Capítulo 2

Lenguajes de Primer Orden. Satisfacibilidad

2.1. 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 2.1.1. Representaremos por $\mathbf{L_A}$ al lenguaje para el que C $= \{c\}$, F $= \{f^1, g^2, h^2\}$ y R $= \{r_1^2\}$ (r_1^2 está destinada a "encarnar" la igualdad). En este caso tenemos, por tanto: C $= \{a\}$, F $= \{f, g, h\}$ y R $= \{r\}$. Para **L** 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 Term(**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 Atom(**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 **L** es representado por Form(**L**).

Ejemplo 2.1.2. Para el lenguaje **L** del ejemplo 2.1.1 mostramos a modo de ejemplo los siguientes términos: a, 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 sobre fórmulas por *inducción*.

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 $v_{i_1}, \ldots, v_{i_k} \in \mathrm{V}$, la expresión $\varphi(v_{i_1}, \ldots, v_{i_k})$ es una representación de φ e indicará que algunas de las variables $v_{i_1}, \ldots, v_{i_k}$ 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 Sent(**L**).

Ejemplo 2.1.3. 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 2.1.4. 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 2.1.5. 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 2.1.1. 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. $\forall y\exists x s(x,y,g(x,y)) \vee \neg\forall x r(y,f(x))$

2.2. 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 2.2.1. 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 $s : \mathrm{V} \longrightarrow A$. 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 2.2.1. Sea $\langle \mathbf{A}, s\rangle$ una interpretación del lenguaje $\mathbf{L}$, definimos $I_{\mathbf{A}}^s$ 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, $I_{\mathbf{A}}^s(r_k^n(t_1 \cdots t_n)) = 1$ 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) $I_{\mathbf{A}}^s(\neg\alpha) = I_{\mathbf{A}}^s(\alpha) + 1$

   b) $I_{\mathbf{A}}^s(\alpha \to \beta) = I_{\mathbf{A}}^s(\alpha) I_{\mathbf{A}}^s(\beta) + I_{\mathbf{A}}^s(\alpha) + 1$

   c) $I_{\mathbf{A}}^s(\alpha \vee \beta) = I_{\mathbf{A}}^s(\alpha) I_{\mathbf{A}}^s(\beta) + I_{\mathbf{A}}^s(\alpha) + I_{\mathbf{A}}^s(\beta)$

   d) $I_{\mathbf{A}}^s(\alpha \wedge \beta) = I_{\mathbf{A}}^s(\alpha) I_{\mathbf{A}}^s(\beta)$

   e) $I_{\mathbf{A}}^s(\alpha \leftrightarrow \beta) = I_{\mathbf{A}}^s(\alpha) + I_{\mathbf{A}}^s(\beta) + 1$
3. Para fórmula α y símbolo de variable x del lenguaje:

   a) $I_{\mathbf{A}}^s(\forall x\alpha) = 1$ sii, por definición, para todo $d \in A$, $I_{\mathbf{A}}^{s(x|d)}(\alpha) = 1$

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

   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 $\cdot$ que aparecen en esta definición son las mismas presentadas y usadas en la Definición 1.2.1.

Definición 2.2.2. Sea $\mathbf{L}$ un lenguaje de primer orden y $\Gamma \cup \{\alpha\} \subseteq \mathrm{Form}(\mathbf{L})$. Γ *implica semánticamente* a φ, en símbolos $\Gamma \models \varphi$, si para toda interpretación $\langle \mathbf{A}, s\rangle$ de $\mathbf{L}$ se tiene $I_{\mathbf{A}}^s(\varphi) = 1$ siempre que para toda $\gamma \in \Gamma$ ocurra $I_{\mathbf{A}}^s(\gamma) = 1$. En el caso de que $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$, escribimos $\gamma_1, \ldots, \gamma_n \models \alpha$ en lugar de $\{\gamma_1, \ldots, \gamma_n\} \models \alpha$ y si $\Gamma = \emptyset$ escribimos simplemente $\models \alpha$ en lugar de $\emptyset \models \alpha$. Las fórmulas ψ y φ son *lógicamente equivalentes* si $\psi \models \varphi$ y $\varphi \models \psi$ (o dicho de otra forma, si $\models \psi \leftrightarrow \varphi$).

Definición 2.2.3. 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, $\models \varphi$).

