{cases} d, & \text{si } y = x \\ s(y), & \text{si } y \neq x \end{cases}$$

Por supuesto que las operaciones + y · que aparecen en esta definición son las mismas presentadas y usadas en la Definición **??**, es decir son respectivamente la suma y producto del cuerpo $\mathbb{Z}_2$.

Ejemplo 2.2. Sea x, y dos símbolos de variable distintos, α la fórmula $\forall x r(y,x)$ en el lenguaje $\mathbf{L}$ y $\mathbf{A}$ la estructura para $\mathbf{L}$ según lo siguiente:

- Como universo A de $\mathbf{A}$, los números naturales y
- como $(r)^{\mathbf{A}}$ la relación binaria $\leq$.

Sea u una asignación fija entre las que cumplen $u(y) = 0$. Se tiene que:

$$\begin{aligned}
\mathrm{I}^u_{\mathbf{A}}(\alpha) = 1 &\Leftrightarrow \text{ para todo } n \in \omega, \quad \mathrm{I}^{u(x|n)}_{\mathbf{A}}(r(y,x)) = 1 \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad \mathrm{I}^{u(x|n)}_{\mathbf{A}}(r(y,x)) = 1 \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad \langle u(x|n)(y), u(x|n)(x)\rangle \in (r)^{\mathbf{A}} \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad \langle u(y), u(x|n)(x)\rangle \in (r)^{\mathbf{A}} \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad \langle 0, n\rangle \in \leq \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad 0 \leq n
\end{aligned}$$

La conocida veracidad de la última afirmación conlleva la veracidad de cualquiera de las más arriba equivalentes por lo que, en particular, $\mathrm{I}^u_{\mathbf{A}}(\alpha) = 1$.

Ejemplo 2.3. Sea x, y dos símbolos de variable distintos y α la fórmula $\forall x \exists y r(x, f(y))$, al ser letras distintas) en el lenguaje $\mathbf{L}$ y sea $\mathbf{A}$ la estructura para $\mathbf{L}$ según lo siguiente:

- Como universo A de $\mathbf{A}$, los números naturales y
- como $(r)^{\mathbf{A}}$ la relación binaria $\leq$.

Sea u una asignación fija entre las que cumplen $u(y) = 1$. Se tiene que:

$$\begin{aligned}
\mathrm{I}^u_{\mathbf{A}}(\alpha) = 1 &\Leftrightarrow \text{ para todo } n \in \omega, \quad \mathrm{I}^{u(x|n)}_{\mathbf{A}}(r(y,x)) = 1 \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad \mathrm{I}^{u(x|n)}_{\mathbf{A}}(r(y,x)) = 1 \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad \langle u(x|n)(y), u(x|n)(x)\rangle \in (r)^{\mathbf{A}} \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad \langle u(y), u(x|n)(x)\rangle \in (r)^{\mathbf{A}} \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad \langle 0, n\rangle \in \leq \\
&\Leftrightarrow \text{ para todo } n \in \omega, \quad 1 \leq n
\end{aligned}$$

La última afirmación es falsa, pues **0 es un número natural** y es estrictamente menor que 1; por lo que su afirmación equivalente $\mathrm{I}^u_{\mathbf{A}}(\alpha) = 1$ es falsa también. En conclusión, $\mathrm{I}^u_{\mathbf{A}}(\alpha) = 0$.

Ejemplo 2.4. Sea x, y dos símbolos de variable distintos, α la fórmula $\forall x r(y,x)$ en el lenguaje $\mathbf{L}$ y $\mathbf{A}$ la estructura para $\mathbf{L}$ según lo siguiente:

- Como universo A de $\mathbf{A}$, los números naturales,
- como $(f)^{\mathbf{A}}$ la función siguiente s y
- como $(r)^{\mathbf{A}}$ la relación binaria $\leq$.

Sea u una asignación cualquiera, pero fija. Se tiene que:

$$\begin{aligned}
\mathrm{I}^u_{\mathbf{A}}(\alpha) = 1 &\Leftrightarrow \text{ para todo } n \in \omega, \quad \mathrm{I}^{u(x|n)}_{\mathbf{A}}(\exists y r(x, f(y))) = 1 \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad \mathrm{I}^{u(x|n,y|m)}_{\mathbf{A}}(r(x, f(y))) = 1 \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad \langle u(x|n, y|m)(x), u(x|n, y|m)(f(y))\rangle \in (r)^{\mathbf{A}} \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad \langle u(x|n, y|m)(x), (f)^{\mathbf{A}}(u(x|n, y|m)(y))\rangle \in (r)^{\mathbf{A}} \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad \langle n, s(m)\rangle \in \leq \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad n \leq s(m)
\end{aligned}$$

La última afirmación es cierta, pues dado n basta tomar como m el propio n cumpliéndose entonces $n < n + 1$, y en particular, $n \leq n + 1$. En conclusión, $\mathrm{I}^u_{\mathbf{A}}(\alpha) = 1$.

Ejemplo 2.5. Sea x, y dos símbolos de variable distintos, α la fórmula $\forall x r(y, x)$ en el lenguaje $\mathbf{L}$ y $\mathbf{A}$ la estructura para $\mathbf{L}$ según lo siguiente:

- Como universo A de $\mathbf{A}$, los números naturales,
- como $(f)^{\mathbf{A}}$ la función siguiente s y
- como $(r)^{\mathbf{A}}$ la relación binaria $>$.

Sea u una asignación cualquiera, pero fija. Se tiene que:

$$\begin{aligned}
\mathrm{I}^u_{\mathbf{A}}(\alpha) = 1 &\Leftrightarrow \text{ para todo } n \in \omega, \quad \mathrm{I}^{u(x|n)}_{\mathbf{A}}(\exists y r(x, f(y))) = 1 \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad \mathrm{I}^{u(x|n,y|m)}_{\mathbf{A}}(r(x, f(y))) = 1 \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad \langle u(x|n, y|m)(x), u(x|n, y|m)(f(y))\rangle \in (r)^{\mathbf{A}} \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad \langle u(x|n, y|m)(x), (f)^{\mathbf{A}}(u(x|n, y|m)(y))\rangle \in (r)^{\mathbf{A}} \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad \langle n, s(m)\rangle \in \leq \\
&\Leftrightarrow \text{ para todo } n \in \omega \text{ existe } m \in \omega \text{ tal que }, \quad n > s(m)
\end{aligned}$$

La última afirmación es falsa. En efecto, sea $n = 0$ y supongamos que existiese un número natural m_0 tal que $s(m_0) < 0$. Como 0 es menor que cualquier número natural no nulo, y en partircular que $s(m_0)$, se tendría $s(m_0) < 0 < s(m_0)$ y así, para el número natural m_0 ocurriría que $s(m_0) \neq s(m_0)$, lo cual es absurdo. Como resultado deducimos que $\mathrm{I}^u_{\mathbf{A}}(\alpha)$ debe valer 0, esto es,

$$\mathrm{I}^u_{\mathbf{A}}(\alpha) = 0$$

Definición 2.2. Sea $\mathbf{L}$ un lenguaje de primer orden y $\Gamma \cup \{\alpha\} \subseteq \mathrm{Form}(\mathbf{L})$. Γ *implica semánticamente* a φ, en símbolos $\Gamma \vDash \varphi$, si para toda interpretación $\langle \mathbf{A}, s\rangle$ de $\mathbf{L}$ se tiene $I^s_{\mathbf{A}}(\varphi) = 1$ siempre que para toda $\gamma \in \Gamma$ ocurra $I^s_{\mathbf{A}}(\gamma) = 1$. En el caso de que $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$, escribimos $\gamma_1, \ldots, \gamma_n \vDash \alpha$ en lugar de $\{\gamma_1, \ldots, \gamma_n\} \vDash \alpha$ y si $\Gamma = \varnothing$ escribimos simplemente $\vDash \alpha$ en lugar de $\varnothing \vDash \alpha$. Las fórmulas ψ y φ son *lógicamente equivalentes* si $\psi \vDash \varphi$ y $\varphi \vDash \psi$ (o dicho de otra forma, si $\vDash \psi \leftrightarrow \varphi$).

Definición 2.3. Sea $\mathbf{L}$ un lenguaje y φ una fórmula de $\mathbf{L}$. La fórmula φ es satisfacible si existe una interpretación $\langle \mathbf{A}, s\rangle$ de $\mathbf{L}$ tal que $I^s_{\mathbf{A}}(\varphi) = 1$. La fórmula φ es insatisfacible si no es satisfacible. La fórmula φ es *válida* si para toda interpretación $\langle \mathbf{A}, s\rangle$ de $\mathbf{L}$, $I^s_{\mathbf{A}}(\varphi) = 1$ (o sea, $\vDash \varphi$).

Teorema 2.1. *Sea* $\mathbf{L}$ *un lenguaje y* $\Gamma \cup \{\psi, \varphi\}$ *un conjunto de fórmulas fórmulas de* $\mathbf{L}$*. Son equivalentes las siguientes afirmaciones:*

1. $\Gamma, \psi \vDash \varphi$
2. $\Gamma \vDash \psi \to \varphi$

Corolario 2.2. *Sea* $\mathbf{L}$ *un lenguaje y* $\gamma_1, \ldots, \gamma_n, \varphi$ *fórmulas de* $\mathbf{L}$*. Son equivalentes las siguientes afirmaciones:*

1. $\gamma_1, \ldots, \gamma_n \vDash \varphi$

2. $\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n \vDash \varphi$

3. $\vDash \gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n \to \varphi$

Corolario 2.3. *Para cualesquiera fórmulas $\gamma_1, \ldots, \gamma_n, \varphi$ $(2 \leq n)$ son equivalentes las siguientes afirmaciones:*

1. $\gamma_1, \ldots, \gamma_n \vDash \varphi$

2. $\{\gamma_1, \gamma_2, \ldots, \gamma_n, \neg\varphi\}$ *es insatisfacible*

3. $\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n \wedge \neg\varphi$ *es insatisfacible*

Teorema 2.4 (Lema de Coincidencia)**.** *Sea* $\mathbf{L}$ *un lenguaje de primer orden,* φ *una fórmula de* $\mathbf{L}$ *y* W *el conjunto de las variables libres en* φ*. Sea* $\mathbf{A}$ *una estructura algebraica para* $\mathbf{L}$ *y* s_1, s_2 *asignaciones coincidentes en los elementos de* W *(en símbolos,* $s_1 \upharpoonright W = s_2 \upharpoonright W$*). Entonces*

$$I_{\mathbf{A}}^{s_1}(\varphi) = I_{\mathbf{A}}^{s_2}(\varphi)$$

Demostración. Procedemos a hacer la demostración por inducción. Sea pues $\mathbf{A}$ una estructura para $\mathbf{L}$ y φ una fórmula de $\mathbf{L}$. Supongamos que $\varphi \in \mathrm{Atom}(\mathbf{L})$ y que $\varphi = r_k^n(t_1 \cdots t_n)$. Cada ocurrencia en φ de cada variable es una ocurrencia libre. Por tanto, la hipótesis se traduce en que s_1 y s_2 coinciden en las variables de φ. Se deduce que $\overline{s}_1(t_i) = \overline{s}_2(t_i)$, $i = 1, \ldots, n$. Consecuentemente $\langle \overline{s}_1(t_1), \ldots, \overline{s}_1(t_n)\rangle \in (r_k^n)^{\mathbf{A}}$ si, y sólo si, $\langle \overline{s}_2(t_1), \ldots, \overline{s}_2(t_n)\rangle \in (r_k^n)^{\mathbf{A}}$. Así pues, $I_{\mathbf{A}}^{s_1}(\varphi) = I_{\mathbf{A}}^{s_2}(\varphi)$.

Supongamos que φ no es atómica y que el resultado es cierto para toda fórmula de "complejidad" menor que la de φ. Si φ tiene la forma $(\neg\alpha)$, $(\alpha \to \beta)$, $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$ o $(\alpha \leftrightarrow \beta)$ entonces el resultado es inmediato por el contenido de la Defininición 2.1 y la hipótesis de inducción. Si existen una variable x y una fórmula α de $\mathbf{L}$ tales que $\varphi = \forall x\alpha$ (resp. $\varphi = \exists x\alpha$), entonces las variables libres de φ son las de α, salvo eventualmente x, que podría ser libre en α pero no en φ. Así, para todo $a \in A$, $s_1(x|a)$ y $s_2(x|a)$ coinciden en las variables libres de α. Por lo anterior y la hipótesis de inducción tenemos que para todo $a \in A$, $I_{\mathbf{A}}^{s_1(x|a)}(\alpha) = I_{\mathbf{A}}^{s_2(x|a)}(\alpha)$. Es decir, $I_{\mathbf{A}}^{s_1}(\forall x\alpha) = I_{\mathbf{A}}^{s_2}(\forall x\alpha)$ (resp. $I_{\mathbf{A}}^{s_1}(\exists x\alpha) = I_{\mathbf{A}}^{s_2}(\exists x\alpha)$). $\square$

Corolario 2.5. *Sea* $\mathbf{L}$ *un lenguaje,* $\mathbf{A}$ *una estructura algebraica para* $\mathbf{L}$ *y* σ *una sentencia de* $\mathbf{L}$*, entonces ocurre una, y sólo una, de las siguientes alternativas:*

1. *Para toda asignación* s *de* V *en* $\mathbf{A}$*,* $I_{\mathbf{A}}^{s}(\sigma) = 1$.

2. *Para toda asignación* s *de* V *en* $\mathbf{A}$*,* $I_{\mathbf{A}}^{s}(\sigma) = 0$.

Observación 2.1. Observar que no pueden ocurrir simultáneamente las opciones 1) y 2) del corolario 2.5. La razón para ello es que A, por definición, es no vacío en cualquier estructura algebraica $\mathbf{A}$.

Definición 2.4. Sea $\mathbf{L}$ un lenguaje, $\mathbf{A}$ una estructura algebraica para $\mathbf{L}$ y φ una fórmula de $\mathbf{L}$. $\mathbf{A}$ es *modelo* de φ si para toda asignación s de V en A, $I_{\mathbf{A}}^{s}(\varphi) = 1$. $\mathbf{A}$ es un modelo de un conjunto de fórmulas Γ si es modelo de cada una de sus fórmulas.

Observación 2.2.

1. Toda estructura es modelo de $\varnothing$

2. Cuando φ es sentencia, $\mathbf{A}$ es modelo de φ si, y sólo si, existe una asignación s de V en A tal que $I_{\mathbf{A}}^{s}(\varphi) = 1$. (cfr. Teorema 2.5.)

Corolario 2.6. *Sea* $\mathbf{L}$ *un lenguaje y* $\Gamma \cup \{\tau\}$ *un conjunto de sentencias de* $\mathbf{L}$*.* $\Gamma \vDash \tau$ *si, y sólo si, todo modelo de* Γ *es un modelo de* τ*.*

Ejemplo 2.6. Supongamos que el lenguaje $\mathbf{L}$ consta de los símbolos r^2, f^1 y c. Consideramos la estructura $\mathbf{A} = \langle \omega, \leq, s, 0 \rangle$, donde $(r^2)^{\mathbf{A}} = \leq$, $(f^1)^{\mathbf{A}}$ es la función siguiente, $(c)^{\mathbf{A}} = 0$. Sea $s\colon \mathrm{V} \longrightarrow \omega$ definida como $s(v_i) = i - 1$. $\overline{s}(f(f(v_3))) = (2')' = 4$, $\overline{s}(f(f(c))) = 2$, $I^s_{\mathbf{A}}(r(c, f(v_1))) = 1$, porque $\langle \overline{s}(c), \overline{s}(f(v_1)) \rangle = \langle 0, 1 \rangle \in \leq$. También se cumple y es fácil verificar que $I^s_{\mathbf{A}}(\forall v_1 r(c, v_1)) = 1$. Sin embargo, $I^s_{\mathbf{A}}(\forall v_1 r(v_2, v_1)) = 0$, porque existe un natural m tal que $\nvDash_{\mathbf{A}} r(v_2, v_1)[s(v_1|m)]$, es decir, $\langle s(v_2), m \rangle \notin \leq$. En efecto, como $s(v_2) = 1$ podemos tomar como m el valor 0.

Ejercicio 2.1.

1. $\forall v_1 q(v_1) \vDash q(v_2)$
2. $q(v_1) \nvDash \forall v_1 q(v_1)$
3. $\forall v_1 q(v_1) \vDash \exists v_2 q(v_2)$
4. $\exists x \forall y p(x, y) \vDash \forall y \exists x p(x, y)$
5. $\forall y \exists x p(x, y) \nvDash \exists x \forall y p(x, y)$
6. $\vDash \exists x(q(x) \to \forall x q(x))$

3. Ejercicios sobre Lenguajes e Interpretaciones

1. Para las siguientes fórmulas concluir qué variables son libres y cuáles son ligadas, detallando el carácter de libertad de cada una de sus ocurrencias respectivas:

   a) $\forall z(r(x, z) \to s(y, z))$
   b) $\exists x r(x, y)$
   c) $\exists x r(y, x)$
   d) $\exists z r(y, x)$
   e) $\exists x(r(x, y) \wedge s(x, y))$
   f) $\exists x r(x, y) \wedge \forall y s(x, y)$
   g) $\exists x(r(x, y) \wedge \forall y s(x, y))$
   h) $\exists x(\forall y r(x, y) \wedge s(x, y))$
   i) $\exists z(r(x, z) \vee p(y))$
   j) $\forall x(p(x) \to r(x, y))$
   k) $\forall x((p(x) \wedge q(x)) \to \neg r(x, y))$
   l) $\exists x(p(x) \wedge s(x, y))$
   m) $\exists x(\exists y q(x) \vee r(x, y))$
   n) $\forall x(\exists y r(x, y) \to r(x, z))$
   ñ) $\forall x(r(x, z) \to \exists y r(x, y))$
   o) $\forall x(\forall z r(x, z) \to s(x, z))$
   p) $((p(x) \vee q(y)) \wedge \forall x \forall y r(x, y))$

2. Consideremos un lenguaje $\mathbf{L}$ con los siguientes símbolos:

   - símbolos de constante: c, d
   - símbolos de función: no hay
   - símbolos de relación: p^1, q^1, e^2, r^2, s^2

   Sea $\mathbf{A}$ la estructura algebraica para $\mathbf{L}$ definida por:

- $A = \mathbb{Z}_4$
- $(c)^{\mathbf{A}} = 0$, $(d)^{\mathbf{A}} = 1$
- $(p)^{\mathbf{A}} = \{x : x \in \mathbf{Z}_4, x^2 = 0\}$
- $(q)^{\mathbf{A}} = \{x : x \in \mathbf{Z}_4, x^2 = 2\}$
- $(e)^{\mathbf{A}} = \Delta(\mathbb{Z}_4)$
- $(r)^{\mathbf{A}} = \{(0,1),(0,2),(2,3),(2,2),(1,2),(3,0)\}$
- $(s)^{\mathbf{A}} = \{(0,1),(0,2),(0,3),(2,3),(0,0)\}$

Estudiar cuales de las siguientes sentencias son verdaderas para esta estructura:

