Tema 5

Lógica de primer orden con igualdad.

5.1 Sintaxis del lenguaje

5.1.1 Introducción

Con la lógica de proposiciones, la lógica formal era capaz de decidir acerca de la validez de cualquier razonamiento en el que se derivara una proposición sin analizar a partir de otra u otras proposiciones que tampoco se analizaban. En otras palabras: la lógica formal, al nivel de la lógica de proposiciones, sólo puede analizar formalmente de manera acabada aquellos razonamientos en cuya validez no desempeña ningún papel la estructura interna de las proposiciones que los componen.

Y, sin embargo, hay razonamientos que, siendo formalmente válidos, no lo son simplemente en virtud de las puras conexiones externas entre los enunciados a partir de los cuales están construidos. Su forma lógica no puede exhibirse cumplidamente tan sólo mediante letras de enunciado y conectivas. Es preciso ir más allá: penetrar en la estructura interna de los enunciados, en busca de elementos relevantes para la validez del razonamiento en cuestión.

Un ejemplo de razonamiento cuya validez escapa a la lógica de proposiciones sería:

Ningún fósil puede estar traspasado de amor.
Una ostra puede estar traspasada de amor.
___
Las ostras no son fósiles.

Desde el punto de vista de la lógica proposicional la forma lógica de este razonamiento es:

$$\begin{array}{l} \alpha \\ \beta \\ \hline \gamma \end{array}$$

Con lo que se daría el caso de que un razonamiento que, a la luz de la intuición es válido, no lo sería a la luz de la Lógica. Si la Lógica se redujera a la lógica proposicional, mal cumpliría su función de análisis formal de la validez de los razonamientos.

Debemos por tanto ampliar el análisis lógico para dar cuenta de estos razonamientos en los que interviene la estructura interna de los enunciados. La lógica de predicados es una extensión de la lógica proposicional que se obtiene analizando lógicamente los enunciados. ¿Qué descubre la lógica dentro de los enunciados? ¿Qué hay en ellos que le interese? A estas preguntas trataremos de responder en las secciones que siguen.

Designadores y functores.

La cadena sonora que sale de nuestras bocas al hablar puede ser segmentada de diversas maneras. De la mayoría de los segmentos no tendría sentido preguntarse por el objeto o individuo al que se refieren o designan. No designan objeto alguno. A los segmentos de la cadena sonora que se refieren a algún objeto o individuo los llamamos *designadores*.

Si en vez de analizar la cadena sonora analizamos textos escritos, nos encontraremos en una situación parecida. Podremos segmentar los textos (o sucesiones finitas de signos gráficos del alfabeto de que se trate, ampliado para abarcar los signos de puntuación y el espacio de separación) de muchas maneras. Algunos segmentos de texto designarán quizás algo o alguien, y les llamaremos *designadores*. La mayoría no designan nada, no se refieren a nada.

Hay muchos tipos de designadores. Uno de los más sencillos esta constituido por los nombres. Todos conocemos ejemplos de nombres. Por ejemplo, "1", "2", "3", "4", "5", "6"... son nombres de números. "París", "Roma", "Barcelona", "Reus", "Sao Paulo", "Yokohama"... son nombres de ciudades. "Pablo Picasso", "André Gide", "José María de Porcioles", etc., son nombres de personas. "Marte", "Tierra", "Venus"... son nombres de planetas. "RENFE", "UNESCO", "ONU", "NATO"... son nombres de empresas u organizaciones.

Los nombres son —limitando ahora nuestra atención al lenguaje escrito— sucesiones de signos gráficos que designan algo —un número, una ciudad, una persona, un planeta, una empresa...—. En esto se comportan como el resto de los designadores, de los que se diferencian por ser generalmante más cortos, más sencillos, más unívocos, más independientes del contexto.

Vamos a ir introduciendo un simbolismo sencillo para formalizar las expresiones lingüísticas que nos interesen. Así, en vez de los nombres del lenguaje ordinario, nosotros utilizaremos las primeras letras minúsculas del alfabeto latino: a, b, c, d,... con subíndices cuando sea necesario.

Consideremos el texto: "Sisebuto ama a Purita, que es la madre de Tecla". Podemos simbolizar a "Sisebuto" por a, "Purita" por b, y "Tecla" por c, con lo que obtenemos: "a ama a b, que es la madre de c".

Los nombres son designadores simples, en el sentido de que ninguna parte propia de ellos es a su vez un designador. Pero no todos los designadores son así. Con frecuencia nos encontramos con designadores compuestos, designadores que pueden segmentarse en varias partes, algunas de las cuales son a su vez designadores.

"El río que atraviesa la capital de Francia" es un designador, una expresión lingüística que se refiere a un objeto o individuo: el río Sena. "Sena" es su nombre, pero no la única expresión que lo designa. "La capital de Francia" es una parte propia del designador citado y, a su vez, un designador, e incluso un designador compuesto también, pues una de sus partes, "Francia", es ella misma un designador; un designador simple en este caso, es decir, un nombre.

Hay algunas expresiones lingüísticas que, seguidas de un número determinado de designadores, forman a su vez un designador. Estas expresiones se llaman *functores*.