Teorema 2.2.1. *Sea* $\mathbf{L}$ *un lenguaje y* $\Gamma \cup \{\psi, \varphi\}$ *un conjunto de fórmulas fórmulas de* $\mathbf{L}$. *Son equivalentes las siguientes afirmaciones:*

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

Corolario 2.2.2. *Sea* $\mathbf{L}$ *un lenguaje y* $\gamma_1, \ldots, \gamma_n, \varphi$ *fórmulas de* $\mathbf{L}$. *Son equivalentes las siguientes afirmaciones:*

1. $\gamma_1, \ldots, \gamma_n \models \varphi$
2. $\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n \models \varphi$
3. $\models \gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n \to \varphi$

Corolario 2.2.3. *Para cualesquiera fórmulas* $\gamma_1, \ldots, \gamma_n, \varphi$ *(*$2 \leq n$*) son equivalentes las siguientes afirmaciones:*

1. $\gamma_1, \ldots, \gamma_n \models \varphi$
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 2.2.4 (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,* $s_1 \restriction W = s_2 \restriction W$*). Entonces*

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

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 $\varphi = r^n_k(t_1 \cdots t_n)$. 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 $\langle \overline{s}_1(t_1), \ldots, \overline{s}_1(t_n)\rangle \in (r^n_k)^{\mathbf{A}}$ si, y sólo si, $\langle \overline{s}_2(t_1), \ldots, \overline{s}_2(t_n)\rangle \in (r^n_k)^{\mathbf{A}}$. Así pues, $I^{s_1}_{\mathbf{A}}(\varphi) = I^{s_2}_{\mathbf{A}}(\varphi)$.

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 2.2.1 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 $a \in A$, $I^{s_1(x|a)}_{\mathbf{A}}(\alpha) = I^{s_2(x|a)}_{\mathbf{A}}(\alpha)$. Es decir, $I^{s_1}_{\mathbf{A}}(\forall x\alpha) = I^{s_2}_{\mathbf{A}}(\forall x\alpha)$ (resp. $I^{s_1}_{\mathbf{A}}(\exists x\alpha) = I^{s_2}_{\mathbf{A}}(\exists x\alpha)$). ■

Corolario 2.2.5. *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* $\mathbf{A}$, $I^s_{\mathbf{A}}(\sigma) = 1$.
2. *Para toda asignación* s *de* V *en* $\mathbf{A}$, $I^s_{\mathbf{A}}(\sigma) = 0$.

Observación 2.2.1. Observar que no pueden ocurrir simultáneamente las opciones 1) y 2) del corolario 2.2.5. La razón para ello es que A, por definición, es no vacío en cualquier estructura algebraica $\mathbf{A}$.

Definición 2.2.4. 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, $I^s_{\mathbf{A}}(\varphi) = 1$. $\mathbf{A}$ es un modelo de un conjunto de fórmulas Γ si es modelo de cada una de sus fórmulas.

Observación 2.2.2.

1. Toda estructura es modelo de $\emptyset$

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 $I^s_{\mathbf{A}}(\varphi) = 1$. (cfr. Teorema 2.2.5.)

Corolario 2.2.6. *Sea* $\mathbf{L}$ *un lenguaje y* $\Gamma \cup \{\tau\}$ *un conjunto de sentencias de* $\mathbf{L}$. $\Gamma \models \tau$ *si, y sólo si, todo modelo de* Γ *es un modelo de* τ.

Ejemplo 2.2.2. 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 $s(v_i) = i - 1$. $\overline{s}(f(f(v_3))) = (2')' = 4$, $\overline{s}(f(f(c))) = 2$, $\models_{\mathbf{A}} r(c, f(v_1))[s]$, 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 $\models_{\mathbf{A}} \forall v_1 r(c, v_1)$. Sin embargo, $\not\models_{\mathbf{A}} \forall v_1 r(v_2, v_1)$, porque existe un natural m tal que $\not\models_{\mathbf{A}} r(v_2, v_1)[s(v_1|m)]$, 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 2.2.1.

1. $\forall v_1 q(v_1) \models q(v_2)$

2. $q(v_1) \not\models \forall v_1 q(v_1)$

3. $\forall v_1 q(v_1) \models \exists v_2 q(v_2)$

4. $\exists x \forall y p(x, y) \models \forall y \exists x p(x, y)$

5. $\forall y \exists x p(x, y) \not\models \exists x \forall y p(x, y)$

6. $\models \exists x(q(x) \to \forall x q(x))$

Ejercicio 2.2.2. Encontrar la clase de los modelos de las siguientes sentencias:

1. $\forall x \forall y (x \approx y)$

2. $\forall x \forall y p(x, y)$

3. $\forall x \forall y \neg p(x, y)$

4. $\forall x \exists y p(x, y)$

2.3. 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 2.3.1. *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 2.3.2. *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 2.3.1. Sea $\mathbf{L}$ un lenguaje, x un símbolo de variable, t y u términos y α una fórmula. Definimos s^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 α^x_t como sigue:

1. Si α es la fórmula atómica $r(t_1, \ldots, t_n)$, $\alpha^x_t = r((t_1)^x_t, \ldots, (t_n)^x_t)$.
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 2.3.1. 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. $(q(x) \to \forall x p(x))^x_y = (q(y) \to \forall x p(x))$
3. Si α es la fórmula $\neg\forall y x \approx y$, 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 2.3.3. *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* $a \in A$, $I_{\mathbf{A}}^{s(x|a)}(\varphi) = I_{\mathbf{A}}^{s(y|a)}(\varphi_y^x)$.