1) $p(c)$
2) $\neg p(d)$
3) $p(c) \wedge p(d)$.
4) $p(c) \to \neg q(d)$
5) $\exists x q(x)$
6) $\neg(\exists x q(x))$
7) $\exists x \neg q(x)$
8) $\exists x(p(x) \wedge q(x))$
9) $\forall x q(x)$
10) $\forall x(p(x) \to q(x))$
11) $\forall x(q(x) \to \neg p(x))$
12) $\forall x(q(x) \to \exists y(p(x) \vee q(y)))$
13) $\forall x r(c,x)$
14) $\forall x s(c,x)$
15) $\forall x(r(c,x) \to s(c,x))$
16) $\exists y \forall x(r(c,x) \to s(c,x))$
17) $\forall x \forall y(r(x,y) \to s(x,y))$
18) $\forall x \forall y(r(x,y) \to \exists z(s(x,z)))$
19) $\forall x(p(x) \to \exists y(r(x,y)))$
20) $\forall x(p(x) \to \exists y(s(x,y) \wedge r(y,x)))$
21) $\forall x \exists y r(x,y)$
22) $\forall x \exists y s(x,y)$
23) $\exists y \forall x r(x,y)$
24) $\exists y \forall x s(x,y)$
25) $\exists y \forall x r(y,x)$
26) $\forall x \forall y \forall z((s(x,y) \wedge s(y,z)) \to r(x,z))$
27) $\forall x \forall y(r(x,y) \to \neg r(y,x))$
28) $\forall x \forall y(\neg s(x,y) \to \neg s(x,y))$
29) $\forall x \forall y(\exists z(r(x,z) \wedge r(z,y)) \to r(x,y))$
30) $\forall x \forall y(\exists z(r(x,z) \wedge s(z,y)) \to r(x,y))$
31) $\forall x \forall y(\exists z(s(x,z) \wedge r(z,y)) \to r(x,y))$
32) $\forall x \forall y(\exists z(r(x,z) \wedge r(z,y)) \to s(x,y))$

33) $\forall x(p(x) \to \exists y r(y,x))$

34) $\forall x(e(x,c) \to \exists y r(y,x))$

35) $\forall x(\exists y r(y,x) \to p(x))$

36) $\forall x(e(x,d) \leftrightarrow r(c,x))$

3. Calcular la interpretación de las siguientes fórmulas en cada una de las $\mathbf{L}$-estructuras que se dan:

   a) $\forall x \forall y e(f(x,y), f(y,x))$ en las $\mathbf{L}$-estructuras:

      1) $\mathbf{A}$ definida por:
         - $A = \{1,2,3,4,5,6,7\}$
         - $(f)^{\mathbf{A}}(x,y) = \begin{cases} x+y & \text{, si } x+y \in A, \\ x & \text{, en otro caso.} \end{cases}$
         - $(e)^{\mathbf{A}} = \{(1,1),(2,2),(3,3),(4,4),(5,5),(6,6),(7,7),(1,7),(7,1),(2,6),(6,2),(3,5),(5,3)\}$

      2) $\mathbf{B}$ definida por:
         - $B = \mathcal{M}_3(\mathbb{R})$
         - $(f)^{\mathbf{B}}(x,y) = xy$
         - $(e)^{\mathbf{B}} = \Delta(B)$

      3) $\mathbf{C}$ definida por:
         - $C = \mathbb{Z}$
         - $(f)^{\mathbf{C}}(x,y) = xy$
         - $(e)^{\mathbf{C}} = \Delta(C)$

   b) $\forall x e(f(x,a), f(a,x))$

      1) $\mathbf{A}$ definida por:
         - $A = \mathcal{M}_3(\mathbb{R})$
         - $(a)^{\mathbf{A}} = I_3$
         - $(f)^{\mathbf{A}}(x,y) = xy$
         - $(e)^{\mathbf{A}} = \Delta(A)$

      2) $\mathbf{B}$ definida por:
         - $B = \mathcal{M}_3(\mathbb{R})$
         - $(a)^{\mathbf{B}} = 0_3$
         - $(f)^{\mathbf{B}}(x,y) = xy$
         - $(e)^{\mathbf{B}} = \Delta(B)$

      3) $\mathbf{C}$ definida por:
         - $C = \mathcal{M}_3(\mathbb{R})$
         - $(a)^{\mathbf{C}} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$
         - $(f)^{\mathbf{C}}(x,y) = xy$
         - $(e)^{\mathbf{C}} = \Delta(C)$

   c) $\forall x \exists y e(f(x,y), a)$

      1) $\mathbf{A}$ definida por:
         - $A = \mathcal{M}_3(\mathbb{R})$
         - $(a)^{\mathbf{A}} = I_3$
         - $(f)^{\mathbf{A}}(x,y) = xy$
         - $(e)^{\mathbf{A}} = \Delta(A)$

2) **B** definida por:
    - $B = \mathcal{M}_3(\mathbb{R})$
    - $(a)^{\mathbf{B}} = 0_3$
    - $(f)^{\mathbf{B}}(x,y) = xy$
    - $(e)^{\mathbf{B}} = \Delta(B)$
3) **C** definida por:
    - $C = \mathbb{Z}$
    - $(a)^{\mathbf{C}} = 1$
    - $(f)^{\mathbf{C}}(x,y) = xy$
    - $(e)^{\mathbf{C}} = \Delta(C)$

d) $\forall x(e(f(x,x),a) \to e(x,a))$

1) **A** definida por:
    - $A = \mathcal{M}_3(\mathbb{R})$
    - $(a)^{\mathbf{A}} = I_3$
    - $(f)^{\mathbf{A}}(x,y) = xy$
    - $(e)^{\mathbf{A}} = \Delta(A)$
2) **B** definida por:
    - $B = \mathcal{M}_3(\mathbb{R})$
    - $(a)^{\mathbf{B}} = 0_3$
    - $(f)^{\mathbf{B}}(x,y) = xy$
    - $(e)^{\mathbf{B}} = \Delta(B)$
3) **C** definida por:
    - $C = \mathbb{Z}$
    - $(a)^{\mathbf{C}} = 1$
    - $(f)^{\mathbf{C}}(x,y) = xy$
    - $(e)^{\mathbf{C}} = \Delta(C)$
4) **D** definida por:
    - $D = \mathbb{R}$
    - $(a)^{\mathbf{D}} = 1$
    - $(f)^{\mathbf{D}}(x,y) = xy$
    - $(e)^{\mathbf{D}} = \Delta(D)$
5) **E** definida por:
    - $E = \mathbb{R}$
    - $(a)^{\mathbf{E}} = 0$
    - $(f)^{\mathbf{E}}(x,y) = xy$
    - $(e)^{\mathbf{E}} = \Delta(E)$
6) **F** definida por:
    - $F = \mathbb{Z}_4$
    - $(a)^{\mathbf{F}} = 0$
    - $(f)^{\mathbf{F}}(x,y) = xy$
    - $(e)^{\mathbf{F}} = \Delta(F)$

4. Determinar el carácter (satisfacible y refutable, universalmente válida o contradicción) de las siguientes fórmulas de primer orden:

a) $\forall x \forall y(p(x,y) \to p(y,x))$

b) $(\forall x \exists y(p(x,y) \to p(y,x))) \to (\exists x \exists y(p(x,y) \to p(y,x)))$

c) $\forall x \forall y (p(x,y) \to \neg p(y,x))$

d) $\forall x \exists y (p(x,y) \to \neg p(y,x))$

5. Consideramos las siguientes sentencias:

    1) $\forall x (p(x) \to \exists y r(x,y))$
    2) $\forall x (q(x) \to \exists y r(y,x))$
    3) $\exists x (p(x) \wedge q(x))$
    4) $\forall x \exists y r(x,y)$
    5) $\exists x \exists y \neg r(x,y)$

    Encontrar, si existen, estructuras en las que:

    a) Las sentencias 1), 3) y 5) sean verdaderas y las restantes falsas.

    b) Las sentencias 1), 2) y 4) sean verdaderas y las restantes falsas.

    c) Las sentencias 2) y 3) sean verdaderas y las restantes falsas.

    d) Las sentencias 2), 3) y 4) sean verdaderas y las restantes falsas.

    e) Las sentencias 2), 4) y 5) sean verdaderas y las restantes falsas.

    f) Las sentencias 2), 3) y 5) sean verdaderas y las restantes falsas.

    g) Todas las sentencias sean verdaderas.

    h) Todas las sentencias sean falsas.

6. Dadas las siguientes sentencias:

    1) $\forall x (\exists y r(x,y) \to p(x))$
    2) $\forall x (p(x) \to q(x))$
    3) $\exists x (q(x) \wedge \forall y \neg r(x,y))$
    4) $\exists x r(x,x)$
    5) $\exists x \neg r(x,x)$

    Encontrar, si existen, estructuras en las que:

    a) Las sentencias 1), 3) y 5) sean verdaderas y las restantes falsas.

    b) Las sentencias 1), 2) y 4) sean verdaderas y las restantes falsas.

    c) Las sentencias 1), 2) y 4) sean falsas y las restantes verdaderas.

    d) Las sentencias 2) y 3) sean verdaderas y las restantes falsas.

    e) Las sentencias 2), 3) y 4) sean verdaderas y las restantes falsas.

    f) Las sentencias 2), 4) y 5) sean verdaderas y las restantes falsas.

    g) Las sentencias 2), 3) y 5) sean verdaderas y las restantes falsas.

    h) Todas las sentencias sean verdaderas.

    i) Todas las sentencias sean falsas.

7. Estudiar para cada una de las siguientes fórmulas, si es universalmente válida, satisfacible, refutable o contradicción:

    a) $p(x) \to (\forall x \neg p(x) \to \neg p(f(a)))$

    b) $\forall x \exists y r(x,y) \to \exists y \forall x r(x,y)$

    c) $\exists x \forall y r(x,y) \to \forall y \exists x r(x,y)$

    d) $\forall x \forall y (r(x,y) \to r(y,x))$

e) $\forall x(p(x) \to p(a))$

f) $\forall x p(x) \to p(a)$

8. Dadas las siguientes fórmulas:

   - $\varphi = \forall x \exists y p(x, y)$
   - $\psi = \exists x \forall y p(x, y)$

   Encuentra una interpretación en la que ambas sean ciertas y otra en la que φ sea cierta y ψ sea falsa. ¿Es cierto $\varphi \vDash \psi$?

9. Consideramos el lenguaje de primer orden $\mathbf{L}$ dado por lo siguiente:

   - símbolos de constante: a
   - símbolos de función: f, g
   - símbolos de relación: P, Q

   y la $\mathbf{L}$-estructura $\mathbf{A}$ dada por:

   - $A = \mathbb{Z}_{10}$
   - $(a)^{\mathbf{A}} = 4$
   - $(f)^{\mathbf{A}} = +$ (suma en $\mathbb{Z}_{10}$), $(g)^{\mathbf{A}}(x) = x^2$
   - $(p)^{\mathbf{A}} = \{(k, k) \colon k \in A\} = \Delta(A)$, $(q)^{\mathbf{A}} = \{(0,0), (1,9), (2,8), (3,7), (4,6), (5,5), (6,4), (7,3), (8,2), (9,1)\}$

   Interpretar las fórmulas siguientes usando una asignación cualquiera

   $$s \colon Var(\mathbf{L}) \longrightarrow \mathbb{Z}_{10}$$

   a condición de que cumpla $s(x) = 0$ y $s(y) = 4$.

   a) $\forall x(p(g(x), a) \to q(f(y, a), x))$

   b) $\forall x \exists y \exists z p(x, f(g(y), g(z)))$

10. Consideremos el lenguaje de primer orden $\mathbf{L}$ dado por lo siguiente:

    - símbolos de constante: a, b, c
    - símbolos de función: f, g
    - símbolos de relación: P

    y la $\mathbf{L}$-estructura $\mathbf{A}$ dada por:

    - $A = \mathbb{Z}_6$
    - $(a)^{\mathbf{A}} = 0$, $(b)^{\mathbf{A}} = 1$, $(c)^{\mathbf{A}} = 2$
    - $(f)^{\mathbf{A}} = +$ (suma en $\mathbb{Z}_6$), $(g)^{\mathbf{A}}(x) = \cdot$ (producto en $\mathbb{Z}_6$)
    - $(q)^{\mathbf{A}} = \{(0,0), (1,1), (2,2), (3,3), (4,4), (5,5)\}$

    a) Describir todas las asignaciones s de $\mathbf{L}$ en $\mathbf{A}$ para las que la siguiente fórmula se interpreta como verdadera:

    $$\neg p(g(x, f(b, b)), a) \to p(f(y, c), g(y, y))$$

    b) Interpretar la sentencia $\forall x \exists y p(g(y, c), x)$.

11. Describir todas las estructuras en las que la fórmula siguiente es válida:

    $$r(x) \to \forall x r(x)$$

12. Consideramos el lenguaje de primer orden $\mathbf{L}$ dado por lo siguiente:

    - símbolos de constante: a
    - símbolos de función: f, g
    - símbolos de relación: P

    y la $\mathbf{L}$-estructura $\mathbf{A}$ dada por:

    - $A = \mathbb{Z}_6$
    - $(a)^{\mathbf{A}} = 2$
    - $(f)^{\mathbf{A}} = +$ (suma en $\mathbb{Z}_6$), $(g)^{\mathbf{A}}(x) = \cdot$ (producto en $\mathbb{Z}_6$),
    - $(p)^{\mathbf{A}} = \{(0,0),(1,1),(2,2),(3,3),(4,4),(5,5)\}$

    Consideremos una asignación s en $\mathbf{A}$ tal que $s(x_1) = 2, s(x_2) = 0, s(x_3) = 0, s(x_4) = 3$. Se pide interpretar las siguientes fórmulas:

    a) $\neg\exists x_1 \forall x_2 p(f(x_1, x_4), a)$
    b) $\forall x_1(p(x_1, g(x_1, x_1)) \leftrightarrow p(g(a, x_1), f(a, a)))$

13. Sea $(X, \leq)$ un conjunto ordenado. Consideramos un lenguaje de primer orden $\mathbf{L}$ con un símbolo de predicado binario r.Sea ahora la $\mathbf{L}$-estructura $\mathbf{A}$, cuyo universo es X, y para la que $(r)^{\mathbf{A}} =\leq$. Se pide escribir una fórmula que signifique exactamente que $(X, \leq)$ es un retículo.

14. Dada la fórmula
$$r(x) \leftrightarrow \exists x r(x)$$
    se pide:

    a) Probar que **no** es universalmente válida.
    b) Encontrar una estructura donde la fórmula no sea válida.
    c) ¿Es satisfacible la fórmula en cualquier estructura?
    d) ¿Es refutable en toda estructura?

15. Interpretar la fórmula
$$\forall x \forall y(p(x, y) \rightarrow \neg p(y, x))$$
    en las estructuras que seguidamente se sugieren:

    a) $A = \{0, 1, 2, 3\}$
    $p^{\mathbf{A}} = \{(0,0),(0,1),(1,2),(2,1),(2,2),(3,3)\}$
    b) $B = \mathbb{R}$
    $p^{\mathbf{B}}$ es la relación binaria "x es estrictamente menor que y"
    c) $C = \mathbb{N}$
    $p^{\mathbf{C}}$ es la relación binaria "x es múltiplo de y"

16. Interpretar la fórmula
$$\forall x \forall y(p(x, y) \rightarrow \neg p(y, x))$$
    en la estructura $\mathbf{A}$ con

    - $A = \{0, 1, 2, 3\}$
    - $(p)^{\mathbf{A}} = \{(0,0),(0,1),(1,2),(2,1),(2,2),(3,3)\}$

17. La hipótesis del *Lema de Coincidencia* es suficiente, pero no necesaria. Para demostrarlo, dar una fórmula φ en un lenguaje $\mathbf{L}$, una $\mathbf{L}$-estructura $\mathbf{A}$ y asignaciones s_1 y s_2 de V en $\mathbf{A}$ tales que si W_φ es el conjunto de símbolos de variable que ocurren libremente en φ, sea $s_1 \restriction W_\varphi \neq s_2 \restriction W_\varphi$ y sin embargo $I_{\mathrm{A}}^{s_1}(\varphi) = I_{\mathrm{A}}^{s_2}(\varphi)$.

4. Forma Normal Prenexa

En esta sección se trata de estudiar como proceder cuando la sustitución de una variable por un término no es posible y, sin embargo, persiste la necesidad de llevarla a cabo.

Lema 4.1. *Sea* $\mathbf{L}$ *un lenguaje y* $\alpha, \alpha' \in \mathrm{Form}(\mathbf{L})$. *Si* $\models \alpha \leftrightarrow \alpha'$ *entonces* $\models \neg\alpha \leftrightarrow \neg\alpha'$.

Lema 4.2. *Sea* $\mathbf{L}$ *un lenguaje y* $\alpha, \beta, \alpha', \beta' \in \mathrm{Form}(\mathbf{L})$. *Entonces:*

1. *Si* $\models \alpha' \to \alpha$ *y* $\models \beta \to \beta'$, *entonces* $\models (\alpha \to \beta) \to (\alpha' \to \beta')$
2. *Si* $\models \alpha \leftrightarrow \alpha'$ *y* $\models \beta \leftrightarrow \beta'$, *entonces* $\models (\alpha \to \beta) \leftrightarrow (\alpha' \to \beta')$

Definición 4.1. Sea $\mathbf{L}$ un lenguaje, x un símbolo de variable, t y u términos y φ una fórmula. Definimos u^x_t como sigue:

$$u^x_t = \begin{cases} a, & \text{si } u = a \\ t, & \text{si } u = x \\ y, & \text{si } u = y \text{ e } y \neq x \\ f((t_1)^x_t, \ldots, (t_n)^x_t), & \text{si } u = f(t_1, \ldots, t_n) \end{cases}$$

Definimos φ^x_t como sigue:

1. $(r(t_1, \ldots, t_n))^x_t = r((t_1)^x_t, \ldots, (t_n)^x_t)$.
2. $(\neg\alpha)^x_t$ es $\neg(\alpha^x_t)$
3. $(\alpha \to \beta)^x_t$ es $(\alpha^x_t \to \beta^x_t)$
4. $(\alpha \vee \beta)^x_t$ es $(\alpha^x_t \vee \beta^x_t)$
5. $(\alpha \wedge \beta)^x_t$ es $(\alpha^x_t \wedge \beta^x_t)$
6. $(\alpha \leftrightarrow \beta)^x_t$ es $(\alpha^x_t \leftrightarrow \beta^x_t)$
7. $(\forall y\alpha)^x_t = \begin{cases} \forall y\alpha, & \text{si } x = y, \\ \forall y(\alpha^x_t), & \text{si } x \neq y. \end{cases}$
8. $(\exists y\alpha)^x_t = \begin{cases} \exists y\alpha, & \text{si } x = y, \\ \exists y(\alpha^x_t), & \text{si } x \neq y. \end{cases}$

Ejemplo 4.1. Sea $\mathbf{L}$ un lenguaje, $p, q \in \mathrm{R}$ y $x, y, z \in \mathrm{V}$.

1. Para toda $\varphi \in \mathrm{Form}(\mathbf{L})$, $\varphi^x_x = \varphi$.
2. $(q(x) \to \forall xp(x))^x_y = (q(y) \to \forall xp(x))$
3. Si α es la fórmula $\neg\forall yx \approx y$, entonces $\forall x\alpha \to \alpha^x_z$ es
$$\forall x\neg\forall y(x \approx y) \to \neg\forall y(z \approx y)$$
4. Si α es la fórmula $\neg\forall y(x \approx y)$, entonces $\forall x\alpha \to \alpha^x_y$ es
$$\forall x\neg\forall y(x \approx y) \to \neg\forall y(y \approx y)$$

Lema 4.3. *Sean* $\mathbf{L}$ *un lenguaje,* φ *una fórmula,* x *e* y *variables distintas y* $\langle \mathbf{A}, s\rangle$ *una interpretación del lenguaje. Si* y *no ocurre en* φ, *entonces para todo* $a \in A$, $I^{s(x|a)}_{\mathbf{A}}(\varphi) = I^{s(y|a)}_{\mathbf{A}}(\varphi^x_y)$.