Un functor que, seguido de un designador de cierto tipo, forma un designador, se llama un functor monádico o monario. Así, hablando de personas, "la madre de ..." es un functor monádico. Junto con los designadores personales

1. "Florita"
2. "Agapito"

forma los designadores

1. "la madre de Florita"
2. "la madre de Agapito"

Hablando de números naturales, "$\ldots^2$" o "el siguiente de ..." son functores monádicos. Junto con los designadores

1. "3"
2. "4"

forman los designadores

1. "3^2" y "el siguiente de 3"
2. "4^2" y "el siguiente de 4"

Un functor que necesita dos designadores de un cierto tipo para formar un nuevo designador se llama functor diádico o binario. Si seguimos hablando de números naturales, "$\ldots + \ldots$" es un funtor diádico o binario. Junto con los designadores "3" y "4^2" forman el designador "$3 + 4^2$"

Un functor que necesita tres designadores de un cierto tipo para formar un nuevo designador se llama functor triádico o ternario. En general, un functor que necesita de n designadores para formar un nuevo designador se llama un functor n-ádico o n-ario.

En nuestra simbolización, en vez de los functores del lenguaje ordinario, utilizaremos las siguientes letras minúsculas del alfabeto latino: f, g, h, si es necesario con subíndices.

Relatores.

Habíamos visto que hay expresiones lingüísticas que, junto con un número determinado de designadores, forman un nuevo designador. Las habíamos llamado functores. Del mismo modo podemos observar que hay expresiones lingüísticas que, junto con un número determinado de designadores, forman un enunciado. Las llamaremos *relatores*.

Un relator que, seguido de un designador de un cierto tipo, forma un enunciado se llama un relator monádico o monario. Así, hablando de personas,

1. "… es bueno",
2. "… está enfermo",
3. "… ronca terriblemente",
4. "… vive en una casa de campo en las afueras de Amsterdam"

son relatores monádicos. Junto con designadores tales como

1. "Juan Peláez",
2. "el rey de Thailandia",
3. "el padre de Juan Peláez"
4. "el alcalde de Amsterdam",

forman enunciados tales como:

1. "Juan Peláez es bueno",
2. "el rey de Thailandia está enfermo",
3. "el padre de Juan Peláez ronca terriblemente",
4. "el alcalde de Amsterdam vive en una casa de campo en las afueras de Amsterdam"

Un relator que necesita dos designadores para formar un enunciado se llama un relator diádico o binario. Así, hablando de personas,

1. "… ama a … ",
2. "… es mucho más alto que … ",

son relatores diádicos. Junto con designadores tales como

1. "Juan Peláez",
2. "la hija mayor del alcalde de Amsterdam",
3. "Julio Quebrantahuesos"
4. "el rey del Nepal",

forman enunciados como

1. "Juan Peláez ama a la hija mayor del alcalde de Amsterdam",
2. "Julio Quebrantahuesos es mucho más alto que el rey del Nepal".

Un relator que necesita tres designadores para formar un enunciado se llama un relator triádico o ternario. Así, hablando de ciudades,

1. "... está situada entre ... y ... "

es un relator triádico. Junto con designadores tales como

1. "Zaragoza",
2. "Madrid",
3. "Barcelona",

forma enunciados tales como

1. "Zaragoza está situada entre Madrid y Barcelona"

En general un relator que necesita n designadores para formar un enunciado se llama un relator n-ádico o n-ario.

En nuestro simbolismo, emplearemos las letras mayúsculas del alfabeto latino P, Q, R, S para representar los relatores del lenguaje ordinario. Si es necesario, emplearemos subíndices de diferenciación.

Consideremos el texto:

"Juan ama a su madre, pero no aguanta a doña Leovigilda".

Podemos simbolizar el nombre "Juan" por a, el nombre "doña Leovigilda" por b, el functor "la madre de ..." por f, el relator "... ama a ... " por P y el relator "... aguanta a ... " por Q. Escribiendo el relator delante de los designadores con los que forma un enunciado, obtenemos:

$$P(a, f(a)), \text{ pero no } Q(a, b).$$

Variables.

Los matemáticos utilizan con frecuencia variables, sobre todo cuando quieren decir algo bastante general, como que la ecuación

$$x + y = y + x$$

siempre resulta satisfecha, cualesquiera que sean los números reales que pongamos en lugar de las variables.

En el lenguaje ordinario, los pronombres juegan con frecuencia el papel de variables. "Él ha sido el asesino". Es como decir: "x ha sido el asesino". "Lo he visto con mis propios ojos". Es como decir: "He visto x con mis propios ojos".

En realidad, a la hora de analizar textos del lenguaje ordinario y simbolizarlos adecuadamente, nos daremos cuenta de que las variables constituyen un valioso recurso de simbolización. Como variables utilizaremos las últimas letras minúsculas del alfabeto latino: u, v, w, x, y, z. Si es necesario utilizaremos subíndices de diferenciación.

Preenunciados y predesignadores.

Si en un enunciado sustituimos un designador por una variable (o varios designadores por otras tantas variables), el resultado es lo que llamamos un *preenunciado* o un *enunciado abierto.*

Así, sustituyendo el designador "Juan" por la variable x en el enunciado "Juan ama a su prima", obtenemos el preenunciado "x ama a su prima". Del mismo modo, sustituyendo el designador "la madre de Luis" por y en el enunciado "la madre de Luis se pasa el día cosiendo", obtenemos el enunciado abierto "y se pasa el día cosiendo". Sustituyendo el designador "5" por x y el designador "7" por y en el enunciado "$5+7 \geq 10$", obtenemos el enunciado abierto "$x + y \geq 10$".