Teorema 2.3.4. *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* $\forall y(\varphi_y^x)$.

Lema 2.3.5. *Sea* $\mathbf{L}$ *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 2.3.6. *Sea* $\mathbf{L}$ *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 2.3.7. *Sea* $\mathbf{L}$ *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 2.3.2. Sea $\mathbf{L}$ un lenguaje y φ una fórmula de $\mathbf{L}$. φ está en *forma prenexa* si se expresa como

$$Q_1x_1 \cdots Q_nx_n\beta \tag{2.1}$$

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 2.1, β se denomina matriz de la fórmula φ y a la expresión $Q_1x_1 \cdots Q_nx_n$ 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 2.3.1. Cuando en la definición 2.3.2 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 $\mathbf{L}$.

Teorema 2.3.8. *Sea* $\mathbf{L}$ *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 2.3.2. Observar que la fórmula en forma prenexa equivalente cuya existencia garantiza el teorema 2.3.8 puede ser encontrada en forma normal prenexa.

Teorema 2.3.9. *Sea* $\mathbf{L}$ *un lenguaje,* $\alpha, \beta \in \mathrm{Form}(\mathbf{L})$ *y* $x \in \mathrm{V}$. *Entonces:*

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

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

- Usar que

$$\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi) \tag{2.2}$$
$$\varphi \to \psi \equiv \neg\varphi \vee \psi \tag{2.3}$$

- Usar repetidamente que:

$$\neg\neg\varphi \equiv \varphi \tag{2.4}$$

  las *leyes de De Morgan* que establecen que:

$$\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi \tag{2.5}$$
$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi \tag{2.6}$$

  así como que:

$$\neg\forall x\varphi \equiv \exists x\neg\varphi \tag{2.7}$$
$$\neg\exists x\varphi \equiv \forall x\neg\varphi \tag{2.8}$$

  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:

$$\forall x\varphi \equiv \forall y(\varphi_y^x) \tag{2.9}$$
$$\exists x\varphi \equiv \exists y(\varphi_y^x) \tag{2.10}$$

- Usar que:

$$\forall x\varphi \wedge \forall x\psi \equiv \forall x(\varphi \wedge \psi) \tag{2.11}$$
$$\exists x\varphi \vee \exists x\psi \equiv \exists x(\varphi \vee \psi), \tag{2.12}$$

y que si x no ocurre libremente en ψ

$$\forall x\psi \equiv \psi \tag{2.13}$$
$$\exists x\psi \equiv \psi \tag{2.14}$$
$$\forall x\varphi \vee \psi \equiv \forall x(\varphi \vee \psi) \tag{2.15}$$
$$\forall x\varphi \wedge \psi \equiv \forall x(\varphi \wedge \psi) \tag{2.16}$$
$$\exists x\varphi \vee \psi \equiv \exists x(\varphi \vee \psi) \tag{2.17}$$
$$\exists x\varphi \wedge \psi \equiv \exists x(\varphi \wedge \psi) \tag{2.18}$$

Ejercicio 2.3.1. Encontrar una fórmula en forma prenexa cuya matriz esté expresada como conjunción de disyunciones de literales y que sea lógicamente equivalente a las siguientes:

1. $\big(\forall x\exists yp(x,y) \wedge \big(\exists yq(y) \rightarrow q(a)\big)\big) \vee \forall y\big(\exists y\forall xp(x,y) \vee \exists zp(y,a)\big)$
2. $\big(\forall x(r(x) \vee \exists y\forall xp(x,y)) \vee \exists xq(x,y)\big) \wedge \big(\exists zr(z) \rightarrow \forall x(r(x) \wedge \forall xp(x,a))\big)$
3. $\forall xp(x,y) \rightarrow \big(\forall yp(y,x) \rightarrow \forall x(q(x) \wedge \exists y\forall zr(a,y,z)\big)$
4. $(\forall xp(a,x) \vee \forall xp(x,a)) \rightarrow \big(\forall zp(x,z) \wedge \forall w\forall y(p(a,y) \rightarrow \exists zq(z))\big)$
5. $\forall x\forall z\big((\forall zp(x,z) \wedge \forall xp(x,z)) \rightarrow \forall x(\exists yp(x,y) \vee \forall xq(x))\big)$

2.4. Forma Normal de Skolem

Definición 2.4.1. Sea φ una fórmula del lenguaje de primer orden $\mathbf{L}$ en forma normal prenexa, esto es, se escribe como:

$$Q_1x_1 \cdots Q_nx_n\beta$$

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 2.4.1. Si en la operación descrita en la Defininición 2.4.1 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 2.4.1.