Teorema 4.4. *Sean* $\mathbf{L}$ *un lenguaje,* φ *una fórmula y variables* x *e* y. *Si* y *no ocurre en* $\forall x\varphi$, *entonces la fórmula* $\forall x\varphi$ *es lógicamente equivalente a* $\forall y(\varphi^x_y)$.

Lema 4.5. *Sea* **L** *un lenguaje,* x *una variable y* φ *una fórmula. Si* x *no ocurre libremente en* φ *entonces* φ*,* $\forall x\varphi$ *y* $\exists x\varphi$ *son fórmulas lógicamente equivalentes dos a dos.*

Lema 4.6. *Sea* **L** *un lenguaje,* $\alpha \in \mathrm{Form}(\mathbf{L})$ *y* $x \in \mathrm{V}$*. Entonces:*

1. $\models \neg\forall x\alpha \leftrightarrow \exists x\neg\alpha$
2. $\models \neg\exists x\alpha \leftrightarrow \forall x\neg\alpha$

Teorema 4.7. *Sea* **L** *un lenguaje,* $\alpha, \beta \in \mathrm{Form}(\mathbf{L})$ *y* $x \in \mathrm{V}$*. Si* x *no ocurre libremente en* α *entonces*

1. $\models \alpha \leftrightarrow \forall x\alpha$
2. $\models \alpha \leftrightarrow \exists x\alpha$
3. $\models (\alpha \to \exists x\beta) \leftrightarrow \exists x(\alpha \to \beta)$
4. $\models (\forall x\beta \to \alpha) \leftrightarrow \exists x(\beta \to \alpha)$
5. $\models (\alpha \to \forall x\beta) \leftrightarrow \forall x(\alpha \to \beta)$*.*
6. $\models (\exists x\beta \to \alpha) \leftrightarrow \forall x(\beta \to \alpha)$*.*
7. $\models (\forall x\beta \vee \alpha) \leftrightarrow \forall x(\beta \vee \alpha)$
8. $\models (\alpha \vee \forall x\beta) \leftrightarrow \forall x(\alpha \vee \beta)$
9. $\models (\alpha \vee \exists x\beta) \leftrightarrow \exists x(\alpha \vee \beta)$
10. $\models (\exists x\beta \vee \alpha) \leftrightarrow \exists x(\beta \vee \alpha)$
11. $\models (\forall x\beta \wedge \alpha) \leftrightarrow \forall x(\beta \wedge \alpha)$
12. $\models (\alpha \wedge \forall x\beta) \leftrightarrow \forall x(\alpha \wedge \beta)$
13. $\models (\alpha \wedge \exists x\beta) \leftrightarrow \exists x(\alpha \wedge \beta)$
14. $\models (\exists x\beta \wedge \alpha) \leftrightarrow \exists x(\beta \wedge \alpha)$

Definición 4.2. Sea **L** un lenguaje y φ una fórmula de **L**. φ está en *forma prenexa* si se expresa como

$$Q_1 x_1 \cdots Q_n x_n \beta \tag{1}$$

donde $Q_i \in \{\exists, \forall\}$, para todo $1 \leq i \leq n$ y en la escritura de β no aparece ningún cuantificador. Llamamos *literal* a cualquier fórmula que sea atómica o de la forma $\neg\alpha$, donde α es una fórmula atómica. En la expresión 1, β se denomina matriz de la fórmula φ y a la expresión $Q_1 x_1 \cdots Q_n x_n$ se le llama *prefijo* de φ. Una fórmula en forma prenexa está en *forma normal prenexa* (resp. *forma normal prenexa perfecta*) si su matriz está expresada en forma normal conjuntiva (resp. en forma normal conjuntiva perfecta). Una fórmula en forma normal prenexa tal que todos los cuantificadores de su prefijo son $\forall$ (resp. $\exists$) se denomina $\forall$-fórmula (resp. $\exists$-fórmula). Si la fórmula es una sentencia, hablaremos, matizando más, de $\forall$-sentencia y $\exists$-sentencia.

Observación 4.1. Cuando en la definición 4.2 hablamos de forma normal conjuntiva, perfecta o no, nos estamos refiriendo al lenguaje proposicional cuyas proposiciones atómicas son precisamente las fórmulas atómicas del lenguaje de primer orden **L**.

Teorema 4.8. *Sea* **L** *un lenguaje. Para todo* $\alpha \in \mathrm{Form}(\mathbf{L})$ *existe* $\alpha' \in \mathrm{Form}(\mathbf{L})$ *tal que:*

1. α' *está en forma prenexa*
2. $\models \alpha \leftrightarrow \alpha'$

Observación 4.2. Observar que la fórmula en forma prenexa equivalente cuya existencia garantiza el teorema 4.8 puede ser encontrada en forma normal prenexa.

Teorema 4.9. *Sea* $\mathbf{L}$ *un lenguaje,* $\alpha, \beta \in \mathrm{Form}(\mathbf{L})$ *y* $x \in \mathrm{V}$*. Entonces:*

1. $\vDash (\forall x\alpha \wedge \forall x\beta) \leftrightarrow \forall x(\alpha \wedge \beta)$.
2. $\vDash (\exists x\alpha \vee \exists x\beta) \leftrightarrow \exists x(\alpha \vee \beta)$.

Para encontrar una fórmula en forma prenexa que sea lógicamente equivalente a otra dada podemos seguir los siguitentes pasos:

- Usar que

$$\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi) \tag{2}$$
$$\varphi \to \psi \equiv \neg\varphi \vee \psi \tag{3}$$

- Usar repetidamente que:

$$\neg\neg\varphi \equiv \varphi \tag{4}$$

  las *leyes de De Morgan* que establecen que:

$$\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi \tag{5}$$
$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi \tag{6}$$

  así como que:

$$\neg\forall x\varphi \equiv \exists x\neg\varphi \tag{7}$$
$$\neg\exists x\varphi \equiv \forall x\neg\varphi \tag{8}$$

  de forma que los símbolos de negación precederán inmediatamente a las fórmulas atómicas.

- Renombrar variables cuantificadas, si fuera necesario, usando que si y no ocurre en $\forall x\varphi$ entonces:

$$\forall x\varphi \equiv \forall y(\varphi_y^x) \tag{9}$$
$$\exists x\varphi \equiv \exists y(\varphi_y^x) \tag{10}$$

- Usar que:

$$\forall x\varphi \wedge \forall x\psi \equiv \forall x(\varphi \wedge \psi) \tag{11}$$
$$\exists x\varphi \vee \exists x\psi \equiv \exists x(\varphi \vee \psi), \tag{12}$$

  y que si x no ocurre libremente en ψ

$$\forall x\psi \equiv \psi \tag{13}$$
$$\exists x\psi \equiv \psi \tag{14}$$
$$\forall x\varphi \vee \psi \equiv \forall x(\varphi \vee \psi) \tag{15}$$
$$\forall x\varphi \wedge \psi \equiv \forall x(\varphi \wedge \psi) \tag{16}$$
$$\exists x\varphi \vee \psi \equiv \exists x(\varphi \vee \psi) \tag{17}$$
$$\exists x\varphi \wedge \psi \equiv \exists x(\varphi \wedge \psi) \tag{18}$$

Ejemplo 4.2. Encuentre una fórmula en forma prenexa que sea lógicamente equivalente a la siguiente fórmula, que llamamos φ:

$$\exists x(q(x) \wedge \forall x p(a, x)) \to \forall x(q(x) \wedge \exists y \forall z r(a, y, z))$$

Solución.

$$
\begin{aligned}
\varphi &= \neg\exists x(q(x) \wedge \forall x p(a,x)) \vee \forall x(q(x) \wedge \exists y \forall z r(a,y,z)) \\
&= \forall x \neg(q(x) \wedge \forall x p(a,x)) \vee \forall x(q(x) \wedge \exists y \forall z r(a,y,z)) \\
&= \forall x(\neg q(x) \vee \neg\forall x p(a,x)) \vee \forall x(q(x) \wedge \exists y \forall z r(a,y,z)) \\
&= (\forall x \neg q(x) \vee \exists x \neg p(a,x)) \vee \forall x(q(x) \wedge \exists y \forall z r(a,y,z)) \\
&= (\forall x \neg q(x) \vee \exists y \neg p(a,y)) \vee \forall x(q(x) \wedge \exists y \forall z r(a,y,z)) \\
&= \exists y(\forall x \neg q(x) \vee \neg p(a,y)) \vee \exists y(\forall z q(z) \wedge \forall z r(a,y,z)) \\
&= \exists y(\forall x(\neg q(x) \vee \neg p(a,y)) \vee \forall z(q(z) \wedge r(a,y,z))) \\
&= \exists y \forall x \forall z(\neg q(x) \vee \neg p(a,y) \vee (q(z) \wedge r(a,y,z))) \\
&= \exists y \forall x \forall z((\neg q(x) \vee \neg p(a,y) \vee q(z)) \wedge (\neg q(x) \vee \neg p(a,y) \vee r(a,y,z)))
\end{aligned}
$$

□

Ejemplo 4.3. Encuentre una fórmula en forma prenexa que sea lógicamente equivalente a la siguiente fórmula, que llamamos φ:

$$\forall x(\forall y \exists z(p(y,z) \to q(x)) \to \exists z(\neg p(x,f(x)) \vee q(z)))$$

Solución.

$$
\begin{aligned}
\varphi &= \forall x(\neg\forall y \exists z(\neg p(y,z) \vee q(x)) \vee \exists z(\neg p(x,f(x)) \vee q(z))) \\
&= \forall x(\exists y \forall z(p(y,z) \wedge \neg q(x)) \vee \exists z(\neg p(x,f(x)) \vee q(z))) \\
&= \forall x(\exists y \forall z(p(y,z) \wedge \neg q(x)) \vee \exists y(\neg p(x,f(x)) \vee q(y))) \\
&= \forall x \exists y(\forall z(p(y,z) \wedge \neg q(x)) \vee (\neg p(x,f(x)) \vee q(y))) \\
&= \forall x \exists y \forall z((p(y,z) \wedge \neg q(x)) \vee (\neg p(x,f(x)) \vee q(y))) \\
&= \forall x \exists y \forall z((p(y,z) \vee \neg p(x,f(x)) \vee q(y)) \wedge (\neg q(x) \vee \neg p(x,f(x)) \vee q(y)))
\end{aligned}
$$

□

Ejemplo 4.4. Encuentre una fórmula en forma prenexa que sea lógicamente equivalente a la siguiente fórmula, que llamamos φ:

$$(\forall x \exists y \neg p(x,y) \to q(z)) \to \forall x \exists z r(x,z)$$

Solución.

$$
\begin{aligned}
\varphi &= \neg(\neg\forall x \exists y \neg p(x,y) \vee q(z)) \vee \forall x \exists z r(x,z) \\
&= (\forall x \exists y \neg p(x,y) \wedge \neg q(z)) \vee \forall x \exists z r(x,z) \\
&= \forall x \exists y(\neg p(x,y) \wedge \neg q(z)) \vee \forall x \exists z r(x,z) \\
&= \forall x(\exists y(\neg p(x,y) \wedge \neg q(z)) \vee \forall x \exists z r(x,z)) \\
&= \forall x(\exists y(\neg p(x,y) \wedge \neg q(z)) \vee \forall w \exists y r(w,y)) \\
&= \forall x \forall w(\exists y(\neg p(x,y) \wedge \neg q(z)) \vee \exists y r(w,y)) \\
&= \forall x \forall w \exists y((\neg p(x,y) \wedge \neg q(z)) \vee r(w,y)) \\
&= \forall x \forall w \exists y((\neg p(x,y) \vee r(w,y)) \wedge (\neg q(z) \vee r(w,y)))
\end{aligned}
$$

□

5. Forma Normal de Skolem

Definición 5.1. Sea φ una fórmula del lenguaje de primer orden $\mathbf{L}$ en forma normal prenexa, esto es, se escribe como:

$$Q_1 x_1 \cdots Q_n x_n \beta$$

estando β en forma normal conjuntiva. Se denomina *forma de Skolem* de φ, representada φ^s, a la fórmula obtenida suprimiento los cuantificadores existenciales $\exists x_i$ que hubiere en el prefijo de φ y reemplazando en β cada una de las ocurrencias de x_i cuantificada con $\exists$ por $f_i(x_{j_1}, \ldots, x_{j_{k_i}})$, donde $x_{j_1}, \ldots, x_{j_{k_i}}$ son los símbolos de variable cuantificados con $\forall$ y situados antes de $\exists x_i$ en el prefijo de φ. Los símbolos de función f_i deben ser distintos de todos los que aparezcan en la fórmula φ y reciben el nombre de *funciones de Skolem*.

Observación 5.1. Si en la operación descrita en la Defininición 5.1 el cuantificador $\exists x_i$ no está precedido de ningún cuantificador $\forall x_j$, el término $f_i(x_{j_1}, \ldots, x_{j_{k_i}})$ que se introduce es sencillamente un símbolo de constante c_i; pues siempre se ha entendido que un símbolo de constante es un símbolo de función 0-ario.

Ejemplo 5.1.

- si φ fuese $\exists x p(x, f(x))$, φ^s sería $p(a, f(a))$
- si φ fuese $\forall x \exists y p(x, f(y))$, φ^s sería $\forall x p(x, f(f_1(x)))$
- si φ fuese $\exists x_1 \forall x_2 \exists x_3 \exists x_4 (\neg p(x_1, x_2) \vee r(x_3, x_4))$, φ^s sería $\forall x_2 (\neg p(a, x_2) \vee r(f_3(x_2), f_4(x_2)))$
- si φ fuese $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 (p(x_1, x_2, x_3, x_4, x_5)$, φ^s sería $\forall x_2 \forall x_4 (p(a, x_2, f_3(x_2), x_4, f_5(x_2, x_4))$

Observación 5.2. Sea Γ un conjunto de sentencias de $\mathbf{L}$. Para cada φ de Γ, sea φ^p un fórmula en forma normal prenexa lógicamente equivalente a φ y para cada φ^p, sea φ^s una forma de Skolem de φ^p. Representaremos por Γ^p al conjunto de las sentencias φ^p y por Γ^s al conjunto de las sentencias φ^s.

Teorema 5.1. *Sea Γ un conjunto de sentencias de un lenguaje de primer orden $\mathbf{L}$. Son equivalentes las siguientes afirmaciones:*

1. *Γ es satisfacible*
2. *Γ^s es satisfacible*

Demostración. La demostración está en la pág. 48 de [3] y mucho mejor en [2], pág. 249. □

El problema general de la *demostración automática de teoremas* es el de saber si $\Gamma \vDash \varphi$, donde $\Gamma \cup \{\varphi\}$ es un conjunto de sentencias del lenguaje de primer orden $\mathbf{L}$; y en los casos prácticos Γ es finito. En virtud del Teorema **??**, sabemos que el problema se reduce a saber si $\Gamma \cup \{\neg\varphi\}$ es insatisfacible o no. Lo que aporta el Teorema 5.1 es que equivalentemente basta con estudiar la insatisfacibilidad de $\Gamma^s \cup \{(\neg\varphi)^s\}$

Definición 5.2. Sea $\mathbf{L}$ un lenguaje de primer orden. Llamamos *cláusula* a cualquier fórmula φ del lenguaje satisfaciendo las siguientes condiciones:

1. φ está en forma normal prenexa
2. en su matriz no aparece el símbolo $\wedge$
3. es una $\forall$-sentencia

Convenimos que la expresión vacía es una cláusula —la *cláusula vacía*, que a tal efecto representamos por el símbolo $\square$, y que es una cláusula insatisfacible. La cláusula que en su matriz no tiene más que un literal (fórmula atómica o negación de ella) se denomina *cláusula unitaria* o también *cláusula unit*. Un *literal* es cualquier fórmula atómica de $\mathbf{L}$ o cualquier negación de una fórmula atómica. El literal que es fórmula atómica se denomina *positivo* , un literal es *negativo* si no es positivo. Dado un literal λ, definimos su *literal complementario* λ^c como sigue:

$$\lambda^c = \begin{cases} \neg r(t_1, \ldots, t_n) & , \textit{ si } \lambda = r(t_1, \ldots, t_n) \\ r(t_1, \ldots, t_n) & , \textit{ si } \lambda = \neg r(t_1, \ldots, t_n) \end{cases}$$

Ejemplo 5.2.

1. $\forall x(\neg p(x) \vee r(f(x)))$ es una cláusula
2. $\forall x((\neg p(x) \vee r(f(x))) \wedge (\neg p(x) \vee q(x, f(x))))$ no es una cláusula.
3. $\forall x_2 \forall x_4 (p(a, x_2, f_3(x_2), x_4, f_5(x_2, x_4))$ es una cláusula.

Observación 5.3. En la práctica las cláusulas vienen representadas por su matriz. Así, eventualmente (y frecuentemente) diremos "la cláusula $\neg p(x) \vee r(f(x))$" en lugar de "la cláusula $\forall x(\neg p(x) \vee r(f(x)))$".

Observación 5.4. Supongamos que φ es una $\forall$-sentencia de la forma $\forall x_1 \forall x_2 \cdots \forall x_n \beta$. Si β es la fórmula $\alpha \wedge \gamma$, es claro que φ es lógicamente equivalente a la siguiente conjunción:

$$(\forall x_1 \forall x_2 \cdots \forall x_n \alpha) \wedge (\forall x_1 \forall x_2 \cdots \forall x_n \gamma)$$

Así pues, toda $\forall$-sentencia es lógicamente equivalente a una conjunción de cláusulas; es por ello que al estudiar la sastisfacibilidad de conjuntos de fórmulas podemos sustituir cualquier $\forall$-sentencia φ que aparezca en él por el conjunto de las cláusulas que conjuntadas dan una fórmula lógicamente equivalente a φ. Es decir, $\Gamma \cup \{\varphi\}$ es insatisfacible si, y sólo si, lo es $\Gamma \cup \{\varphi_1, \ldots, \varphi_n\}$, donde las φ_i son cláusulas y la $\forall$-sentencia φ es lógicamente equivalente a $\varphi^{sc} = \bigwedge_{i=1}^{n} \varphi_i$. Así pues, si nos preguntamos por la satisfacibilidad de un conjunto de sentencias Γ, el problema es fácilmente trasladable al estudio de la satisfacibilidad de un conjunto de cláusulas Γ^{sc}; para ello pasaremos de Γ a Γ^{s} y posteriormente a Γ^{sc} cambiando cada φ^{s} de Γ^{s} por los conjuntos de la fórmula φ^{sc}.

Ejemplo 5.3. Nos preguntamos si es cierto o no que $\forall x(p(x) \to \exists y(r(y) \wedge q(x, y))), \exists x p(x) \vDash \exists x \exists y q(x, y)$. Para intentar dar una respuesta consideramos las sentencias:

- en cuanto a la fórmula $\forall x(p(x) \to \exists y(r(y) \wedge q(x, y)))$:
  - $\forall x(p(x) \to \exists y(r(y) \wedge q(x, y)))$
  - $\forall x(\neg p(x) \vee \exists y(r(y) \wedge q(x, y)))$
  - $\forall x \exists y(\neg p(x) \vee (r(y) \wedge q(x, y)))$
  - $\forall x \exists y((\neg p(x) \vee r(y)) \wedge (\neg p(x) \vee q(x, y)))$
- en cuanto a la fórmula $\exists x p(x)$ no es preciso transformarla.
- en cuanto a la fórmula $\neg \exists x \exists y q(x, y)$ se transforma en $\forall x \forall y \neg q(x, y)$.