Cuantificadores.

A veces nos encontramos con expresiones lingüísticas que nos sirven para decir algo de todos los objetos de una clase determinada. Por ejemplo, la expresión “todos los” en “todos los chinos aman a Mao”, o la expresión “cada” en “cada uno tiene sus gustos”, o la expresión “el” en “el hombre es un mamífero”. Otras expresiones nos sirven para decir algo de algunos objetos de una clase determinada, para afirmar que en esa clase hay al menos un objeto que cumple lo que se dice. Por ejemplo, la expresión “unos” en “unos tipos sospechosos me seguían”, o la expresión “algunos” en “algunos chinos aman a Liu Chao-chi”, o la expresión “hay” en “hay personas que pesan más de 120 kg.”. A todas estas expresiones las llamaremos cuantificadores. A las primeras (“todo”, “cada”, “el” ...), cuantificadores universales o generales, a las segundas (“algún”, “unos”, “hay”, ...), cuantificadores existenciales o particulares.

Ejemplo 5.1
$\forall x \forall y\ A(x,y)$ *Todos se aman entre sí (un mundo ideal)*
$\forall x \exists y\ A(x,y)$ *Para cada persona existe otra a la que ama*
$\exists x \forall y\ A(x,y)$ *Existe una persona que ama a todos (¿un santo?)*
$\exists x \exists y\ A(x,y)$ *Existe por lo menos una persona que quiere a otra (no todo está perdido, a no ser que sea un narcisista)*
$\forall y \forall x\ A(x,y)$ *Todos se aman entre sí (un mundo ideal)*
$\forall y \exists x\ A(x,y)$ *Cada persona tiene alguien que lo ama*
$\exists y \forall x\ A(x,y)$ *Existe una persona al que todo el mundo quiere*
$\exists y \exists x\ A(x,y)$ *Existe por lo menos una persona que quiere a otra (no todo está perdido, a no ser que sea un narcisista)*

5.1.2 Descripción de un lenguaje de primer orden

Definición 5.1 (lenguaje de primer orden) *Un* lenguaje de primer orden *es* $L = (\mathcal{C}, \mathcal{F}, \mathcal{R}, \rho)$ *donde* $\mathcal{C}$*,* $\mathcal{F}$ *y* $\mathcal{R}$ *son tres conjuntos disjuntos y* ρ *es una aplicación de* $\mathcal{F} \cup \mathcal{R}$ *en* $\mathbb{N}^+$*. Un lenguaje* L *es* algebraico *si* $\mathcal{R} = \emptyset$*. Un lenguaje* L *es relacional puro si* $\mathcal{C} \cup \mathcal{F} = \emptyset$*.*

$\mathcal{C}$ es el conjunto de los símbolos de constante y sus elementos los notaremos con la letra genérica c.
ρ es la función ariedad.
$\mathcal{F}$ es el conjunto de los símbolos de función y sus elementos los notaremos con la letra f y el conjunto de los de ariedad n se nota $\mathcal{F}_n$
$\mathcal{R}$ es el conjunto de los símbolos de relación y sus elementos los notaremos con la letra R y el conjunto de los de ariedad n se nota $\mathcal{R}_n$.
Cuando la naturaleza y ariedad de los símbolos es conocida se suele identificar L con $\mathcal{C} \cup \mathcal{F} \cup \mathcal{R}$ que se llaman símbolos específicos del lenguaje L.

Además de los símbolos de L utilizaremos en la sintaxis de primer orden los siguientes símbolos, que son comunes a todos los lenguajes de primer orden:

$\mathcal{V} = \{v_i :\ i \in \mathbb{N}^+\}$ conjunto de las variables.

$\mathbb{C}\text{ons} = \{\neg, \vee, \wedge, \rightarrow, \leftrightarrow, \approx\} \cup \{\forall x :\ x \in \mathcal{V}\} \cup \{\exists x :\ x \in \mathcal{V}\}$ constantes lógicas.

Como símbolos de variable utilizaremos las últimas letras (minúsculas) del abecedario con o sin subíndices. Las constantes lógicas están constituidas por las conectivas de la lógica de enunciados, la relación de igualdad (cuando se incluye este símbolo en los lenguajes se suelen llamar de primer orden con igualdad) y los cuantificadores (obsérvese que hay dos para cada variable uno universal o general y otro particular o existencial).

- Algunos autores en la descripción de los lenguajes incluyen como símbolos auxiliares del lenguaje los paréntesis y la coma, pues son símbolos que se emplean al escribir las fórmulas del lenguaje.
- De acuerdo con la definición, para dar un lenguaje tendremos que especificar cuáles son los símbolos de constante, de función y de relación y sus ariedades. Es decir, los símbolos específicos del lenguaje.
- A veces, si queremos dar un lenguaje de primer orden para hablar en un determinado contexto, podemos extender la notación y emplear símbolos que nos sugieran lo que van a representar (por ejemplo, si estamos hablando de personas, un símbolo de constante puede ser *purita*, un símbolo de función *el_padre_de* y un símbolo de predicado *Es_hipertimésico*.

- Un símbolo de función o de predicado cuya ariedad sea 1 diremos que es unario o monario. Si tiene ariedad 2 diremos que es binario. Si tiene ariedad 3, diremos que es ternario. Cuando su ariedad sea n diremos que es n-ario.
- Los símbolos de constante pueden ser considerados como símbolos de función 0-arios (también llamados nularios).