- si φ fuese $\exists xp(x, f(x))$, φ^s sería $p(a, f(a))$
- si φ fuese $\forall x\exists yp(x, f(y))$, φ^s sería $p(x, f(f_1(x)))$
- 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 2.4.2. 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 2.4.1. *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 [4] y mucho mejor en [2], pág. 249. ■

El problema general de la *demostración automática de teoremas* es el de saber si $\Gamma \models \varphi$, 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 1.2.1, sabemos que el problema se reduce a saber si $\Gamma \cup \{\neg\varphi\}$ es insatisfacible o no. Lo que aporta el Teorema 2.4.1 es que equivalentemente basta con estudiar la insatisfacibilidad de $\Gamma^s \cup \{(\neg\varphi)^s\}$

Definición 2.4.2. 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 2.4.2.

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 2.4.3. 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 2.4.4. 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 2.4.3. Nos preguntamos si es cierto o no que $\forall x(p(x) \rightarrow \exists y(r(y) \wedge q(x, y))), \exists x p(x) \models \exists x \exists y q(x, y)$. Para intentar dar una respuesta consideramos las sentencias:

- en cuanto a la fórmula $\forall x(p(x) \rightarrow \exists y(r(y) \wedge q(x, y)))$:
  - $\forall x(p(x) \rightarrow \exists y(r(y) \wedge q(x, y)))$
  - $\forall x(\neg p(x) \vee \exists y(r(y) \wedge q(x, y)))$
  - $\forall x \exists y(\neg p(x) \vee (r(y) \wedge q(x, y)))$
  - $\forall x \exists y((\neg p(x) \vee r(y)) \wedge (\neg p(x) \vee q(x, y)))$
- 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

$$\{\forall x \exists y((\neg p(x) \vee r(y)) \wedge (\neg p(x) \vee q(x, y))), \exists x p(x), \forall x \forall y \neg q(x, y)\}. \tag{2.19}$$

Por el Teorema 2.4.1, la insatisfacibilidad de 2.19 es la de

$$\{\forall x\big((\neg p(x) \vee r(f(x))) \wedge (\neg p(x) \vee q(x, f(x)))\big), p(a), \forall x \forall y \neg q(x, y)\} \tag{2.20}$$

o más esquemáticamente la de

$$\{\neg p(x) \vee r(f(x)), \neg p(x) \vee q(x, f(x)), p(a), \neg q(x, y)\} \tag{2.21}$$

Si el conjunto 2.20, entendido como sugiere 2.21, no tiene modelos, ello significa que $\forall x(p(x) \rightarrow \exists y(r(y) \wedge q(x, y))), \exists x p(x) \models \exists x \exists y q(x, y)$. Es preciso que el conjunto 2.21 sea lo más simple posible, para lo cual usaremos de forma óptima todas las herramientas de que disponemos.

2.5. 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 2.4.3 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 2.5.1 (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 térmimos 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 2.5.1. 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 2.5.1. 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 2.5.2 (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 2.5.2. 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 2.5.3. 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 2.5.4. 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 2.5.3. En nuestro ejemplo

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

Teorema 2.5.1 (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 2.5.2. El Teorema 2.5.1 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 2.5.4. Siguiendo con nuestro ejemplo, el subconjunto $\{p(a), \neg p(a) \lor q(a, f(a)), \neg q(a, f(a))\}$ de S_Σ —que es un subconjunto de cláusulas proposicionales— es insatisfacible. En consecuencia, la pregunta formulada en el Ejemplo 2.4.3 tiene respuesta afirmativa, es decir,

$$\forall x(p(x) \to \exists y(r(y) \land q(x,y))), \exists x p(x) \models \exists x \exists y q(x,y)$$

Observación 2.5.3. 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

[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.

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 [5], [4] y [8].

Ejemplo 2.5.5. 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:

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

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:

$$\begin{aligned}
&\neg p(x) \vee p(y)\\
&p(x) \vee q(x)\\
&\neg p(a)\\
&\neg p(b)
\end{aligned}$$

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 2.4.1, el Corolario 2.2.2 y el Teorema de Herbrand deducimos que la pregunta inicial tiene respuesta afirmativa.

Ejemplo 2.5.6. 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 2.5.7. 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

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