Llegamos, pues, a preguntarnos por la satisfacibilidad del conjunto

$$\{\forall x \exists y((\neg p(x) \vee r(y)) \wedge (\neg p(x) \vee q(x, y))), \exists x p(x), \forall x \forall y \neg q(x, y)\}. \tag{19}$$

Por el Teorema 5.1, la insatisfacibilidad de 19 es la de

$$\{\forall x\big((\neg p(x) \vee r(f(x))) \wedge (\neg p(x) \vee q(x, f(x)))\big), p(a), \forall x \forall y \neg q(x, y)\} \tag{20}$$

o más esquemáticamente la de

$$\{\neg p(x) \vee r(f(x)), \neg p(x) \vee q(x, f(x)), p(a), \neg q(x, y)\} \tag{21}$$

Si el conjunto 20, entendido como sugiere 21, no tiene modelos, ello significa que $\forall x(p(x) \to \exists y(r(y) \wedge q(x, y))), \exists x p(x) \vDash \exists x \exists y q(x, y)$. Es preciso que el conjunto 21 sea lo más simple posible, para lo cual usaremos de forma óptima todas las herramientas de que disponemos.

6. Teorema de Herbrand

En esta sección y en las siguientes trataremos de métodos de demostración. Realmente el encontrar un procedimiento general de decisión para verificar la inconsistencia de una fórmula es un deseo muy antiguo. Lo intentó primero Leibniz (1646–1716), lo revivió después Peano a principios del siglo 20 y posteriormente la escuela de Hilbert en la década de 1920; pero no fue hasta 1936 cuando demostraron Church y Turing que ello es imposible. Church y Turing independientemente demostraron que no hay un método general de decisión para decidir la validez de fórmulas en la lógica de primer orden. No obstante, existen métodos de demostración que pueden verificar que una fórmula es válida si ciertamente es válida. Para fórmulas no válidas, esos métodos en general no acaban nunca. A la vista del resultado de Church y Turing, esto es lo mejor que podemos esperar obtener de un método de demostración.

A priori, para saber si un conjunto de sentencias del cálculo de predicados admite un modelo es necesario realizar una infinidad de intentos: intentar encontrar un modelo sobre un universo de un elemento, intentar encontrar un modelo sobre un universo de dos elementos,..., intentar encontrar un modelo sobre un universo infinito. Cada uno de estos intentos se divide a su vez en un gran número de intentos. El interés del *Teorema de Herbrand* que veremos es que nos permite reducir nuestros intentos a uno: para saber si un conjunto de sentencias Γ tiene modelos, basta con saber si tiene un “modelo sintáctico”, es decir construído de forma standard a partir del vocabulario utilizado en las fórmulas de Γ. Además, saber si este modelo sintáctico existe se reduce al estudio de conjuntos de fórmulas del cálculo proposicional.

Para ilustrar las definiciones que vienen, retomamos el Ejemplo 5.3 que nos proporcionaba el conjunto Σ de cláusulas:

- $\neg p(x) \vee r(f(x))$
- $\neg p(x) \vee q(x, f(x))$
- $p(a)$
- $\neg q(x, y)$

Definición 6.1 (Universo de Herbrand)**.** Dado un conjunto de cláusulas Σ en un lenguaje de primer orden $\mathbf{L}$, sea:

1. H_0 el conjunto de los símbolos de constante que aparecen en los elementos de Σ. Si no apareciera ninguno, entonces consideramos como H_0 el simplete de un símbolo de constante nuevo (del lenguaje o no), esto es, $H_0 = \{c\}$.
2. Dado $0 < i$, definimos H_i como la unión de H_{i-1} y el conjunto de todos los términos de $\mathbf{L}$ de la forma $f^n(t_1, \ldots, t_n)$, donde f^n es cualquier símbolo de función n-ario que ocurra en Σ y $t_1, \ldots, t_n$ son cualesquiera elementos de H_{i-1}.

El *universo de Herbrand* de Σ, representado U_Σ, es por definición la unión de todos los H_i (es decir, $\bigcup_{i\in\omega} H_i$). Cada H_i, con i natural, se denomima *conjunto escalón i-ésimo de constantes de* Σ.

Observación 6.1. El universo de Herbrand de cualquier conjunto de cláusulas Σ es siempre no vacío; es finito si, y sólo si, en Σ no aparecen símbolos de función.

Ejemplo 6.1. En nuestro ejemplo tenemos que $H_0 = \{a\}$, $H_1 = \{a, f(a)\}$, $H_2 = \{a, f(a), f(f(a))\}$, etc. De esta forma:

$$\mathrm{U}_\Sigma = \{a, f(a), f(f(a)), \ldots, f(f(\cdots a)\cdots)), \ldots\}$$

Definición 6.2 (Base de Herbrand)**.** Sea Σ un conjunto de cláusulas en un lenguaje de primer orden $\mathbf{L}$. La *base de Herbrand* de Σ, representada como B_Σ, es el conjunto de fórmulas atómicas $r^n(t_1, \ldots, t_n)$, donde r^n es cualquier símbolo de predicado n-ario que ocurra en Σ y $t_1, \ldots, t_n$ son cualesquiera términos de U_Σ.

Ejemplo 6.2. En nuestro ejemplo:

$$\begin{aligned}\mathrm{B}_\Sigma = \{&p(a), r(a), q(a,a), p(f(a)), r(f(a)), q(a, f(a)), q(f(a), a),\\ &q(f(a), f(a)), p(f(f(a))), \ldots\}\end{aligned}$$

Definición 6.3. Sea Σ un conjunto de cláusulas en un lenguaje de primer orden $\mathbf{L}$ y sea σ una cláusula de Σ. Una *instancia básica* de σ es cualquier cláusula obtenida reemplazando cada símbolo de variable de σ por un elemento de U_Σ.[1]

Definición 6.4. Sea Σ un conjunto de cláusulas en un lenguaje de primer orden $\mathbf{L}$. El *sistema de Herbrand* asociado a Σ, representado por S_Σ es el conjunto de las instancias básicas de las cláusulas de Σ.

Ejemplo 6.3. En nuestro ejemplo

$$\begin{aligned}\mathrm{S}_\Sigma = \{&\neg p(a) \vee q(a, f(a)), \neg p(a) \vee r(f(a)), p(a)\\ &\neg q(a,a), \neg p(f(a)) \vee q(f(a), f(f(a))), \neg p(f(a)) \vee r(f(f(a))),\\ &\neg q(a, f(a)), \neg q(f(a), a), \neg q(f(a), f(a)), \ldots\}\end{aligned}$$

Teorema 6.1 (de Herbrand)**.** *Sea Σ un conjunto de cláusulas de un lenguaje de primer orden $\mathbf{L}$. Son equivalentes las siguientes afirmaciones:*

1. *Σ es satisfacible*
2. *Σ tiene un modelo cuyo universo es U_Σ*
3. *Todo subconjunto finito de S_Σ es satisfacible*

Demostración. Tomamos la demostración del teorema 19.11 que hay en la página 254 del [2]. □

Observación 6.2. El Teorema 6.1 se usa frecuentemente en negativo. Para demostrar que un conjunto de cláusulas Σ es insatisfacible basta con exhibir un subconjunto finito de S_Σ que sea insatisfacible.

Ejemplo 6.4. Siguiendo con nuestro ejemplo, el subconjunto $\{p(a), \neg p(a) \vee q(a, f(a)), \neg q(a, f(a))\}$ de S_Σ —que es un subconjunto de cláusulas proposicionales— es insatisfacible. En consecuencia, la pregunta formulada en el Ejemplo 5.3 tiene respuesta afirmativa, es decir,

$$\forall x(p(x) \to \exists y(r(y) \wedge q(x,y))), \exists x p(x) \vDash \exists x \exists y q(x,y)$$

Observación 6.3. Para implementar el *teorema de Herbrand* utilizamos el *método de Davis y Putnam.* Dicho algoritmo se desenvuelve en el ámbito del lenguaje proposicional construido tomando como variables proposicionales los elementos de la *base de Herbrand* B_Σ del problema en cuestión. Primero se generan metódicamente todas las fórmulas de B_Σ y después se genera metódicamente el conjunto S_Σ del problema: si S_Σ es finito (debido a que U_Σ y B_Σ lo son) todo acaba estudiando su satisfacibilidad con el método de Davis y Putnam; pero si no lo es, estudiamos —de nuevo con el método de Davis y Putnam— la satisfacibilidad de sus r primeras fórmulas y en función de que lo sean o no repetimos todo con las $r+1$ primeras o nos detenemos.

Éste es de hecho el primero de los algoritmos practicables sobre demostración automática basados en el teorema de Herbrand. Resulta útil en problemas sencillos. Con la base del teorema de Herbrand le precedieron el método basado en las tablas de verdad y el de Gilmore en 1960, que utilizaba la transformación en forma normal disyuntiva con simplificación.

Cuando se trabaja a mano con base en el teorema de Herbrand algunos utilizan el método de los *árboles semánticos* que dejamos por ahora y que se puede consultar en [4], [3] y [7].

Ejemplo 6.5. Nos preguntamos si $\exists x \neg q(x) \to \forall y p(y)$ es consecuencia de las fórmulas $\exists x p(x) \to \forall y p(y)$ y $\forall x(p(x) \vee q(x))$, es decir, nos preguntamos si es cierto:

$$\exists x p(x) \to \forall y p(y), \forall x(p(x) \vee q(x)) \vDash \exists x \neg q(x) \to \forall y p(y)$$

[1]Se entiende que por cada variable de σ seleccionamos un término de U_Σ y procedemos a sustituir cada ocurrencia en σ de la variable por ese término asociado a ella. Efectuamos esta operación con cada símbolo de variable y su término asociado.

Todo se reduce a estudiar la insatisfacibilidad del conjunto

$$\{\exists x p(x) \to \forall y p(y), \forall x(p(x) \vee q(x)), \neg(\exists x \neg q(x) \to \forall y p(y))\}$$

Al transformar ese conjunto en cláusulas (pasando a forma prenexa y luego a forma de Skolem) obtenemos las siguientes:

$$\neg p(x) \vee p(y)$$
$$p(x) \vee q(x)$$
$$\neg p(a)$$
$$\neg p(b)$$

La puesta en forma de Skolem ha introducido dos símbolos de constante: a y b, por ello y por no encontrar en las cláusulas signo de función alguno el universo de Herbrand es el conjunto finito $\{a, b\}$. Así el conjunto de sus instancias básicas tiene ocho elementos (¿cuáles?), pero basta con considerar el siguiente subconjunto suyo:

$$\{p(a) \vee q(a), \neg q(a), \neg p(a) \vee p(b), \neg p(b)\}$$

que por el *método de Davis y Putnam* sabemos que es insatisfacible. Por el Teorema 5.1, el Corolario 2.2 y el Teorema de Herbrand deducimos que la pregunta inicial tiene respuesta afirmativa.

Ejemplo 6.6. Sea el conjunto de cláusulas $\Sigma = \{\neg p(x) \vee q(f(x), x), p(g(b)), \neg q(y, z)\}$. Uno de los conjuntos insatisfacibles de instancias básicas de cláusulas de Σ es

$$\Sigma' = \{\neg p(g(b)) \vee q(f(g(b)), g(b)), p(g(b)), \neg q(f(g(b)), g(b))\}$$

Ejemplo 6.7. Sea el conjunto de cláusulas

$$\begin{aligned}\Sigma = \{&\neg p(x, y, u) \vee \neg p(y, z, v) \vee \neg p(x, v, w) \vee p(u, z, w),\\ &\neg p(x, y, u) \vee \neg p(y, z, v) \vee \neg p(u, z, w) \vee p(x, v, w),\\ &p(g(x, y), x, y), p(x, h(x, y), y),\\ &p(x, y, f(x, y)), \neg p(k(x), x, k(x))\}.\end{aligned}$$

Uno de los conjuntos insatisfacibles de instancias básicas de cláusulas de Σ es

$$\begin{aligned}\Sigma' = \{&p\big(a, h(a, a), a\big),\\ &p\big(g(a, k(h(a, a))), a, k(h(a, a))\big),\\ &\neg p\big(k(h(a, a)), h(a, a), k(h(a, a))\big),\\ &\neg p\big(g(a, k(h(a, a))), a, k(h(a, a))\big) \vee \neg p\big(a, h(a, a), a\big)\\ &\vee \neg p\big(g(a, k(h(a, a))), a, k(h(a, a))\big) \vee p\big(k(h(a, a)), h(a, a), k(h(a, a))\big)\}.\end{aligned}$$

7. Ejercicios sobre Forma Prenexa

1. Sea $\mathbf{L}$ un lenguaje, $\alpha, \beta \in \mathrm{Form}(\mathbf{L})$ y $x \in \mathrm{V}$. Demostrar que si x no ocurre libremente en α entonces

   a) $\vDash \alpha \leftrightarrow \forall x \alpha$

   b) $\vDash \alpha \leftrightarrow \exists x \alpha$

   c) $\vDash (\alpha \to \exists x \beta) \leftrightarrow \exists x(\alpha \to \beta)$

   d) $\vDash (\forall x \beta \to \alpha) \leftrightarrow \exists x(\beta \to \alpha)$

   e) $\vDash (\alpha \to \forall x \beta) \leftrightarrow \forall x(\alpha \to \beta)$.

   f) $\vDash (\exists x \beta \to \alpha) \leftrightarrow \forall x(\beta \to \alpha)$.

g) $\vDash (\forall x\beta \vee \alpha) \leftrightarrow \forall x(\beta \vee \alpha)$

h) $\vDash (\alpha \vee \forall x\beta) \leftrightarrow \forall x(\alpha \vee \beta)$

i) $\vDash (\alpha \vee \exists x\beta) \leftrightarrow \exists x(\alpha \vee \beta)$

j) $\vDash (\exists x\beta \vee \alpha) \leftrightarrow \exists x(\beta \vee \alpha)$

k) $\vDash (\forall x\beta \wedge \alpha) \leftrightarrow \forall x(\beta \wedge \alpha)$

l) $\vDash (\alpha \wedge \forall x\beta) \leftrightarrow \forall x(\alpha \wedge \beta)$

m) $\vDash (\alpha \wedge \exists x\beta) \leftrightarrow \exists x(\alpha \wedge \beta)$

n) $\vDash (\exists x\beta \wedge \alpha) \leftrightarrow \exists x(\beta \wedge \alpha)$

y demostrar que:

a) $\forall x\alpha \wedge \forall x\beta \equiv \forall x(\alpha \wedge \beta)$

b) $\exists x\alpha \vee \exists x\beta \equiv \exists x(\alpha \vee \beta)$

c) $\neg\forall x\alpha \equiv \exists x\neg\alpha$

d) $\neg\exists x\alpha \equiv \forall x\neg\alpha$

Demostrar también que es imprescindible la condición "x no ocurre libremente en α" en la primera parte del ejercicio.

2. Sea **L** un lenguaje, α una fórmula de **L** y x un símbolo de variable. Demostrar que si y es un símbolo de variable tal que $y \neq x$ e $y \notin lib(\alpha)$ entonces:

   a) $\forall x\alpha \equiv \forall y\alpha[x/y]$

   b) $\exists x\alpha \equiv \exists y\alpha[x/y]$

3. Demostrar que:

   a) $\vDash (\forall x\alpha \vee \forall x\beta) \to \forall x(\alpha \vee \beta)$

   b) $\vDash \exists x(\alpha \wedge \beta) \to (\exists x\alpha \wedge \exists x\beta)$

   pero que en general **no** son ciertas las afirmaciones:

   a) $\vDash \forall x(\alpha \vee \beta) \to (\forall x\alpha \vee \forall x\beta)$

   b) $\vDash (\exists x\alpha \wedge \exists x\beta) \to \exists x(\alpha \wedge \beta)$

4. Encontrar una fórmula en forma prenexa cuya matriz esté expresada como conjunción de disyunciones de literales y que sea lógicamente equivalente a las siguientes:

   a) $\big(\forall x\exists yp(x,y) \wedge \big(\exists yq(y) \to q(a)\big)\big) \vee \forall y\big(\exists y\forall xp(x,y) \vee \exists zp(y,a)\big)$

   b) $\big(\forall x(r(x) \vee \exists y\forall xp(x,y)) \vee \exists xq(x,y)\big) \wedge \big(\exists zr(z) \to \forall x(r(x) \wedge \forall xp(x,a))\big)$

   c) $\forall xp(x,y) \to \big(\forall yp(y,x) \to \forall x(q(x) \wedge \exists y\forall zr(a,y,z))\big)$

   d) $(\forall xp(a,x) \vee \forall xp(x,a)) \to \big(\forall zp(x,z) \wedge \forall w\forall y(p(a,y) \to \exists zq(z))\big)$

   e) $\forall x\forall z\big((\forall zp(x,z) \wedge \forall xp(x,z)) \to \forall x(\exists yp(x,y) \vee \forall xq(x))\big)$

   f) $\big(\forall w(\forall xr(x,w) \to (\forall xp(x) \to \exists x(q(x) \vee p(w))))\big) \wedge \forall z(p(z) \vee \exists zq(z))$

   g) $\forall x\big(r(x) \wedge \neg\exists x(p(x) \to \exists yq(f(y),x))\big) \wedge \forall w\exists z\big(q(z,a) \vee p(w) \vee (\forall yp(f(y)) \to q(x,z))\big)$

   h) $\big(\forall yp(x,y) \to \exists xr(x)\big) \wedge \neg\exists x\big((\forall yr(y)) \wedge \neg p(x,a)\big) \wedge \forall x\big((\exists yp(x,y)) \vee r(x)\big)$

5. Repetir el ejercicio 4 dando un resultado con el número mínimo de cuantificadores, y ellos óptimamente situados en el preámbulo de la nueva fórmula.

6. Para cada fórmula obtenida en el ejercicio 4, encontrar una fórmula de Skolem asociada.

7. Dada una fórmula en forma prenexa y una fórmula de Skolem asociada a ella. ¿Están ambas expresadas en el mismo lenguaje de primer orden? ¿Qué relación existe entre ambas?

8. Dado el conjunto de cláusulas:

   - $\neg p(x) \vee r(f(x))$
   - $\neg p(x) \vee q(x, f(x))$
   - $p(a)$
   - $\neg q(x, y)$

   expresar el universo de Herbrand, la base de Herbrand y el sistema de Herbrand.

9. Sea el conjunto de cláusulas

$$\begin{aligned}\Sigma = \{&\neg p(x,y,u) \vee \neg p(y,z,v) \vee \neg p(x,v,w) \vee p(u,z,w),\\ &\neg p(x,y,u) \vee \neg p(y,z,v) \vee \neg p(u,z,w) \vee p(x,v,w),\\ &p(g(x,y),x,y), p(x,h(x,y),y),\\ &p(x,y,f(x,y)), \neg p(k(x),x,k(x))\}.\end{aligned}$$

   Usar el *Teorema de Herbrand*, combinado con el algoritmo de Davis-Putnam, para demostrar que es insatisfacible.