A partir de estas observaciones, para describir un lenguaje de primer orden diremos algo así como:

Sea L el lenguaje de primer orden con:

$\mathcal{C} = \{a, b, c\}$;

$\mathcal{F} = \{f, g, h\}$, f y h son unarios y g es binario, o bien, $\mathcal{F}_1 = \{f, h\}$ y $\mathcal{F}_2 = \{g\}$

$\mathcal{R} = \{P, Q, R, S\}$, R es monario, P y S son binarios y Q es ternario, o bien, $\mathcal{R}_1 = \{R\}$, $\mathcal{R}_2 = \{P, S\}$ y $\mathcal{R}_3 = \{Q\}$.

Hasta ahora, únicamente tenemos los símbolos que vamos a usar para escribir en nuestro lenguaje. Una fórmula en nuestro lenguaje será una sucesión finita de estos símbolos junto con las conectivas, cuantificadores, paréntesis y comas. Pero no toda sucesión de tales símbolos será una fórmula, sino que seguirán una serie de reglas.

Vamos a dar ahora las reglas para, a partir de estos símbolos, formar los distintos elementos del lenguaje. Comenzamos definiendo los términos, que tal y como dijimos en la introducción van a ser los designadores o predesignadores de los objetos de los que hablaremos en nuestro lenguaje.

Términos de un lenguaje.

Definición 5.2 (términos) *Sea $L = (\mathcal{C}, \mathcal{F}, \mathcal{R}, \rho)$ un lenguaje de primer orden. Definimos recursivamente el conjunto de términos del lenguaje L, que notaremos $Ter(L)$, como sigue:*

1. *Todo símbolo de constante es un término. Es decir, $\mathcal{C} \subseteq Ter(L)$.*
2. *Toda variable es un término. Es decir, $\mathcal{V} \subseteq Ter(L)$.*
3. *Si f es un símbolo de función n-aria, y $t_1, t_2, \ldots, t_n$ son términos, entonces $f(t_1, t_2, \ldots, t_n)$ es un término. Si $f \in \mathcal{F}_n$ y $t_1, t_2, \ldots, t_n \in Ter(L)$, entonces $f(t_1, t_2, \ldots, t_n) \in Ter(L)$.*
4. *Todo sucesión de símbolos que no se haya construido siguiendo las reglas anteriores no es un término.*

Las reglas primera y segunda nos dicen que los símbolos de constante y de variable son términos. La regla tercera nos dice cómo construir nuevos términos a partir de términos ya construidos. Por último, la regla cuarta nos dice que esta es la única forma de obtener nuevos términos.

Ejemplos 5.2

1. *Sea L el lenguaje vacío. Los únicos términos posibles son las variables, por tanto, $Ter(L) = \mathcal{V}$.*
2. *Sea L el lenguaje de primer orden con $\mathcal{C} = \{a, b, c\}$, $\mathcal{F}_1 = \{f, h\}$, $\mathcal{F}_2 = \{g\}$, $\mathcal{R}_1 = \{R\}$, $\mathcal{R}_2 = \{P, S\}$ y $\mathcal{R}_3 = \{Q\}$.*

   Entonces son términos:

   $a,\ b,\ x,\ f(b),\ g(a, x),\ h(f(a)),\ g(g(x, y), h(c)),\ f(g(g(x, y), h(c))),\ h(h(h(h(h(h(a)))))),\ g(a, a).$

   No son términos:

   $f(a, a),\ f(g),\ g(x),\ h(d),\ h(g(a, b), x).$

Obsérvese cómo los símbolos de predicado no intervienen a la hora de formar los términos.

Definición 5.3 (variables de un término) *Definimos la aplicación* $\mathrm{Var} : Ter(L) \longrightarrow \mathcal{P}(\mathcal{V})$ *mediante:*

1. $\mathrm{Var}(c) = \emptyset$ *para toda* $c \in \mathcal{C}$.
2. $\mathrm{Var}(x) = \{x\}$ *para toda* $x \in \mathcal{V}$.

3. $\mathrm{Var}(f(t_1, t_2, \ldots, t_n)) = \bigcup_{1 \leq i \leq n} \mathrm{Var}(t_i)$

Definición 5.4 (término cerrado) *Un término t es cerrado si* $\mathrm{Var}(t) = \emptyset$.

Utilizaremos la notación $t(x_1, x_2, \ldots, x_n)$ para indicar que $\mathrm{Var}(t) \subseteq \{x_1, x_2, \ldots, x_n\}$.

Una vez definidos los términos podemos dar lo que son las fórmulas atómicas.

Fórmulas de un lenguaje.

Definición 5.5 (fórmulas atómicas) *Una* fórmula atómica *de un lenguaje L es un elemento de alguno de los conjuntos:*

1. $\{t_1 \approx t_2 : \ t_1, t_2 \in Ter(L)\}$.
2. $\{R(t_1, t_2, \ldots, t_n) : t_1, t_2, \ldots, t_n \in Ter(L) \ \text{ y } R \in \mathcal{R}_n\}$.

Las fórmulas del primer conjunto se llaman ecuaciones del lenguaje. Notaremos $At(L)$ al conjunto de las fórmulas atómicas del lenguaje L.

Ejemplo 5.3 *Sea L el lenguaje del ejemplo 5.2.*
Son fórmulas atómicas de este lenguaje:
$P(a, x), \ R(h(b)), \ Q(h(f(a)), g(a, x), h(c)), \ f(y) \approx f(g(a, x)), \ S(g(g(a, a), x), f(c)),$
$P(f(h(f(h(c)))), g(f(x), h(f(a)))), \ P(g(x, y), g(a, a)).$
No son fórmulas atómicas:
$P(g(a, x)), \ R(f(g)), \ Q(a, g(x), h(d)), \ Q(R(a), g(a, b), f(x)).$

Definición 5.6 (fórmulas) *Sea L un lenguaje de primer orden. Definimos recursivamente las fórmulas del lenguaje L, que notaremos $For(L)$ como sigue:*

1. *Toda fórmula atómica es una fórmula.*
2. *Si φ y ψ son fórmulas, también lo son $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$.*
3. *Si φ es una fórmula y x es un símbolo de variable, entonces son fórmulas $\forall x\varphi$ y $\exists x\varphi$.*
4. *No hay más fórmulas que las que cumplen alguna de las tres reglas anteriores.*

- A la hora de escribir las fórmulas vamos a seguir las mismas reglas de prioridad que teníamos en el lenguaje proposicional. La conectiva $\neg$ tiene prioridad sobre las restantes. Las conectivas $\vee$ y $\wedge$ tienen prioridad sobre $\to$ y $\leftrightarrow$.
- Los cuantificadores tienen prioridad sobre todas las conectivas. Así una expresión de la forma $\forall x\varphi \vee \psi$ significa lo mismo que $(\forall x\varphi) \vee \psi$, y diferente de $\forall x(\varphi \vee \psi)$. Esto vale lo mismo para el resto de conectivas binarias ($\wedge$, $\to$, $\leftrightarrow$) y para el otro cuantificador ($\exists x$). Con la conectiva $\neg$ no hay lugar a la confusión, pues la fórmula $\forall x \neg\varphi$ tiene una única lectura, al igual que $\neg\forall x\varphi$, $\exists x\neg\varphi$ y $\neg\exists x\varphi$.

Ejemplo 5.4 *Consideramos el lenguaje del ejemplo 5.2. Son fórmulas de este lenguaje:*

$P(f(a), g(a, x)) \wedge \neg Q(a, b, h(b)).$

$S(g(g(a, a), x), f(x)) \to \forall x P(a, x).$

$\forall x \forall y \exists z (x \not\approx z \wedge y \not\approx z)$

$\exists y (R(g(a, y)) \to (P(g(a, b), x) \wedge \neg\forall x Q(x, x, x))).$

Y no son fórmulas de este lenguaje:

$R(f(c)) \neg \wedge Q(a, b, h(d)).$

$\forall a Q(a, g(x, b), h(d)) \to \forall x P(x, f(y))$.

$\forall x Q(R(x), g(x, a), h(f(c))) \wedge \exists y S(f(a), g(a, x))$.

$R(\forall x P(x, a))$.

$R(x) \to f(x)$.

1. A la hora de escribir las fórmulas vamos a seguir las mismas reglas de prioridad que teníamos en el lenguaje proposicional. La conectiva $\neg$ tiene prioridad sobre las restantes. Las conectivas $\vee$ y $\wedge$ tienen prioridad sobre $\to$ y $\leftrightarrow$.

2. Los cuantificadores tienen prioridad sobre todas las conectivas. Así una expresión de la forma $\forall x\alpha \vee \beta$ significa lo mismo que $(\forall x\alpha) \vee \beta$, y diferente de $\forall x(\alpha \vee \beta)$. Esto vale lo mismo para el resto de conectivas binarias ($\wedge$, $\to$, $\leftrightarrow$) y para el otro cuantificador ($\exists$). Con la conectiva $\neg$ no hay lugar a la confusión, pues la fórmula $\forall x\neg\alpha$ tiene una única lectura, al igual que $\neg\forall x\alpha$, $\exists x\neg\alpha$ y $\neg\exists x\alpha$.

3. En el lenguaje proposicional las fórmulas se obtenían a partir de la fórmulas atómicas y las conectivas siguiendo unas reglas que nos permitían obtener nuevas fórmulas. El conjunto de las fórmulas atómicas era un conjunto prefijado. En un lenguaje de primer orden, como hemos podido ver, la situación es algo más compleja. Las fórmulas atómicas ya no son un conjunto prefijado, sino que se obtienen a partir del conjunto de símbolos de predicado, el símbolo de la igualdad y los términos, y éstos a su vez se obtienen mediante una combinación adecuada de símbolos de constante, de variable y de función.

   Las mismas reglas que teníamos en el lenguaje proposicional para formar las fórmulas a partir de las fórmulas atómicas y las conectivas nos valen ahora. Y añadimos dos más por la existencia de los cuantificadores.