10. Demostrar vía el *Teorema de Herbrand* y el Algoritmo de Davis-Putnam que:

$$\forall x(p(x) \to \exists y(r(y) \wedge q(x,y))), \exists x p(x) \vDash \exists x \exists y q(x,y)$$

8. Algoritmo de Unificación

El método descrito en el capítulo anterior para determinar si una fórmula es consecuencia de un conjunto finito de hipótesis es inutilizable una vez que hemos de manejar una decena de fórmulas. Ello es debido a que se trata de encontrar una contradicción con átomos de Herbrand obtenidos reemplazando variables por términos de forma absolutamente no sistemática.

Esta idea de hacer coincidir átomos los unos con los otros para encontrar más rápidamente contradicciones, es la idea que sustenta la resolución con variables y que estudiaremos después.

El hacer conincidir átomos por medio de una buena elección de términos que sustituyen variables es *la unificación*. Se trata de una idea muy vieja en lógica matemática puesto que aparece en la tesis de Herbrand, en 1930. No obstante su utilización como elemento fundamental de un "algoritmo" de demostración automática de teoremas, y como herramienta privilegiada de la programación lógica es debida a J.A. Robinson en el año 1965.

Definición 8.1. Sea $\mathbf{L}$ un lenguaje de primer orden. Una *sustitución elemental* es cualquier expresión de forma $(x|t)$, donde x es un símbolo de variable y t es término de $\mathbf{L}$. Si φ es una fórmula de $\mathbf{L}$, $\varphi(x|t)$ representa la fórmula obtenida reemplazando todas las ocurrencias libres de x en φ por t, es decir, $\varphi(x|t)$ representa ahora la fórmula φ_t^x. Una sustitución elemental $(x|t)$ es una *sustitución elemental de renombramiento* si t es un símbolo de variable.

Ejemplo 8.1. Sea la sustición elemental $(x|f(y,g(a)))$ y la fórmula $(p(u) \to r(x))$. Entonces $(p(u) \to r(x))(x|f(y,g(a)))$ es la fórmula $(p(u) \to r(f(y,g(a))))$.

Definición 8.2. Sea $\mathbf{L}$ un lenguaje de primer orden. Una *sustitución* Φ es cualquier composición $c_1 \circ \cdots \circ c_n$ de sustituciones elementales en cantidad finita $c_1, \ldots, c_n$. La sucesión finita $c_1, \ldots, c_n$ se denomina *descomposición* de Φ y ello nos lleva a representar a Φ simplemente como $[c_1 \cdots c_n]$. La sustitución identidad será representada como ϵ o simplemente $[\ ]$. Una sustitución $[c_1 \cdots c_n]$ es una *sustitución de renombramiento* si para todo $1 \leq i \leq n$, c_i es una sustitución elemental de renombramiento.

Ejemplo 8.2. Tomemos $\Phi = [(x|f(a))(y|f(x))]$ como sustitución y $p(x,y)$ como fórmula.

$$\begin{aligned}p(x,y)\Phi &= (p(x,y)(x|f(a)))(y|f(x))\\ &= p(f(a),y)\\ &= p(f(a),f(f(x)))\end{aligned}$$

Observación 8.1.

1. La aplicación sobre fórmulas $[c_1c_2]$ es distinta de la aplicación $[c_2c_1]$, siendo $c_1 = (x|f(a))$ y $c_2 = (y|f(x))$. Para mostrarlo sea, por ejemplo:

$$\begin{aligned}p(x,y)[(x|f(a))(y|f(x))] &= p(f(a),y)(y|f(x))\\ &= p(f(a),f(x))\\ &\neq p(f(a),f(f(a)))\\ &= p(x,f(x))(x|f(a))\\ &= (p(x,y)(y|f(x)))(x|f(a))\\ &= p(x,y)[(y|f(x))(x|f(a))]\end{aligned}$$

2. La descomposición de una sustitución en sustituciones elementales no es en general única:
   - $[(x|y)(z|y)] = [(z|y)(x|y)]$
   - $[(x|t)(y|t)(z|t)] = [(y|x)(z|x)(x|t)]$

Definición 8.3. Sean $\varphi_1, \varphi_2, \ldots, \varphi_m$ fórmulas atómicas de un lenguaje de primer orden $\mathbf{L}$. Una sustitución Φ es un *unificador* de $\varphi_1, \ldots, \varphi_m$ si, y sólo si, por definición, se cumple:

$$\varphi_1\Phi = \varphi_2\Phi = \cdots = \varphi_m\Phi$$

Ejemplo 8.3. Sea $\varphi_1 = p(x,z)$, $\varphi_2 = p(f(y),g(a))$ y $\varphi_3 = p(f(u),z)$. La sustitución $\Phi_1 = [(x|f(u))(y|u)(z|g(a))]$ es un unificador porque:

$$\varphi_1\Phi_1 = \varphi_2\Phi_1 = \varphi_3\Phi_1 = p(f(u),g(a))$$

La sustitución $\Phi_2 = [(u|f(a))]\Phi_1$ es también un unificador de φ_1, φ_2 y φ_3 porque:

$$\varphi_1\Phi_2 = \varphi_2\Phi_2 = \varphi_3\Phi_2 = p(f(f(a)),g(a))$$

Observación 8.2.

1. Es claro que si Φ es un unificador de un conjunto finito de fórmulas atómicas, para cualquier sustitución Ψ se cumple que $\Psi\Phi$ es un unificador de dicho conjunto de fórmulas.

2. Para ciertos conjuntos finitos de fórmulas atómicas no existe unificador. Por ejemplo, para $\varphi_1 = p(x,y)$ y $\varphi_2 = r(f(t),y)$, lo cual resulta evidente dado que los símbolos de predicado son distintos; pero aún evitando esta eventualidad es posible encontrar ejemplos de fórmulas atómicas sin unificador. El siguiente es uno: $\varphi_1 = p(x,f(x))$ y $\varphi_2 = p(f(y),y)$.

Dado un conjunto finito de fórmulas atómicas es posible determinar si son unificables y caso de serlo encontrar un unificador. Seguidamente damos una explicación algorítmica de como llevar a cabo dicha tarea.

Algoritmo de Unificación de *2* fórmulas atómicas

Input: fórmulas atómicas φ_1 y φ_2.
Output: Respuesta a la pregunta de si son unificables φ_1 y φ_2 y un unificador en caso afirmativo.

- $\Phi \leftarrow \epsilon$.-
- mientras que $\varphi_1\Phi \neq \varphi_2\Phi$, hacer

    - determinar el símbolo más a la izquierda de $\varphi_1\Phi$ que es diferente de su homólogo en $\varphi_2\Phi$.-
    - determinar los subtérminos de $\varphi_1\Phi$ y $\varphi_2\Phi$, t_1 y t_2 respectivamente, que comienzan en los símbolos determinados en el paso anterior.-
    - si "ninguno de los términos es una variable" o "uno es una variable que está contenida en el otro"
        - Imprimir "φ_1 y φ_2 no son unificables"; detenerse.-

      en otro caso
        - determinar x un símbolo de variable entre t_1 y t_2.-
        - determinar t entre t_1 y t_2 que no es x.-
        - hacer $\Phi \leftarrow \Phi(x|t)$.-
    - fin–de–si.-
- fin–mientras–que.-
- imprimir "Φ es un unificador de φ_1 y φ_2".-
- concluir.-

Ejemplo 8.4.

$\varphi_1\Phi$	$\varphi_2\Phi$	Φ
$\underset{\uparrow}{p}(x,f(x),a)$	$\underset{\uparrow}{p}(u,w,w)$	$\Phi \leftarrow \epsilon$
$p(\underset{\uparrow}{x},f(x),a)$	$p(\underset{\uparrow}{u},w,w)$	$\Phi \leftarrow [(x|u)]$
$p(u,\underset{\uparrow}{f}(u),a)$	$p(u,\underset{\uparrow}{w},w)$	$\Phi \leftarrow [(x|u)(w|f(u))]$
$p(u,f(u),\underset{\uparrow}{a})$	$p(u,f(u),\underset{\uparrow}{f}(u))$	**fallo**

El fallo es debido a que ni a ni $f(u)$ es un signo de variable; $p(x,f(x),a)$ y $p(u,w,w)$ no son unificables.

Ejemplo 8.5.

$\varphi_1\Phi$	$\varphi_2\Phi$	Φ
$\underset{\uparrow}{p}(x,f(g(x)),a)$	$\underset{\uparrow}{p}(b,y,z)$	$\Phi \leftarrow \epsilon$
$p(\underset{\uparrow}{x},f(g(x)),a)$	$p(\underset{\uparrow}{b},y,z)$	$\Phi \leftarrow [(x|b)]$
$p(b,\underset{\uparrow}{f}(g(b)),a)$	$p(b,\underset{\uparrow}{y},z)$	$\Phi \leftarrow [(x|b)(y|f(g(b)))]$
$p(b,f(g(b)),\underset{\uparrow}{a})$	$p(b,f(g(b)),\underset{\uparrow}{z})$	$\Phi \leftarrow [(x|b)(y|f(g(b)))(z|a)]$

Las fórmulas $p(x,f(g(x)),a)$ y $p(b,y,z)$ son unificables y $[(x|b)(y|f(g(b)))(z|a)]$ es un unificador.

Ejemplo 8.6.

$\varphi_1\Phi$	$\varphi_2\Phi$	Φ
$\underset{\uparrow}{p}(x,f(x))$	$\underset{\uparrow}{p}(f(y),y)$	$\Phi \leftarrow \epsilon$
$p(\underset{\uparrow}{x},f(x))$	$p(\underset{\uparrow}{f}(y),y)$	$\Phi \leftarrow [(x|f(y))]$
$p(f(y),\underset{\uparrow}{f}(f(y)))$	$p(f(y),\underset{\uparrow}{y})$	**fallo**

El fallo es debido a que y es un símbolo de variable presente en el término $t = f(f(y))$; así $p(x,f(x))$ y $p(f(y),y)$ no son unificables.

Algoritmo de Unificación de m fórmulas atómicas

Input: fórmulas atómicas $\varphi_1,\ldots,\varphi_m$ $(2 \leq m)$.
Output: Respuesta a la pregunta de si son unificables $\varphi_1,\ldots,\varphi_m$ y un unificador Φ en caso de respuesta afirmativa.-

- Para i desde 1 hasta $m-1$ hacer
  - si $\varphi_i\Phi_1\cdots\Phi_{i-2}\Phi_{i-1}$ y $\varphi_{i+1}\Phi_1\cdots\Phi_{i-2}\Phi_{i-1}$ son unificables
    1. determinar un unificador Φ_i.-
    2. en otro caso imprimir "$\varphi_1,\dots,\varphi_m$ no son unificables"; concluir.-
  - fin–de–si.-
- fin–de–para.-
- $\Phi \leftarrow \Phi_1\cdots\Phi_{m-1}$.-
- imprimir "Φ es un unificador de $\varphi_1,\dots,\varphi_m$".-
- concluir.-

Ejemplo 8.7. Sean $m=3$, $\varphi_1 = p(x,y)$, $\varphi_2 = p(f(z),x)$ y $\varphi_3 = p(w,f(x))$:

- $\Phi_1 = (x|f(z))(y|f(z))$ es un unificador de $\varphi_1 = p(x,y)$, $\varphi_2 = p(f(z),x)$
- $\Phi_1 p(f(z),x) = p(f(z),f(z))$
  $\Phi_1 p(w,f(x)) = p(w,f(f(z)))$
- fracaso porque $p(f(z),f(z))$ y $p(w,f(f(z)))$ no son unificables.

Observación 8.3. Es preciso prestar atención cuando se unifica por este método, para no olvidarse de aplicar $\Phi_1\cdots\Phi_{i-2}\Phi_{i-1}$ a las fórmulas atómicas no unificadas aún antes de aplicarles el algoritmo de unificación. En el ejemplo precedente, si no hubieramos aplicado Φ_1 a $p(w,f(x))$, la segunda etapa de unificación hubiera sido posible (porque $p(f(z),f(z))$ y $p(w,f(x))$ se unifican en $p(f(z),f(z)))$ y hubiésemos concluido, falazmente, que las tres fórmulas son unificables.

Definición 8.4. Un unificador Φ de las fórmulas atómicas $\varphi_1,\dots,\varphi_m$ es un *unificador de máxima generalidad* o *unificador principal* para ellas si, y sólo si, por definición, para todo unificador Φ' de las misma existe una sustición Ψ tal que $\Phi' = \Phi\Psi$.

Ejemplo 8.8. La sustitución $\Phi_1 = [(x|f(u))(y|u)(z|g(a))]$ es un unificador de máxima generalidad de $\varphi_1 = p(x,z)$, $\varphi_2 = p(f(y),g(a))$ y $\varphi_3 = p(f(u),z)$. No es el único,

$$\Phi_3 = [(u|v)(x|f(v))(y|v)(z|g(a))]$$

es otro y se tiene

$$\Phi_1 = \Phi_3(v|u) \qquad\qquad \Phi_3 = \Phi_1(u|v)$$

Observación 8.4. Intuitivamente, un unifcador (cuando existe al menos uno) de máxima generalidad es una sustitución tan general (es decir, ordenando tan pocas operaciones) como sea posible a condición de que unifique las fórmulas.

Teorema 8.1. *Sea* **L** *un lenguaje de primer orden y $\varphi_1,\dots,\varphi_m$ fórmulas atómicas. Si $\varphi_1,\dots,\varphi_m$ son fórmulas unificables entonces el algoritmo de unificación se detiene aportando Φ, un unificador de máxima generalidad de $\varphi_1,\dots,\varphi_m$.*

9. Resolución

La resolución de Robinson (1965) evita la generación de conjuntos de instancias básicas como exigen los métodos basados en el teorema de Herbrand. Puede ser aplicada directamente al cualquier conjunto Σ de cláusulas para sondear la insatisfacibilidad de Σ. La idea esencial del principio de resolución es indagar si Σ contiene la cláusula vacía $\square$ o no: si Σ contiene $\square$, entonces es insatisfacible. Si Σ no contiene $\square$, la siguiente cosa es saber cuando $\square$ puede ser derivada de Σ. Ciertamente el principio de resolución puede ser entendido como una regla de inferencia para generar nuevas cláusulas a partir de las de Σ y las ya generadas. La clave del asunto está en que el "engrosamiento" de Σ con sus consecuencia trasmite la insatisfacibilidad de Σ; de forma que si llegamos a generar la cláusula vacía por la regla de resolución, Σ será insatisfacible.

Definición 9.1. Sea $\mathbf{L}$ un lenguaje de primer orden. Definimos la relación $\vdash_{rcvc}$ entre conjuntos de cláusulas y cláusulas de $\mathbf{L}$ como sigue:

- **Axiomas**.- No constan.
- **Reglas**.- Constan las siguientes:
  - *regla de resolución*, representada por $\mathrm{res}_{\mathrm{cv}}(\sigma, \tau; \theta)$ donde:
    - λ es una fórmula atómica y σ es una cláusula de la forma $\lambda \vee \sigma_1$.
    - δ es una fórmula atómica y τ es una cláusula de la forma $\neg\delta \vee \tau_1$.
    - θ es una cláusula de la forma $(\sigma_1\Psi \vee \tau_1)\Phi$, donde Ψ es una sustitución de renombramiento tal que $\sigma\Psi$ y τ no tienen símbolos de variable en común y Φ es un unificador de máxima generalidad de $\lambda\Psi$ y δ.

    regla de disminución, representada por $\mathrm{dis}(\sigma; \theta)$ donde:
    - σ es una cláusula de la forma $\lambda \vee \delta \vee \sigma_1$, siendo los literales λ y δ fórmulas atómicas ambas o λ^c y δ^c fórmulas atómicas ambas.
    - θ es una cláusula de la forma $\lambda\Phi \vee \sigma_1\Phi$ y Φ es un unificador de λ y δ (si ambas son fórmulas atómicas) o de λ^c y δ^c (si ambas son fórmulas atómicas).

    Si $\mathrm{dis}(\sigma; \theta)$ decimos que θ es un *factor* o *disminución* de de σ. o

Dado un conjunto de cláusulas Σ y una cláusula γ, $\Sigma \vdash_{rcvc} \gamma$ cuando exista (al menos) una sucesión finita de cláusulas $\gamma_1, \ldots, \gamma_n$ tal que $\gamma_n = \gamma$ y para todo $1 \le i \le n$ valga una de las siguientes alternativas:

- γ_i pertenece a Σ
- existen $j, k < i$ tales que $\mathrm{res}_{\mathrm{cv}}(\gamma_j, \gamma_k; \gamma_i)$
- existe $j < i$ tal que $\mathrm{dis}(\gamma_j; \gamma_i)$

Ejemplo 9.1. De aplicación de la regla de resolución:

$$\mathrm{res}_{\mathrm{cv}}\big(p(x,c) \vee r(x), \neg p(c,c) \vee q(x); r(c) \vee q(x)\big)$$

donde:

- λ es $p(x,c)$
- δ es $p(c,c)$
- Ψ es $(x|y)$ y $\lambda\Psi$ es $p(y,c)$
- Φ es $(y|c)$

Ejemplo 9.2. De aplicación de la regla de resolución:

$$\mathrm{res}_{\mathrm{cv}}\big(p(x, f(x)), \neg p(a,y) \vee r(f(y)); r(f(f(a)))\big)$$

donde:

- λ es $p(x, f(x))$
- δ es $p(a,y)$
- Ψ es ϵ

- Φ es $(y|f(a))(x|a)$

Ejemplo 9.3. De aplicación de la regla de disminución:

$$\text{dis}\big(p(x,g(y)) \vee p(f(c),z) \vee r(x,y,z); p(f(c),g(y)) \vee r(f(c),y,g(y))\big)$$

donde:

- λ es $p(x,g(y))$
- δ es $p(f(c),z)$
- Φ es $(z|g(y))(x|f(c))$

Definición 9.2. Dado un conjunto de cláusulas Σ y una cláusula γ, $\Sigma \vdash_{rcv} \gamma$ (leído γ es consecuencia por resolución de Σ) cuando exista (al menos) una sucesión finita de cláusulas $\gamma_1,\dots,\gamma_n$ tal que $\gamma_n = \gamma$ y para todo $1 \leq i \leq n$ valga una de las siguientes alternativas:

- γ_i pertenece a Σ
- $\text{res}_{\text{cv}}(\alpha_1,\alpha_2;\gamma_i)$, donde existen $j,k < i$ tales que ($\alpha_1 = \gamma_j$ ó $\text{dis}(\gamma_j;\alpha_1)$) y ($\alpha_2 = \gamma_k$ ó $\text{dis}(\gamma_k;\alpha_2)$). A γ_j y γ_k se les llama *cláusulas padre*.

La sucesión finita $\gamma_1,\dots,\gamma_n$ antes mencionada recibe el nombre de Σ-*demostración de* γ *por resolución* o sencillamente *demostración por resolución* si no es necesario nombrar el conjunto Σ. Una Σ-demostración por resolución de $\square$ recibe el nombre de *refutación* de Σ.

Observación 9.1. Dado un conjunto de cláusulas Σ del lenguaje, es claro que $\Sigma \vdash_{rcvc} \square$ si, y sólo si, $\Sigma \vdash_{rcv} \square$, aunque en general ambas relaciones son distintas.