   Esta situación podemos esquematizarla como sigue:

$$\left.\begin{array}{l}\left.\begin{array}{l}\left.\begin{array}{l}\bullet\ \text{Simb. constante}\\ \bullet\ \text{Simb. variable}\\ \bullet\ \text{Simb función}\end{array}\right\}\ \text{Términos}\\ \\ \bullet\ \text{Simb. igualdad}\\ \bullet\ \text{Simb. predicado}\end{array}\right\}\ \text{Fórmulas atómicas}\\ \\ \bullet\ \text{Conectivas}\\ \bullet\ \text{Cuantificadores}\end{array}\right\}\ \text{Fórmulas}$$

   mientras que en el lenguaje proposicional el esquema sería:

$$\left.\begin{array}{l}\bullet\ \text{Fórmulas atómicas}\\ \bullet\ \text{Conectivas}\end{array}\right\}\ \text{Fórmulas}$$

4. Vemos entonces que una fórmula en un lenguaje proposicional es una sucesión de símbolos. Estos símbolos están divididos en 6 grupos (símbolos de constante, de variable, de función, de relación (el de igualdad aparece en todos los lenguajes), conectivas y cuantificadores) más unos símbolos auxiliares (los paréntesis y la coma). Pero no toda sucesión de estos símbolos constituye una fórmula, sino que están sujetos a determinadas reglas sintácticas que hemos dado en las definiciones 5.2, 5.5 y 5.6.

   Sin embargo, cuando tenemos una sucesión de símbolos no siempre es fácil saber si se ha construido siguiendo las reglas adecuadas. Para ayudar a ver esto, es conveniente desmenuzar la posible fórmula en sus componentes más simples.

Definición 5.7 *Sea L un lenguaje de primer orden, y sea φ una fórmula del lenguaje. Definimos el conjunto de las subfórmulas de φ, que notamos $Sub(\varphi)$, como sigue:*

1. $Sub(\varphi) = \{\varphi\}$ *si φ es una fórmula atómica.*

2. $Sub(\neg\varphi) = \{\neg\varphi\} \cup Sub(\varphi)$.

3. $Sub(\varphi \vee \psi) = \{\varphi \vee \psi\} \cup Sub(\varphi) \cup Sub(\psi)$.

4. *Sub*$(\varphi \wedge \psi) = \{\varphi \wedge \psi\} \cup Sub(\varphi) \cup Sub(\psi)$.

5. *Sub*$(\varphi \to \psi) = \{\varphi \to \psi\} \cup Sub(\varphi) \cup Sub(\psi)$.

6. *Sub*$(\varphi \leftrightarrow \psi) = \{\varphi \leftrightarrow \psi\} \cup Sub(\varphi) \cup Sub(\psi)$.

7. *Sub*$(\forall x\varphi) = \{\forall x\varphi\} \cup Sub(\varphi)$.

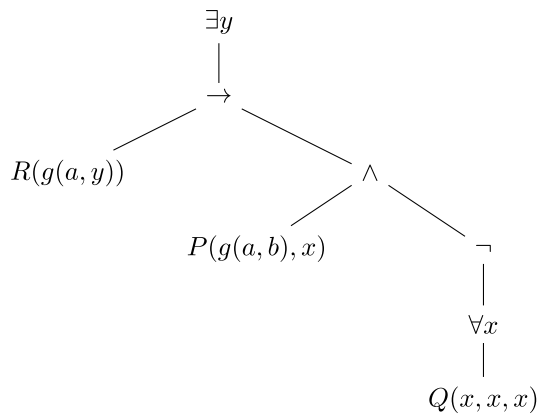
8. *Sub*$(\exists x\varphi) = \{\exists x\varphi\} \cup Sub(\varphi)$.

Definición 5.8 (fórmulas como árboles) *Sea* $L = (\mathcal{C}, \mathcal{F}, \mathcal{R}, \rho)$ *un lenguaje de primer orden. Definimos una fórmula de* L *como un árbol con etiquetas en* $\mathbb{C}\mathrm{ons}(L) \cup At(L)$ *que cumple:*

1. *Si un nodo tiene la etiqueta en* $\mathbb{C}\mathrm{ons}(L)$, *entonces la ariedad del símbolo es el número de hijos del nodo.*

2. *Si un nodo tiene la etiqueta en* $At(L)$, *entonces el nodo es una hoja.*

Ejemplo 5.5 *Sea* $\varphi = \exists y(R(g(a,y)) \to (P(g(a,b),x) \wedge \neg\forall x Q(x,x,x)))$ *una fórmula del lenguaje de primer orden del ejemplo 5.2. Vamos a escribirla en forma de árbol.*

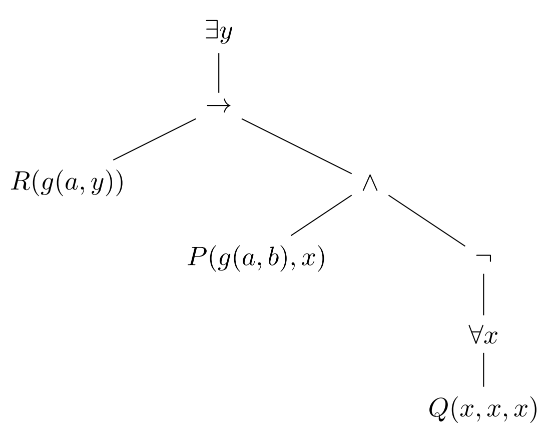


Ocurrencias libres y ligadas. Sentencias de un lenguaje.

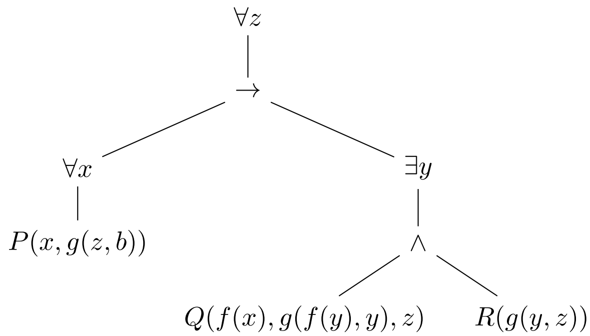
Definición 5.9 (dominio de un cuantificador) *Dado un cuantificador de una fórmula* φ *llamamos ámbito, dominio o radio de acción del cuantificador, a la subfórmula determinada por su nodo hijo.*

Ejemplo 5.6

1. *En la fórmula* $\exists y(R(g(a,y)) \to (P(g(a,b),x) \wedge \neg\forall x Q(x,x,x)))$ *aparecen dos cuantificadores. El radio de acción de* $\exists y$ *es* $R(g(a,y)) \to (P(g(a,b),x) \wedge \neg\forall x Q(x,x,x))$, *mientras que el radio de acción de* $\forall x$ *es* $Q(x,x,x)$.



2. *Sea ahora* $\varphi = \forall z(\forall x P(x, g(z,b)) \to \exists y(Q(f(x), g(f(y),y), z) \wedge R(g(y,z))))$



El radio de acción de $\forall z$ *es* $\forall x P(x, g(z, b)) \rightarrow \exists y(Q(f(x), g(f(y), y), z) \wedge R(g(y, z)))$.

El radio de acción de $\forall x$ *es* $P(x, g(z, b))$.

El radio de acción de $\exists y$ *es* $Q(f(x), g(f(y), y), z) \wedge R(g(y, z))$

Definición 5.10 (ocurrencia) *Una ocurrencia de una variable* x *en una fórmula* φ *es cada una de las veces en que* x *aparece en una fórmula atómica de* φ.

Definición 5.11 (ocurrencia ligada) *Dada una ocurrencia de una variable* x *en una fórmula* φ. *La ocurrencia de* x *es ligada si está en el ámbito de un cuantificador de la forma* $\forall x$ *o* $\exists x$. *Las ocurrencias de* x *que no son ligadas se llaman ocurrencias libres.*

Definición 5.12 (variables ligadas, variables libres) *Dada una* x *y una fórmula* φ. x *está ligada en* φ *si* x *tiene ocurrencias ligadas en* φ *y* x *está libre en* φ *si* x *tiene ocurrencias libres en* φ. *Notaremos* $\mathrm{Lig}(\varphi)$ *y* $\mathrm{Lib}(\varphi)$ *a los conjuntos de variables ligadas y libres de* φ.

Teorema 5.1 *La aplicación* $\mathrm{Lig} : For(L) \longrightarrow \mathcal{P}(\mathcal{V})$, *variables ligadas de una fórmula, verifica:*

1. $\mathrm{Lig}(\varphi) = \emptyset$ *para toda* $\varphi \in At(L)$
2. $\mathrm{Lig}(\neg\varphi) = \mathrm{Lig}(\varphi)$ *para toda* $\varphi \in For(L)$
3. $\mathrm{Lig}(\varphi \vee \psi) = \mathrm{Lig}(\varphi) \cup \mathrm{Lig}(\psi)$ *para cualesquiera* $\varphi, \psi \in For(L)$
4. $\mathrm{Lig}(\varphi \wedge \psi) = \mathrm{Lig}(\varphi) \cup \mathrm{Lig}(\psi)$ *para cualesquiera* $\varphi, \psi \in For(L)$
5. $\mathrm{Lig}(\varphi \rightarrow \psi) = \mathrm{Lig}(\varphi) \cup \mathrm{Lig}(\psi)$ *para cualesquiera* $\varphi, \psi \in For(L)$
6. $\mathrm{Lig}(\forall x\varphi) = \mathrm{Lig}(\varphi) \cup \{x\}$ *para toda* $\varphi \in For(L)$
7. $\mathrm{Lig}(\exists x\varphi) = \mathrm{Lig}(\varphi) \cup \{x\}$ *para toda* $\varphi \in For(L)$

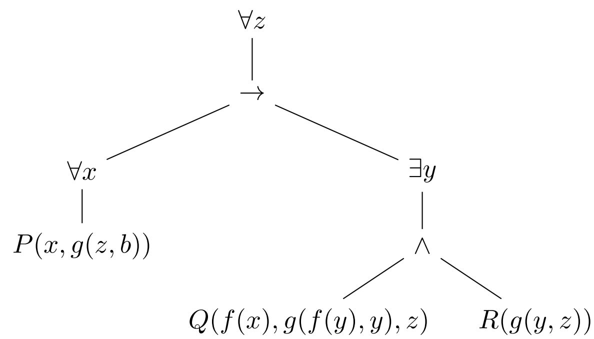
Teorema 5.2 *La aplicación* $\mathrm{Lib} : For(L) \longrightarrow \mathcal{P}(\mathcal{V})$, *variables libres de una fórmula, verifica:*