Teorema 9.1. *Sea* $\mathbf{L}$ *un lenguaje de primer orden y* Σ *un conjunto de cláusulas de* $\mathbf{L}$. *Son equivalentes las siguientes afirmaciones:*

1. Σ *es insatisfacible*
2. $\Sigma \vdash_{rcv} \square$

Demostración. Consultar [4], [3] y [7], por ese orden. $\square$

Ejemplo 9.4. Demostremos que el conjunto Σ de cláusulas

$$\{\neg p(x) \vee p(f(x)), p(a), \neg p(f(z))\}$$

es insatisfacible demostrando que $\Sigma \vdash_{rcv} \square$. Para ello basta con la siguiente sucesión finita de cláusulas:

c.1) $\neg p(x) \vee p(f(x))$, hipótesis

c.2) $p(a)$, hipótesis

c.3) $p(f(a))$, por resolución a partir de c.1 y c.2 con $\Phi = (x|a)$

c.4) $\neg p(f(z))$, hipótesis

c.5) $\square$, por resolución a partir de c.3 y c.4 con $\Phi = (z|a)$.

Esta conclusión es a su vez, según sabemos, la evidencia de que:

$$\forall x (p(x) \to p(f(x))), \exists y p(y) \vDash \exists z p(f(z))$$

Pero, entendamos que hemos ido haciendo al tiempo que escribíamos la anterior demostración de 5 cláusulas:

- La cláusula hipótesis $\neg p(x) \vee p(f(x))$, que debe ser entendida como $\forall x(p(x) \to p(f(x)))$, da en particular $p(a) \to p(f(a))$
- Junto a la hipótesis $p(a)$, deducimos $p(f(a))$; pues si vale $p(a) \to p(f(a))$ y vale $p(a)$, debe valer $p(f(a))$
- La hipótesis $\neg p(f(z))$, que debe entenderse como $\forall z \neg p(f(z))$, da en particular $\neg p(f(a))$ y así se obtiene una contradicción, es decir $\square$.

10. Ejemplos

Ejemplo 10.1. Tomado de [7] y tiene que ver con los axiomas de la teoría de grupos. Para evitar usar un símbolo de relación que interprete la igualdad, se introduce $p(x,y,z)$ que se interpreta como $x\cdot y = z$.

φ1) $\forall x\forall y\exists z p(x,y,z)$ (operación interna binaria)

φ2) $\forall x\forall y\forall z\forall u\forall v\forall w\big((p(x,y,u)\wedge p(y,z,v))\rightarrow(p(x,v,w)\leftrightarrow p(u,z,w))\big)$ (asociatividad de la operación interna)

φ3) $\exists x\big(\forall y p(x,y,y)\wedge\forall z\exists u p(u,z,x)\big)$ (existencia de elemento neutro a izquierda y de inverso a derecha)

Planteemos con la fórmula φ la existencia de un inverso a izquierda:

$$\exists x\big(\forall y p(x,y,y)\wedge\forall z\exists u p(z,u,x)\big)$$

La negación de φ es lógicamente equivalente a:

$$\forall x\big(\exists y\neg p(x,y,y)\vee\exists z\forall u\neg p(z,u,x)\big)$$

Consideramos el conjunto de fórmulas $\{\varphi_1,\varphi_2,\varphi_3,\neg\varphi\}$ y lo expresamos en forma de cláusulas. Obtenemos:

σ1) $p(x,y,f(x,y))$

σ2) $\neg p(x,y,u)\vee\neg p(y,z,v)\vee\neg p(x,v,w)\vee p(u,z,w)$

σ3) $\neg p(x,y,u)\vee\neg p(y,z,v)\vee\neg p(u,z,w)\vee p(x,v,w)$

σ4) $p(e,y,y)$

σ5) $p(g(z),z,e)$

σ6) $\neg p(x,h(x),h(x))\vee\neg p(k(x),u,x)$

Deducción de la cláusula vacía:

σ.1) $p(x,y,f(x,y))$, es σ_1

σ.2) $\neg p(x,y,u)\vee\neg p(y,z,v)\vee\neg p(x,v,w)\vee p(u,z,w)$, es σ_2

σ.3) $\neg p(x,y,u)\vee\neg p(y,z,v)\vee\neg p(u,z,w)\vee p(x,v,w)$, es σ_3

σ.4) $p(e,y,y)$, es σ_4

σ.5) $p(g(z),z,e)$, es σ_5

σ.6) $\neg p(x,h(x),h(x))\vee\neg p(k(x),u,x)$, es σ_6

σ.7) $\neg p(k(e),u,e)$, resolución entre σ.4 y σ.6

σ.8) $\neg p(x,y,k(e))\vee\neg p(y,z,v)\vee\neg p(x,v,e)$, resolución entre σ.2d y σ.7

σ.9) $\neg p(g(v),y,k(e))\vee\neg p(y,z,v)$, resolución entre σ.5 y σ.8c

σ.10) $\neg p(g(v),e,k(e))$, resolución entre σ.4 y σ.9b

σ.11) $\neg p(g(v),y,u)\vee\neg p(y,z,e)\vee\neg p(u,z,k(e))$, resolución entre σ.3d y σ.10

σ.12) $\neg p(g(v),y,e)\vee\neg p(y,k(e),e)$, resolución entre σ.4 y σ.11c

σ.13) $\neg p(g(v),g(k(e)),e)$, resolución entre σ.5 y σ.12b

σ.14) $\square$, resolución entre σ.5 y σ.13

Ejemplo 10.2. Retomamos el ejemplo 5.3 que nos proporcionaba las cláusulas:

$\sigma1)$ $\neg p(x) \vee q(x, f(x))$

$\sigma2)$ $\neg p(x) \vee r(f(x))$

$\sigma3)$ $p(a)$

$\sigma4)$ $\neg q(x, y)$

Deducción de la cláusula vacía

$\sigma.1)$ $\neg p(x) \vee q(x, f(x))$, es σ_1

$\sigma.2)$ $p(a)$, es σ_2

$\sigma.3)$ $q(a, f(a))$, resolución entre $\sigma.1$ y $\sigma.2$

$\sigma.4)$ $\neg q(x, y)$, es σ_4

$\sigma.5)$ $\square$, resolución entre $\sigma.3$ y $\sigma.4$

Ejemplo 10.3. Tomado de [4] para mostrar el progreso que supone la resolución con respecto a los métodos primitivos basados en el Teorema de Herbrand. Consideremos las dos cláusulas siguientes:

- $p(a, x_2, f(x_2), x_4, f(x_4), \ldots, x_{2n}, f(x_{2n}))$
- $\neg p(x_1, f(x_1), x_3, f(x_3), x_5, \ldots, f(x_{2n-1}), x_{2n+1})$

La resolución da la cláusula vacía en una etapa. El método de Herbrand, considerando que las fórmulas son clasificadas primero todas las que usan a, después todas las que utilizan a y $f(a)$, etc. no nos llevaría a la cláusula vacía buscada más que con el conjunto $\{\sigma_0, \sigma_1, \ldots, \sigma_m\}$ para $m = 2 \cdot (2n)^{2n}$ al menos. Tomando $n = 20$ se obtiene un valor que sobrepasa todo lo que podemos eperar tratar con máquinas, incluso en un futuro lejano.

Ejemplo 10.4. Tomado de [4] para mostrar un ejemplo sencillo de trabajo en base de datos. Se quiere establecer a partir de los enunciados hipotéticos siguientes:

- para todo crimen, hay alguien que lo ha cometido
- sólo la gente deshonesta comete crímenes
- no es detenida más que la gente deshonesta
- la gente deshonesta detenida no comete crímenes
- ocurren crímenes

el enunciado:

- hay gente deshonesta no detenida

como pretendida tesis. A tal efecto introducimos las siguientes abreviaturas:

1. $det(y)$ por "y está detenido"
2. $des(y)$ por "y es deshonesta"
3. $com(y, x)$ por "y comete x"
4. $cri(x)$ por "x es un crimen"

La traducción de los enunciados hipotéticos y la negación de la pretendida tesis da:

$\varphi1)$ $\forall x(cri(x) \rightarrow \exists y com(y, x))$

$\varphi 2)$ $\forall y \forall x((cri(x) \wedge com(y,x)) \to des(y))$

$\varphi 3)$ $\forall y(det(y) \to des(y))$

$\varphi 4)$ $\forall y((des(y) \wedge det(y)) \to \neg \exists x(cri(x) \wedge com(y,x)))$

$\varphi 5)$ $\exists x cri(x)$

$\varphi 6)$ $\neg \exists y(des(y) \wedge \neg det(y))$

que pasamos a cláusulas:

$\sigma 1)$ $\neg cri(x) \vee com(f(x),x)$

$\sigma 2)$ $\neg cri(x) \vee \neg com(y,x) \vee des(y)$

$\sigma 3)$ $\neg det(y) \vee des(y)$

$\sigma 4)$ $\neg des(y) \vee \neg det(y) \vee \neg cri(x) \vee \neg com(y,x)$

$\sigma 5)$ $cri(a)$

$\sigma 6)$ $\neg des(y) \vee det(y)$

Para establecer que la pretendida tesis es ciertamente consecuencia de nuestras hipótesis construimos una deducción por resolución con variables:

$\sigma.1)$ $\neg cri(x) \vee com(f(x),x)$

$\sigma.2)$ $\neg cri(x) \vee \neg com(y,x) \vee des(y)$

$\sigma.3)$ $\neg det(y) \vee des(y)$

$\sigma.4)$ $\neg des(y) \vee \neg det(y) \vee \neg cri(x) \vee \neg com(y,x)$

$\sigma.5)$ $cri(a)$

$\sigma.6)$ $\neg des(y) \vee det(y)$

$\sigma.7)$ $com(f(a),a)$, res. 5 y 1, alguien ha cometido el crimen

$\sigma.8)$ $\neg cri(a) \vee des(f(a))$, res. 7 y 2

$\sigma.9)$ $des(f(a))$, res. 8 y 5, este alguien es deshonesto

$\sigma.10)$ $\neg des(f(a)) \vee \neg det(f(a)) \vee \neg cri(a)$, res. 7 y 4

$\sigma.11)$ $\neg des(f(a)) \vee \neg det(f(a))$, res. 10 y 5

$\sigma.12)$ $\neg det(f(a))$, res. 11 y 9, ese alguien no está detenido

$\sigma.13)$ $\neg des(f(a))$, res. 12 y 6, ese alguien no es deshonesto (en contradicción con 9)

$\sigma.14)$ $\square$, res. 13 y 9

Ejemplo 10.5. Tomado de [4] para ejemplificar el uso de la regla de disminución, la cual es indispensable. Consideremos las cláusulas:

- $p(x) \vee p(y)$
- $\neg p(x) \vee \neg p(y)$

Es imposible deducir la cláusula vacía utilizando solamente la regla de resolución, pues la resolución aplicada a dos cláusulas de dos literales da una cláusula de dos literales. Sin embargo, con la regla de disminución se tiene la deducción siguiente:

σ.1) $p(x) \vee p(y)$, hipótesis

σ.2) $\neg p(x) \vee \neg p(y)$, hipótesis

σ.3) $\neg p(y)$, resolución entre 2 y la disminución $p(x)$ de 1

σ.4) $\square$, resolución entre 3 y la disminución $p(x)$ de 1

Ejemplo 10.6. La sustitución de renombramiento, Ψ, de la definición de la regla de resolución es imprescindible en determinados casos, aunque éstos no queden detallados en el enunciado de la regla. Considerar el caso del conjunto de cláusulas:

$$\begin{aligned} \Sigma = \{ & p(a) \vee q(x,y), \\ & \neg p(a) \vee r(y), \\ & \neg r(f(a)), \\ & \neg q(x, f(b)) \} \end{aligned}$$

Si iniciamos el intento de refutación desde la cláusula $p(a) \vee q(x,y)$ y decidimos confrontarla con $\neg p(a) \vee r(y)$, es posible encontrar una refutación que comience en estas dos cláusulas:

- $\gamma_0 = p(a) \vee q(x,y)$; hip.
- $\beta_0 = \neg p(a) \vee r(y)$; hip.
- $\gamma_1 = q(x,y_1) \vee r(y)$; $res_{cv}(\gamma_0, \beta_0, \gamma_1)$, con $\Psi = (y|y_1)$ aplicada a γ_0 y $\Phi = \epsilon$ —la idendtidad— aplicada a $q(x,y_1) \vee r(y)$
- $\beta_1 = \neg q(x, f(b))$; hip.
- $\gamma_2 = r(y)$; $res_{cv}(\gamma_1, \beta_1, \gamma_2)$ con $\Psi = (x|x_1)$ aplicada a $\neg q(x, f(b))$ quedando $(\neg q(x, f(b)))(x|x_1) = \neg q(x_1, f(b))$ y aplicando el unificador principal $\Psi = (x_1|x)(y_1|f(b))$ de $q(x,y_1)$ y $q(x_1, f(b))$ a $r(y)$. Este momento es clave, aquí se puede observar cómo gracias al primer renombramiento $(y|y_1)$, $r(y)$ ha quedado a salvo del unificador recién obtenido; y ello nos da la "flexibilidad" necesaria para poder serguir resolviendo.
- $\beta_2 = \neg r(f(a))$; hip.
- $\square$ por resolución entre γ_2 y β_2 con $\Psi = \epsilon$ y $\Phi = (y|f(a))$.

Si hubiésemos aplicado erróneamente y descuidadamente las definiciones, tendríamos el **siguiente razonamiento incorrecto**:

- $\gamma_0 = p(a) \vee q(x,y)$; hip.
- $\beta_0 = \neg p(a) \vee r(y)$; hip.
- $\gamma_1 = q(x,y) \vee r(y)$; **¡incorrecto!** por no renombrar.
- $\beta_1 = \neg q(x, f(b))$; hip.
- $\gamma_2 = r(f(b))$; resolvente entre γ_1 y β_1.
- $\beta_2 = \neg r(f(a))$; hip.
- $\square$ por resolución entre γ_2 y β_2 con $\Psi = \epsilon$ y $\Phi = (a|b)$. **¡Incorrecto!** pues al no ser a un símbolo de variable, $(a|b)$ no es una sustitución. En realidad, no hay forma de obtener $\square$ entre γ_2 y β_2

La administración de la regla de resolución en los sistemas de demostración automática de teoremas ha sido utilizada con dos técnicas esencialmente:

- la técnica de gestión de conjuntos de cláusulas
- la técnica de exploración del árbol de las deducciones

No obstante, cada una de estas técnicas ha dado lugar a numerosas variantes.

En general, dada una técnica nos interesa de ella su *corrección*, su *completitud* y su *eficiencia*. Una técnica es *correcta* cuando no conduce jamás a considerar como insatisfacible un conjunto de cláusulas que es satisfacible. Una técnica es *completa* cuando termina revelando que es insatisfacible el conjunto que trata, si realmente lo es. La eficiencia es un concepto relativo.

11. Gestión por Saturación

De entre las estrategias, la más simple y natural es la *estrategia de saturación*:

- Partimos de un conjunto de cláusulas Σ_0 del cual buscamos conocer si es satisfacible o no.
- Dado $0 < i$, efectuamos todas las resoluciones y todas las disminuciones posibles a partir de las cláusulas de Σ_{i-1}, lo que genera unas nuevas cláusulas que añadimos a Σ_{i-1} para obtener el conjunto Σ_i.
- Se detiene el proceso cuando $\square$ es un elemento de Σ_i o cuando $\Sigma_i = \Sigma_{i+1}$.

La estrategia de gestión por saturación es completa y correcta según el Teorema 9.1. Sin embargo, no es eficiente; en realidad, desde el punto de vista teórico no supone mejora alguna respecto al algoritmo primario sugerido en la Observación 6.3 y es profuso en la generación y uso de cláusulas de poca utilidad, de hecho los conjuntos Σ_i crecen de forma exponencial. Ejemplo de ineficiciencia es el tratamiento que da al conjunto $\Sigma_0 = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$ (desarrollado en [3], pag. 92 y 93). Además, es posible que siendo Σ_0 satisfacible sin embargo sea Σ_i distinto de Σ_{i+1} para todo $0 \leq i$; es el caso de $\Sigma_0 = \{p(a), \neg p(x) \vee p(f(x))\}$.

Ejemplo 11.1. Queremos saber si es insatisfacible el conjunto $\{p, \neg p \vee q, \neg q \vee p, \neg q\}$. Entonces generamos:

- $\Sigma_0 = \{p, \neg p \vee q, \neg q \vee p, \neg q\}$
- $\Sigma_1 = \Sigma_0 \cup \{q, \neg q \vee q, \neg p \vee p, \neg p\}$
- $\Sigma_2 = \Sigma_1 \cup \{p, \square, \ldots\}$

12. Gestión por Saturación con Simplificación

Definición 12.1. La cláusula $\lambda_1 \vee \cdots \vee \lambda_k$ (en realidad la sentencia $\forall x_1 \cdots \forall x_n (\lambda_1 \vee \cdots \vee \lambda_k)$, no lo olvidemos) distinta de la vacía es una *cláusula tautológica* cuando existen $1 \leq i \neq j \leq k$ tales que $\lambda_i = \lambda_j^c$.

Ejemplo 12.1. La cláusula

$$\forall x\big(p(f(x)) \vee q(x) \vee \neg r(x) \vee \neg p(f(x))\big)$$

es tautológica.

Definición 12.2. La cláusula σ *subyace bajo* la cláusula τ (o también τ *generaliza* a σ) si existe una sustitución Φ tal que τ es la cláusula $\xi \vee \Phi\sigma$, para cierta ξ.

Ejemplo 12.2.

1. $p(f(a)) \vee q(a)$ generaliza a $p(x)$
2. $p(a) \vee q(a) \vee r(f(x)) \vee r(b)$ generaliza a $p(x) \vee q(a)$

3. $\neg p(x) \vee q(f(x), a)$ subyace bajo $\neg p(h(y)) \vee q(f(h(y)), a) \vee \neg p(z)$, basta con considerar la sustitución $(x|h(y))$

Teorema 12.1. *Sea Σ un conjunto de cláusulas y Σ' el conjunto obtenido de Σ suprimiendo en él todas las cláusula tautológicas. Son equivalentes las siguientes afirmaciones:*

1. *Σ es satisfacible*
2. *Σ' es satisfacible*

Teorema 12.2. *Sea Σ un conjunto de cláusulas sin cláusulas tautológicas y Σ' el conjunto obtenido de Σ suprimiendo cualquier cláusula que generalize a (bajo la que subyazca) otra cláusula del conjunto. Son equivalentes las siguientes afirmaciones:*

1. *Σ es satisfacible*
2. *Σ' es satisfacible*

Los teoremas 12.1 y 12.2 sugieren una modificación interesante en la estrategia de gestión por saturación explicado anteriormente (ver sección 11), que justamente atenua el efecto de generar cláusulas inútiles. Se procede como se indicó en la estrategia de saturación, pero antes de cada etapa de saturación se suprimen las tautologías y las generalizaciones. Esta nueva *estrategia de saturación con simplificación* es correcta, porque lo era la estrategia de saturación, y completa por los teoremas 12.1 y 12.2.

Ejemplo 12.3. Estudiamos la satisfacibilidad del conjunto $\{r(a) \vee p(a), p(x) \vee \neg r(b), r(b), \neg p(b), r(a) \vee r(c) \vee \neg r(a)\}$ y empleamos la estrategia de saturación con simplificación.

- $\Sigma_0 = \{r(a) \vee p(a), p(x) \vee \neg r(b), r(b), \neg p(b), r(a) \vee r(c) \vee \neg r(a)\}$
- $\Sigma_0' = \{r(a) \vee p(a), p(x) \vee \neg r(b), r(b), \neg p(b)\}$
- $\Sigma_1 = \{r(a) \vee p(a), p(x) \vee \neg r(b), r(b), \neg p(b), p(x), \neg r(b)\}$
- $\Sigma_1' = \{r(b), \neg p(b), p(x), \neg r(b)\}$
- $\Sigma_2 = \{\square, \ldots\}$

Hay otras simplificaciones posibles utilizando los "predicados evaluables".

13. Gestión por Preferencia de Cláusulas Simples

En el método sugerido en la sección 11, mejor que construir de una vez todas las cláusulas que pueden ser obtenidas a partir de Σ_i, podemos no construir más que una sola, "bien elegida", añadirla y después volver a comenzar. No obstante, ¿qué significa "bien elegida"? Es una expresión ambigua que da pie a muchos métodos: se puede proceder al azar; se puede elegir la obtenida en primer lugar, cuando hay un orden establecido entre todas las resolventes y disminuciones posibles (por ejemplo, numerando las cláusalas o los predicados), etc.

Pero entre todas las estrategias, la más natural es sin duda la que para pasar de Σ_i a Σ_{i+1} une la cláusula más corta entre las que se pueden obtener con Σ_i. Esta idea se sustenta en que como queremos obtener la cláusula vacía, alcanzamos nuestro objetivo tanto más rápido como cortas sean las cláusulas que se generan.

Ejemplo 13.1.

- $\Sigma_0 = \{r(x) \vee \neg p(x), \neg r(b), \neg r(c) \vee q(x) \vee r(f(a)), p(b)\}$
- $\Sigma_1 = \Sigma_0 \cup \{\neg p(b)\}$

- $\Sigma_2 = \Sigma_1 \cup \{\square\}$

Aunque correcta, esta estrategia usada en toda su pureza no es completa. En efecto, consideremos el siguiente ejemplo:

Ejemplo 13.2.

- $\Sigma_0 = \{\neg p(x) \vee p(f(x)), p(a), \neg p(a) \vee \neg p(b) \vee \neg p(c), p(b), p(c)\}$
- $\Sigma_1 = \Sigma_0 \cup \{\neg p(f(a))\}$
- $\Sigma_2 = \Sigma_1 \cup \{\neg p(f(f(a)))\}$

Nunca obtenemos la cláusula vacía, salvo que pasemos por la obtención de una cláusula de dos literales, lo cual da la cláusula vacía en tres etapas.

Para hacer esta estratégia completa podemos, por ejemplo, introducir de cuando en cuando una etapa de saturación.

Entre las variantes posibles, destacamos la estrategia de "preferir las cláusulas unitarias" en la que se efectuan prioritariamente todas las resoluciones que hacen intervenir a una cláusula unit. Tal cual esta estrategia no es completa.

14. Exploración. Conceptos Generales

Definición 14.1. Sea Σ un conjunto de cláusulas de un lenguaje de primer orden. El *árbol de deducciones* de Σ es el árbol etiquetado —en nodos— con raíz (la raíz es un símbolo ajeno al lenguaje, por ejemplo $\bullet$), tal que todo segmento inicial suyo es, en cuanto a la lista ordenada de sus etiquetas excluyendo la raíz, una Σ demostración por resolución. Una *estrategia de exploración* de árboles estará determinada fijando (cfr. [4] para ver ejemplos):

- un subárbol del árbol de las deducciones, normalmente esto se hará imponiendo limitaciones sobre las deducciones que se habrán de tener en cuenta.
- el procedimiento de exploración del subárbol :
  - *primero en profundidad con retroceso* al nodo anterior: ello permite llegar a un nodo profundo más rápidamente, se programa fácilmente y no emplea demasiado espacio de memoria. Si no se fija profundidad, este método de exploración se pierde en la primera rama infinita que alcanza y no permite la exploración entera de un árbol finito. Si en la exploración se fija una profundidad corremos el riesgo de no encontrar un objetivo en una rama por no haber profundizado lo suficiente.
  - *primero en anchura*: ello permite alcanzar cualquier nodo con tal de disponer de tiempo suficiente. La exploración se puede hacer lenta si los nodos tienen en general muchos hijos, con lo cual puede ser muy costoso en tiempo profundizar en el árbol.

Las estrategias generales son análogas a la estrategia de saturación. Consisten simplemente en explorar un árbol en el que han sido incluídas todas las demostraciones por resolución. Si la exploración se hace primero en anchura se obtiene una estrategia correcta y completa. Si se hace primero en profundidad se obtiene una estrategia correcta pero no completa. Estas estrategias son muy ineficaces, más aún que las de saturación, porque consideran diferentes las deducciones que no cambian más que el orden de las fórmulas. Por ello es necesario restringirse en la búsqueda dentro del árbol global a un subárbol suyo en que el aparece la cláusula vacía.

15. Estrategias Lineales

Definición 15.1. Sea Σ un conjunto de cláusulas, σ_0 un elemento de Σ y γ una cláusula. Una Σ-*demostración lineal* de γ con raíz en σ_0 es una Σ-demostración de γ

$$< \gamma_0, \beta_0, \gamma_1, \beta_1 \ldots, \gamma_{n-1}, \beta_{n-1}, \gamma_n >$$

donde:

- γ_0 es σ_0,
- para todo $0 \leq i \leq n-1$, γ_{i+1} es obtenido por resolución a partir de γ_i (denominada *cláusula central*) y β_i (denominada *cláusula lateral*), o a partir de algún factor de alguna de ellas o de cada una de ellas.
- Cada β_i es un elementos de Σ o existe $j < i$ tal que $\beta_i = \gamma_j$.

Ejemplo 15.1. Tomado de [3]. Consideremos el conjunto

$$\begin{aligned}\Sigma = \{&r(a, f(c), f(b)),\\ &p(a),\\ &r(x, x, f(x)),\\ &\neg r(x, y, z) \vee r(y, x, z),\\ &\neg r(x, y, z) \vee q(x, z),\\ &\neg p(w) \vee \neg r(x, y, z) \vee \neg q(w, z) \vee q(w, x) \vee q(w, y),\\ &\neg q(a, b)\}\end{aligned}$$

y sea $\sigma_0 = \neg q(a, b)$. La siguiente es una Σ-demostración lineal con raíz σ_0:

γ_0: $\neg q(a, b)$

β_0: $\neg p(w) \vee \neg r(x, y, z) \vee \neg q(w, z) \vee q(w, x) \vee q(w, y)$

γ_1: $\neg p(a) \vee \neg r(x, b, z) \vee \neg q(a, z) \vee q(a, x)$

β_1: $\neg q(a, b)$

γ_2: $\neg p(a) \vee \neg r(b, b, z) \vee \neg q(a, z)$

β_2: $r(x, x, f(x))$

γ_3: $\neg p(a) \vee \neg q(a, f(b))$

β_3: $p(a)$

γ_4: $\neg q(a, f(b))$

β_4: $\neg r(x, y, z) \vee q(x, z)$

γ_5: $\neg r(a, y, f(b))$

β_5: $r(a.f(c), f(b))$

γ_6: $\square$

Ejemplo 15.2. Tomado de [4]. Consideremos el conjunto $\Sigma = \{\neg a \vee \neg b, a \vee \neg c, c, b \vee \neg d, d \vee b\}$ y sea $\sigma_0 = \neg a \vee \neg b$. La siguiente es una Σ-demostración lineal con raíz σ_0:

γ_0: $\neg a \vee \neg b$

β_0: $a \vee \neg c$

γ_1: $\neg b \vee \neg c$

β_1: c

γ_2: $\neg b$

β_2: $b \vee \neg d$

γ_3: $\neg d$

β_3: $d \vee b$

γ_4: b

β_4: $\neg b$

γ_5: $\square$

No obstante, no es ésta la única Σ-demostración lineal con raíz en σ_0, pues también existe la siguiente:

γ_0: $\neg a \vee \neg b$

β_0: $a \vee \neg c$

γ_1: $\neg b \vee \neg c$

β_1: $d \vee b$

γ_2: $\neg c \vee d$

β_2: $b \vee \neg d$

γ_3: $b \vee \neg c$

β_3: c

γ_4: b

β_4: $\neg b \vee \neg c$

γ_5: $\neg c$

β_5: c

γ_5: $\square$

El árbol de todas la Σ-demostraciones lineales con raíz σ_0 es ya demasiado grande, aunque se podría visualizar una parte (cfr. [4], pag. 175)

Se ha establecido (cfr. [3] y [7]) que si existe una demostración por resolución de la cláusula vacía a partir de Σ, entonces existe un Σ-demostración lineal de la cláusula vacía.

Más aún, si $\Sigma = \Sigma' \cup \{\sigma_0\}$ con Σ' satisfacible y Σ insatisfacible, entonces existe una Σ-demostración lineal de la cláusula vacía con raíz σ_0. Esto quiere decir que buscamos saber si el literal τ es consecuencia de Σ', basta con recorrer el árbol de las demostraciones lineales con raíz $\sigma_0 = \tau^c$.

Si se utiliza una estrategia de exploración "primero en anchura" estamos seguros de obtener la cláusula vacía de estar (estrategia completa). Por contra si se utiliza la estrategia de "primero en profundidad con retorno al nodo anterior" podemos hundirnos en una rama infinita y no encontrar la cláusula vacía, aunque esté. El ejemplo más conocido es es siguiente:

Ejemplo 15.3. Sea $\Sigma = \{\neg p(x) \vee p(f(x)), \neg p(a), p(x)\}$ y tomemos como ráiz a $\sigma_0 = p(x)$. En el árbol de las Σ-demostraciones lineales con raíz σ_0 se encuentran estas dos:

γ_0: $p(x)$

β_0: $\neg p(a)$

γ_1: $\square$

y “ésta” otra tan larga como se quiera, pero nunca concluyente:

γ_0: $p(x)$

β_0: $\neg p(w) \vee p(f(w))$

γ_1: $p(f(x))$

β_1: $\neg p(w) \vee p(f(w))$

γ_2: $p(f(f(x)))$

β_2: $\cdots$

16. Estrategia “Input”

Definición 16.1. Sea Σ un conjunto de cláusulas, σ_0 un elemento de Σ y γ una cláusula. Una Σ-*demostración input* de γ con raíz en σ_0 es una Σ-demostración lineal de γ

$$< \gamma_0, \beta_0, \gamma_1, \beta_1 \ldots, \gamma_{n-1}, \beta_{n-1}, \gamma_n >$$

donde para todo $1 \leq i$, γ_i es obtenida por una resolución en la cual una de las cláusulas padre es de Σ.

Ejemplo 16.1. Consideremos el conjunto de cláusulas

$$\begin{aligned}\Sigma = \{&\neg p(x,y,u) \vee \neg p(y,z,v) \vee \neg p(x,v,w) \vee p(u,z,w),\\ &p(g(x,y),x,y),\\ &p(x,h(x,y),y),\\ &\neg p(k(x),x,k(x))\}\end{aligned}$$

Tomando como σ_0 la cláusula $\neg p(x,y,u) \vee \neg p(y,z,v) \vee \neg p(x,v,w) \vee p(u,z,w)$ tenemos la siguiente refutación input de Σ.

γ_0: $\neg p(x,y,u) \vee \neg p(y,z,v) \vee \neg p(x,v,w) \vee p(u,z,w)$

β_0: $p(g(x,y),x,y)$

γ_1: $\neg p(y,z,v) \vee \neg p(g(y,u),v,w) \vee p(u,z,w)$

β_1: $p(g(x,y),x,y)$

γ_2: $\neg p(v,z,v) \vee p(w,z,w)$

β_2: $p(x,h(x,y),y)$

γ_3: $p(w,h(v,v),w)$

β_3: $\neg p(k(x),x,k(x))$

γ_4: $\square$

Ejemplo 16.2. Sea $\Sigma = \{\neg a, a \vee \neg b, a \vee \neg c \vee \neg d, c, d \vee \neg c\}$ y $\sigma_0 = \neg a$. La Σ-demostración lineal siguiente es input:

γ_0: $\neg a$

β_0: $a \vee \neg c \vee \neg d$

γ_1: $\neg c \vee \neg d$

β_1: c

γ_2: $\neg d$

β_2: $d \vee \neg c$

γ_3: $\neg c$

β_3: c

γ_4: $\square$

De hecho no hay más Σ-demostraciones input con raíz en σ_0 que la anterior y

γ_0: $\neg a$

β_0: $a \vee \neg b$

γ_1: $\neg b$

El Ejemplo 16.2 muestra como, aún en un caso sencillo en el que el árbol resulta finito, no es posible prescidir del "retroceso al nodo anterior" en la estrategia de "primero en profundidad" si se quiere encontrar la cláusula vacía.

Contrariamente al caso general de la estrategia lineal, incluso una estrategia de exploración "primero en anchura" no resulta completa. Es el caso de $\Sigma = \{a \vee b, \neg a \vee b, a \vee \neg b, \neg a \vee \neg b\}$ que el árbol de las deducciones input no tiene a la cláusula vacía mientras que el árbol de todas las deducciones sí la tiene.

Definición 16.2. Una cláusula distinta de la vacía es una *cláusula de Horn* si posee exactamente un literal positivo. Un *conjunto de Horn* es un conjunto de cláusulas cada de una de las cuales es una cláusula de Horn. Una cláusula no vacía es *negativa* si cada uno de sus literales es negativo.

Teorema 16.1. *Sea $\Sigma = \Sigma' \cup \{\sigma_0\}$ un conjunto insatisfacible de cláusulas. Si σ_0 es negativa y Σ' es un conjunto de Horn, entonces existe una Σ-demostración input de $\square$ con raíz σ_0 para la que no se utiliza la regla de disminución.*

Demostración. Consultar [5] $\square$

Definición 16.3. Sea Σ un conjunto de cláusulas, σ_0 un elemento de Σ y γ una cláusula. Una *Σ-demostración unit* de γ con raíz en σ_0 es una Σ-demostración lineal de γ

$$< \gamma_0, \beta_0, \gamma_1, \beta_1 \ldots, \gamma_{n-1}, \beta_{n-1}, \gamma_n >$$

donde para todo $1 \leq i$, γ_i se obtiene por resolución a partir de al menos una cláusula padre unit o un factor unit de alguna cláusula padre.

Teorema 16.2. *Existe una refutación unit de un conjunto Σ de cláusulas si, y sólo si, existe una refutación input de Σ.*

Demostración. Se puede ver en [3] $\square$

Ejemplo 16.3. Se pide demostrar que la fórmula $\exists z r(z)$ es consecuencia de las hipótesis: $\forall x(p(x) \rightarrow p(f(x)))$, $\forall y((p(f(y)) \wedge p(y)) \rightarrow r(y))$ y $p(a)$. En términos de cláusulas, este problema es equivalente a saber si es insatisfacible o no el conjunto $\Sigma = \Sigma' \cup \{\sigma_0\}$, donde:

$$\Sigma' = \{\neg p(x) \vee p(f(x)), \neg p(f(y)) \vee \neg p(y) \vee r(y), p(a)\}$$

y $\sigma_0 = \neg r(z)$. Encontramos la siguiente refutación unit (e input):

γ_0: $\neg r(z)$

β_0: $\neg p(f(y)) \vee \neg p(y) \vee r(y)$

γ_1: $\neg p(f(y)) \vee \neg p(y)$

β_1: $p(a)$

γ_2: $\neg p(f(a))$

β_2: $\neg p(x) \vee p(f(x))$

γ_3: $\neg p(a)$

β_3: $p(a)$

γ_4: $\square$

Observación 16.1. La consecuencia esencial del Teorema 16.1 es que para la clase de los problemas de la forma $\Sigma = \Sigma' \cup \{\sigma_0\}$, con Σ' un conjunto de cláusulas de Horn y σ_0 una cláusula negativa, la estrategia de exploración "primero en anchura" del árbol de deducciones input de raíz σ_0 es una estrategia correcta y completa.

La estrategia de exploración "primero en profundidad con retorno al nodo anterior" del mismo árbol es correcta y completa para todos los problemas antes citados que engendren un árbol de deducción input finito (y por tanto, para todos los problemas sin variables), pero no es completo en general.

Observación 16.2. Observar que cualquier sentencia de la forma

$$\varphi_1 \wedge \cdots \wedge \varphi_n \rightarrow \psi \tag{22}$$

se escribe de forma equivalente como

$$\neg\varphi_1 \vee \cdots \vee \neg\varphi_n \vee \psi$$

que bien podría originar una cláusula de Horn. Por otra parte, toda sentencia de la forma

$$\psi_1 \wedge \cdots \wedge \psi_m \tag{23}$$

al ser negada se convierte en

$$\neg\psi_1 \vee \cdots \vee \neg\psi_m$$

que bien podría originar una cláusula negativa. Casi todos los problemas naturales de implicación tienen por hipótesis sentencias (en cantidad finita) de la forma 22 y como tesis una fórmula de la forma 23. Además, la naturalidad del problema hace que las hipótesis "tengan modelo". Esto realza la importancia del Teorema 16.1.

17. Estrategias Ordenadas

Existen numerosas formas de tener en cuenta el orden de los literales en las cláusulas para limitar las demostraciones tomadas en consideración dentro del árbol de las deducciones (cfr. [7] y [3]). Describimos aquí la más simple.

En la *resolución ordenada* las cláusulas son cosideradas como sucesiones finitas de literales (sus literales están ordenados). Para poder llevar a cabo la resolución entre dos cláusulas tienen que concurrir todas las circunstancias que requiere la aplicación de la regla de resolución; pero además, la resolución tiene que ser practicada con los literales que están en cabeza de las cláusulas. Más aún, la resolvente, que debe estar ordenada, debe llevar en las primeras posiciones los literales que resultan de la cláusula resolvente con el literal positivo y después los literales de la otra. Se denomina *demostración ordenada* a toda demostración en la que siempre que se resuelve se utiliza la resolución ordenada.

Ejemplo 17.1. Es posible la resolución ordenada entre las cláusulas $p(x) \vee r(x) \vee \neg q(x,y)$ y $\neg p(a) \vee \neg r(f(a))$ y da como resultado $r(a) \vee \neg q(a,y) \vee \neg r(f(a))$. Sin embargo, no es posible la resolución ordenada entre $p(x) \vee r(x) \vee \neg q(x,y)$ y $\neg r(f(a)) \vee \neg p(a)$.

Ejemplo 17.2. Sea $\Sigma = \{a \vee b, \neg a \vee c, \neg b \vee d, \neg c, \neg d\}$. La siguiente es una Σ-demostración ordenada de $\square$ con raíz $a \vee b$:

γ_0: $a \vee b$

β_0: $\neg a \vee c$

γ_1: $b \vee c$

β_1: $\neg b \vee d$

γ_2: $c \vee d$

β_2: $\neg c$

γ_3: d

β_3: $\neg d$

γ_4: $\square$

Teorema 17.1. *Sea $\Sigma = \Sigma' \cup \{\sigma_0\}$ un conjunto insatisfacible de cláusulas, donde σ_0 es negativa y Σ' es un conjunto de cláusulas de Horn ordenadas colocando el literal positivo en cabeza de las cláusulas. Entonces existe una refutación input ordenada de raíz σ_0.*

Demostración. Ver, por ejemplo, [6] o [1]. □

Ejemplo 17.3. Sea

$$\Sigma' = \{p(x) \vee \neg r(x), p(x) \vee \neg q(f(x)) \vee \neg q(x), q(f(x)) \vee \neg q(x), q(a)\}$$

y $\sigma_0 = \neg p(a)$. El árbol de las demostraciones input ordenadas de raíz σ_0 no tiene más que dos ramas, una no es una refutación:

γ_0: $\neg p(a)$

β_0: $p(x) \vee \neg r(x)$

γ_1: $\neg r(a)$

y la otra sí lo es:

γ_0: $\neg p(a)$

β_0: $p(x) \vee \neg q(f(x)) \vee \neg q(x)$

γ_1: $\neg q(f(a)) \vee \neg q(a)$

β_1: $q(f(x)) \vee \neg q(x)$

γ_2: $\neg q(a) \vee \neg q(a)$

β_2: $q(a)$

γ_3: $\neg q(a)$

β_3: $q(a)$

γ_4: $\square$

Observación 17.1. En resumen, para la clase de problemas de la forma $\Sigma = \Sigma' \cup \{\sigma_0\}$, donde Σ' es un conjunto de cláusulas de Horn y σ_0 es una cláusula negativa, la estrategia de exploración "primero en anchura" del árbol de las deducciones input ordenadas de raíz σ_0 es una estrategia correcta y completa. La estrategia de exploración "primero en profundidad con retorno" del mismo árbol es completa y correcta para todos los problemas antes citados que engendren un árbol de deducciones input ordenadas finito.

Observación 17.2. El lenguaje *Prolog* está fundado exactamente en el principio de la Observación 17.1: no se pueden escribir más que *cláusulas de Horn* y cuando es propuesta una cuestión, se pone en marcha un algoritmo de exploración "primero en profundidad con retorno" del árbol de deducciones input ordenadas de raíz la negación de la cuestión propuesta. Si el árbol es finito, está asegurada por lo dicho la obtención o bien de la cláusula vacía —es decir, un éxito— si la cuestión propuesta admite una respuesta positiva, o bien un fallo si —cuando todo el árbol es recorrido sin ser encontrada la cláusula vacía— si la cuestión propuesta admite una respuesta negativa. Dicho de otra forma, la estrategia de utilización de la resolución de Prolog es incompleta en general y completa cuando el árbol de deducciones particular engendrado es finito.

18. Estrategias Input Ordenadas y Extracción de Respuestas

Consideremos el ejemplo siguiente:

$$\begin{aligned}\Sigma' =& \{p(a), p(b), r(f(x)) \vee \neg p(x), q(y) \vee \neg r(y), q(c)\}\\ \sigma_0 =& \neg q(z)\end{aligned}$$

El árbol de deducciones tiene tres deducciones cada una de las cuales conduce a la cláusula vacía (bibujar el árbol como ejercicio). Una de ellas es la siguiente:

γ_0: $\neg q(z)$

β_0: $q(y) \vee \neg r(y)$ (sutitución: $(y|z)$)

γ_1: $\neg r(z)$

β_1: $r(f(x)) \vee \neg p(x)$ (sutitución: $(z|f(x))$)

γ_2: $\neg p(x)$

β_2: $p(a)$ (sutitución: $(x|a)$)

γ_3: $\square$

En el curso de la demostración por resolución la variable z es reemplazada por $f(x)$ y seguidamente x por a; así pues z es reemplazada por $f(a)$. Por tanto, $\Sigma' \cup \{\neg q(f(a))\}$ conduce a la cláusula vacía por resolución:

γ_0: $\neg q(f(a))$

β_0: $q(y) \vee \neg r(y)$

γ_1: $\neg r(f(a))$

β_1: $r(f(x)) \vee \neg p(x)$

γ_2: $\neg p(a)$

β_2: $p(a)$

γ_3: $\square$

lo que significa que $q(f(a))$ resulta de Σ'. Por la misma razón, pero utilizando las otras refutaciones $q(f(b))$ y $q(c)$ resultan de Σ'. Se constata pues que cuando se consideran cláusulas con variables (como $\neg q(z)$), cada demostración de la cláusula vacía da un objeto t (un elemento del universo de Herbrand) tal que $q(t)$ resulta de Σ'. Esto es general y se tiene el resultado siguiente.

Teorema 18.1. *Sea $\Sigma' \cup \{\sigma_0(x_1, \ldots, x_n)\}$ un conjunto insatisfacible de cláusulas tal que todas las cláusulas de Σ' son de Horn ordenadas anteponiendo el literal positivo y*

$$\begin{aligned}\sigma_0(x_1, x_2, \ldots, x_n) &= \neg\varphi(x_1, x_2, \ldots, x_n)\\ &= \neg\lambda_1(x_1, x_2, \ldots, x_n) \vee \neg\lambda_2(x_1, x_2, \ldots, x_n) \vee \cdots \neg\lambda_k(x_1, x_2, \ldots, x_n)\end{aligned}$$

es negativa. Entonces cada refutación input ordenada en la que las sustituciones son $(x_1|t_1)$, $(x_2|t_2)$,...,$(x_n|t_n)$ define elementos del universo de Herbrand de Σ' tales que:

$$\varphi(t_1, t_2, \ldots, t_n) = \lambda_1(t_1, t_2, \ldots, t_n) \wedge \lambda_2(t_1, t_2, \ldots, t_n) \wedge \cdots \lambda_k(t_1, t_2, \ldots, t_n)$$

es consecuencia de Σ'. Recíprocamente para toda n-upla $\langle t_1, t_2, \ldots, t_n\rangle$ en el universo de Herbrand de Σ', tal que $\varphi(t_1, t_2, \ldots, t_n)$ sea consecuencia de Σ', existe una refutación input ordenada de raíz $\neg\varphi(x_1, x_2, \ldots, x_n)$ donde las sustituciones $(x_1|t'_1)$, $(x_2|t'_2)$,...,$(x_n|t'_n)$ cumplen que $\varphi(t_1, t_2, \ldots, t_n)$ es una instanciación de $\varphi(t'_1, t'_2, \ldots, t'_n)$.[2]

Esto signfica que para saber cuales son los objetos t para los cuales $q(t)$ es consecuencia de Σ' (o lo que es equivalente $q(t)$ es verdadero en todo modelo de Herbrand de Σ'), basta con explorar el árbol de deducciones input ordenadas por una estrategia "primero en profundidad", (o por una estrategia "primero en profundidad con retorno" si se está seguro que el árbol es finito) anotando por cada cláusula vacía que se encuentre el objeto t asociado.

Éste es el principio de los métodos de extracción de respuestas utilizados en numerosos sistemas de demostración automática (cfr. [8]) y en ciertos lenguajes de interrogación de bases de datos. En particular, el lenguaje Prolog está basado en este resultado. Puede decirse que calcula lo que pasa en el más pequeño modelo de Herbrand de Σ', el conjunto de cláusulas que definen el programa (cfr. [6]).

19. Ejercicios sobre Resolución

1. Decir si son unificables o no las fórmulas $r(x, f(a), y)$ y $r(f(y), x, b)$. En caso de respuesta afirmativa, dar un unificador principal. Igual pregunta para las fórmulas $r(x, f(a), y)$ y $r(f(y), x, z)$.

2. Dados los literales $p(g(f(x), u), f(a), g(z, f(y)))$ y $p(g(f(f(y)), g(v, a)), f(v), g(g(x, b), x))$, di cuales de las siguientes sustituciones es un unificador de máxima generalidad o principal.

   a) $(v|a)(u|g(v, a))(z|g(a, b))(x|a)$.
   b) $(v|a)(u|g(a, a))(z|g(f(b), b))(y|b)(x|f(b))$.
   c) $(v|a)(u|g(a, a))(z|g(x, b))(x|f(u))$.
   d) $(z|g(f(y), b))(v|a)(u|g(a, a))(x|f(y))$.

3. Decir si son unificables o no las fórmulas atómicas:

   - $\varphi_1 = p(x, y)$
   - $\varphi_2 = p(f(z), x)$
   - $\varphi_3 = p(w, f(x))$

   e igual pregunta para:

[2] Es decir es obtenida a partir de $\varphi(t'_1, t'_2, \ldots, t'_n)$ reemplazando ciertas variables por términos.

- $\varphi_1 = p(x, z)$
- $\varphi_2 = p(f(y), g(a))$
- $\varphi_3 = p(f(u), z)$

4. Comprobar si son unificables los siguientes conjuntos de literales. En caso afirmativo, calcular un unificador de máxima generalidad:

    a) $\{p(g(y), f(x, h(x), y)), p(x, f(g(z), w, z))\}$
    b) $\{p(x, g(f(a)), f(x)), p(f(a), g(y), y), p(y, z, y)\}$
    c) $\{p(x, f(g(y), b)), p(h(a, y), f(g(f_1(x)), b))\}$
    d) $\{p(x, z), p(g(f(z)), g(b)), p(g(f(w)), w)\}$

5. Mediante el "algoritmo de subsumisión", determinar

    a) si la cláusula $p(x, x)$ subsume a (es generalizada por) la cláusula $p(f(x), y) \vee p(y, f(x))$.
    b) si la cláusula $\neg p(x) \vee q(f(x), a)$ subsume a (es generalizada por) la cláusula $\neg p(h(y)) \vee q(f(h(y)), a) \vee \neg p(z)$.

6. Probar que el conjunto formado por las siguientes cláusulas es insatisfacible:

    - $t(x, g(a), z, f(u)) \vee \neg p(g(a)) \vee s(y, z) \vee \neg r(f(y))$
    - $q(x, f(y)) \vee \neg p(g(x))$
    - $\neg s(f(a), f(a)) \vee \neg r(f(f(a))) \vee \neg p(g(x))$
    - $r(f(x))$
    - $p(g(a)) \vee \neg q(g(b), f(x))$
    - $\neg p(x) \vee s(y, z) \vee \neg t(a, x, y, f(z)) \vee \neg q(g(b), z)$
    - $p(g(x))$

7. Encontrar el conjunto apropiado de cláusulas a través del cual estudiar si la fórmula $\exists x p(f(x))$ es consecuencia semántica del conjunto de hipótesis:

    - $\exists x \neg p\big(f(x)\big) \to \forall x q(x)$
    - $\exists y \forall z\big(r(z, y) \wedge r(z, a)\big) \to \forall x p(x)$
    - $\forall x \forall z\big(q(z) \to p(x) \vee q(z, a)\big)$

8. Dar una refutación lineal ordenada del conjunto de las siguientes cláusulas:

    - $\neg r\big(x, f(a), f(g(z))\big) \vee \neg q(a, x)$
    - $\neg p(z, y)$
    - $\neg q(a, x)$
    - $q\big(y, g(x)\big) \vee p(y, z)$
    - $q(x, z) \vee r\big(g(y), f(y), z\big) \vee p(x, z)$

9. Se quiere establecer a partir de los enunciados hipotéticos siguientes:

    - para todo crimen, hay alguien que lo ha cometido
    - sólo la gente deshonesta comete crímenes
    - no es detenida más que la gente deshonesta
    - la gente deshonesta detenida no comete crímenes
    - ocurren crímenes

el enunciado:

- hay gente deshonesta no detenida

como pretendida tesis. Utilizar para ello la resolución en lógica de primer orden.

10. A partir de las fórmulas:

    $\varphi 1)$ $\forall x \forall y \exists z p(x,y,z)$

    $\varphi 2)$ $\forall x \forall y \forall z \forall u \forall v \forall w\big((p(x,y,u) \wedge p(y,z,v)) \to (p(x,v,w) \leftrightarrow p(u,z,w))\big)$

    $\varphi 3)$ $\exists x\big(\forall y p(x,y,y) \wedge \forall z \exists u p(u,z,x)\big)$

    concluir como consecuencia semántica:

    $$\exists x\big(\forall y p(x,y,y) \wedge \forall z \exists u p(z,u,x)\big)$$

    utilizando la resolución. Enunciar un problema de la teoría de grupos que quede demostrado con lo que precede en este ejercicio.

11. Si es posible, extraer la cláusula vacía como resolvente del siguiente conjunto de cláusulas:

    $\sigma 1)$ $\neg p(x) \vee q(x, f(x))$

    $\sigma 2)$ $\neg p(x) \vee r(f(x))$

    $\sigma 3)$ $p(a)$

    $\sigma 4)$ $\neg q(x,y)$

12. Si en el sistema $\vdash_{rcv}$ la regla de resolución no impusiera el renombramiento previo, en una de las cláusulas padre, de las variables comunes ¿sería cierto el resultado de que todo conjunto insatisfacible de cláusulas es refutable? Sustentar la respuesta, según su naturaleza, con una demostración o un ejemplo.

13. Mostrar con un ejemplo que la conocida como *regla de disminución* es indispensable para la deducción semántica en lógica de primer orden utilizando la resolución. Para ello, encontrar un conjunto de cláusulas (de al menos dos) que siendo insatisfacible, sea imposible generar la cláusula vacía a partir de él con el mero y exclusivo uso de la regla de resolución.

14. Demostrar que el conjunto de fórmulas $\{\forall x\big(p(x) \to \forall y(q(x,y) \to r(y))\big),\ \neg\exists z \neg q(f(z), g(z)),\ p(f(a))\}$ implica semánticamente $r(g(a))$.

15. Demuestra, usando resolución, que la fórmula

    $$\forall x \forall y((r(x,y) \vee q(x)) \wedge \neg r(x, g(x)) \wedge \neg q(y))$$

    es insatisfacible.

16. Construir una derivación de la cláusula vacía a partir de las formas clausulares que se obtienen del conjunto de fórmulas $\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ donde:

    - $\varphi_1 = \forall x(\exists y p(x,y) \vee \neg q(x))$
    - $\varphi_2 = \forall x(r(x) \to (t(x) \vee q(x)))$
    - $\varphi_3 = \forall y(t(y) \to \exists z p(y,z))$
    - $\varphi_4 = \exists z(\forall y \neg p(z,y) \wedge r(z))$

17. Construir una derivación de la cláusula vacía a partir de las formas clausulares que se obtienen del conjunto de fórmulas $\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ donde:

    - $\varphi_1 = m(a)$

- $\varphi_2 = \forall x(m(x) \to m(f(x)) \lor m(g(x)))$
- $\varphi_3 = \forall x(q(x, f(x)) \land q(x, g(x)))$
- $\varphi_4 = \forall x(\neg m(x) \lor \forall y \neg q(y, x))$

18. Demostrar, usando la resolución, que la fórmula $\exists x \exists y(p(x, y) \land q(y))$ es consecuencia semántica de las fórmulas:

    - $q(a)$
    - $\forall x(q(x) \to q(g(x)) \lor q(f(x)))$
    - $\forall x(p(x, g(x)) \land p(x, f(x)))$

19. Demostrar, usando resolución lineal ordenada, que la fórmula

    $$\exists x(o(x) \land \forall y(t(y) \to \exists z p(y, z, x)))$$

    es consecuencia de las fórmulas:

    - $\forall x((t(x) \land \neg q(x)) \to \forall y \forall z(o(y) \to p(x, z, y)))$
    - $\forall x((t(x) \land q(x)) \to \forall y \forall z((o(y) \land s(y)) \to p(x, z, y)))$
    - $\exists x(o(x) \land r(x)) \land \exists y(o(y) \land s(y))$

20. Demostrar, usando resolución lineal ordenada, que la fórmula

    $$\exists x(r(x) \land t(x))$$

    es consecuencia de las fórmulas:

    - $\exists x(r(x) \land \forall y(p(y) \to q(x, y)))$
    - $\forall x(r(x) \to \exists y(s(x, y) \land p(y)))$
    - $\forall x(\exists y(s(x, y) \land q(x, y)) \to t(x))$

21. Demostrar, usando resolución lineal ordenada, que la fórmula

    $$\exists x \exists y(pl(x) \land cn(y) \land cp(x, y))$$

    es consecuencia de las fórmulas:

    - $\forall x(je(x) \land cs(x) \to \exists y(pp(y) \land cp(x, y)))$
    - $\forall x(je(x) \to pl(x)) \land \exists x(je(x) \land cs(x))$
    - $\forall x(pp(x) \to \neg cd(x) \land jg(x))$
    - $\forall x(jg(x) \to cn(x) \lor cd(x))$

22. Construir una OL-refutación del conjunto de cláusulas:

    - $r(a, f(c), f(b))$
    - $p(a)$
    - $r(x, x, f(x))$
    - $\neg r(x, y, z) \lor r(y, x, z)$
    - $\neg r(x, y, z) \lor q(x, z)$
    - $\neg p(x) \lor \neg r(y, z, u) \lor \neg q(x, u) \lor q(x, y) \lor q(x, z)$
    - $\neg q(a, b)$

23. Demostrar, usando OL-resolución, que la fórmula

$$\exists x(r(x) \lor \exists y(q(x,y) \land p(y)))$$

es consecuencia de las fórmulas:

- $\forall x((o(x) \land \exists y(q(y,x) \land s(y))) \to p(x))$
- $\forall x(o(x) \land v(x) \to s(x))$
- $\forall x(o(x) \land \neg v(x) \to r(x))$
- $\forall x(\exists y(q(y,x) \land o(y)) \to o(x))$
- $o(a) \land q(a,b)$

24. Demostrar, usando resolución lineal ordenada, que la fórmula

$$\neg \forall x o(x)$$

es consecuencia de las fórmulas:

- $\forall x(q(x) \land \neg t(x) \to s(x))$
- $\forall x(q(x) \land t(x) \to r(x) \lor p(x))$
- $\forall x((s(x) \lor r(x) \lor p(x)) \to \neg o(x)$
- $\exists x q(x)$

25. Demostrar, usando resolución lineal ordenada, que la fórmula

$$\exists x as(s)$$

es consecuencia de las fórmulas:

- $\forall x(md(x) \land \neg bl(x) \to ca(x))$
- $\forall x(un(x) \land ca(x) \to mx(g(x)) \lor fm(f(x)))$
- $\forall x(bl(x) \to dg(x))$
- $\forall x(mx(x) \to il(x))$
- $\forall x(fm(x) \to fa(x))$
- $\forall x(dg(x) \lor il(x) \lor fa(x) \to as(x))$
- $un(a) \lor md(a)$

26. Demostrar, usando resolución lineal ordenada, que la fórmula

$$\neg \exists x(r(x) \land s(x))$$

es consecuencia de las fórmulas:

- $\forall x(r(x) \land s(x) \to \exists y(q(y) \land p(x,y)))$
- $\forall x(q(x) \to t(x))$
- $\forall x(t(x) \to o(x))$
- $\forall x \forall y(r(x) \land o(y) \to \neg p(x,y))$

27. Demostrar utilizando resolución que la siguiente argumentación es correcta: “El profesor es feliz si a todos sus estudiantes les gusta la lógica; luego un profesor es feliz si no tiene estudiantes”.

Referencias

[1] Apt, K. and Van Emden, M. H. Contribution to the Theory of Logic Programming. *Journal of the Association for Computing Machinery*, 29(3):841–862, 1982. (Artículo fundamental concerniente a las estrategias input ordenadas, llamadas SLD en este documento).

[2] Boolos, G.S., Burgess, J.P., and Jeffrey, R.C. *Computability and Logic*. Cambridge University Press, fourth edition, 2003.

[3] Chang C. and Lee, R.C. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.

[4] Delahaye, J.P. *Outils Logiques pour l'Intelligence Artificielle*. Eyrolles, 1986.

[5] Henschen, L. and Wos, L. Unit Refutations and Horn Sets. *Journal of the Association for Computing Machinery*, 21(4):590–605, 1974. (Resultado importante sobre las dem. input).

[6] Lloyd, J.W. *Foundations of Logic Programming (Symbolic Computation: Artificial Intelligence)*. Springer-Verlag, 1987.

[7] Loveland, D.W. *Automated Theorem Proving: A Logical Basis*, volume 6 of *Fundamental Studies in Computer Science*. North-Holland Publishing Company, 1978.

[8] Nilsson, N.J. *Principios de Inteligencia Artificial*. Ediciones Dias de Santos, S.A., 1987.