1. $\mathrm{Lib}(\varphi) = \mathrm{Var}(\varphi)$ *para toda* $\varphi \in At(L)$
2. $\mathrm{Lib}(\neg\varphi) = \mathrm{Lib}(\varphi)$ *para toda* $\varphi \in For(L)$
3. $\mathrm{Lib}(\varphi \vee \psi) = \mathrm{Lib}(\varphi) \cup \mathrm{Lib}(\psi)$ *para cualesquiera* $\varphi, \psi \in For(L)$
4. $\mathrm{Lib}(\varphi \wedge \psi) = \mathrm{Lib}(\varphi) \cup \mathrm{Lib}(\psi)$ *para cualesquiera* $\varphi, \psi \in For(L)$
5. $\mathrm{Lib}(\varphi \rightarrow \psi) = \mathrm{Lib}(\varphi) \cup \mathrm{Lib}(\psi)$ *para cualesquiera* $\varphi, \psi \in For(L)$
6. $\mathrm{Lib}(\forall x\varphi) = \mathrm{Lib}(\varphi) \setminus \{x\}$ *para toda* $\varphi \in For(L)$
7. $\mathrm{Lib}(\exists x\varphi) = \mathrm{Lib}(\varphi) \setminus \{x\}$ *para toda* $\varphi \in For(L)$

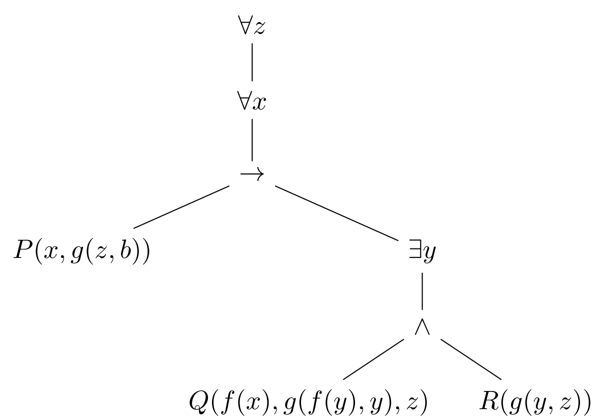
Definición 5.13 (sentencia) *Dada una fórmula* σ. σ *es una sentencia si* $\mathrm{Lib}(\sigma) = \emptyset$. *El conjunto de todas las sentencias de un lenguaje* L *lo notaremos* $\mathbb{S}\mathrm{ent}(L)$.

Ejemplo 5.7

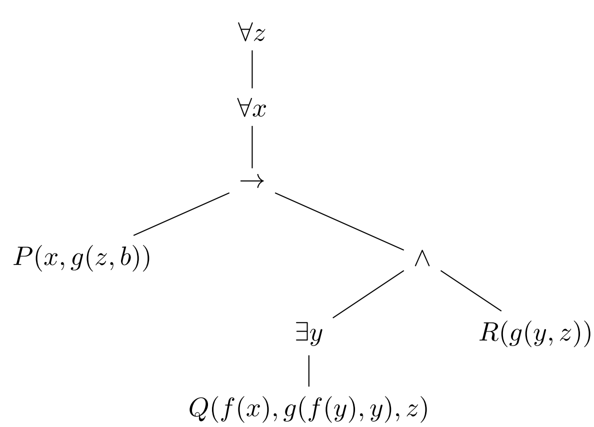
1. *La fórmula* $\forall z(\forall x P(x, g(z, b)) \to \exists y(Q(f(x), g(f(y), y), z) \wedge R(g(y, z))))$ *no es una sentencia. Hemos visto que tiene una aparición libre de la variable* x.



2. *La fórmula* $\forall z \forall x(P(x, g(z, b)) \to \exists y(Q(f(x), g(f(y), y), z) \wedge R(g(y, z))))$ *sí es una sentencia. En este caso, el radio de acción de* $\forall x$ *es* $P(x, g(z, b)) \to \exists y(Q(f(x), g(f(y), y), z) \wedge R(g(y, z)))$, *luego todas las ocurrencias de* x *son ligadas. Las de las otras variables son también ligadas.*



3. *La fórmula* $\forall z \forall x(P(x, g(z, b)) \to \exists y Q(f(x), g(f(y), y), z) \wedge R(g(y, z)))$ *no es una sentencia. En este ejemplo, la última ocurrencia de la variable* y *es libre (el radio de acción de* $\exists y$ *es* $Q(f(x), g(f(y), y), z)$.



5.2 Semántica de un lenguaje de primer orden.

5.2.1 Estructuras de un lenguaje L.

Definición 5.14 (L-estructura) *Dado un lenguaje L de primer orden (con igualdad), una L-estructura es una 4-upla*

$$\mathfrak{A} = \langle A, \{c^{\mathfrak{A}} : c \in \mathcal{C}\}, \{f^{\mathfrak{A}} : f \in \mathcal{F}\}, \{R^{\mathfrak{A}} : R \in \mathcal{R}\}\rangle$$

que cumple:

1. *$A \neq \emptyset$ es un conjunto llamado universo o dominio de la estructura. Nos indica de que cosas hablamos y precisa el significado de los cuantificadores.*
2. *Para cada $c \in \mathcal{C}$, $c^{\mathfrak{A}}$ es un elemento del universo A que leemos significado de c en $\mathfrak{A}$.*
3. *$f^{\mathfrak{A}}$ es una operación n-aria del universo A para cada $f \in \mathcal{F}_n$, es decir, $f^{\mathfrak{A}} : A^n \to A$*
4. *$R^{\mathfrak{A}}$ es una relación n-aria del universo A para cada $R \in \mathcal{R}_n$, es decir, $R^{\mathfrak{A}} \subseteq A^n$, o bien, $R^{\mathfrak{A}} : A^n \longrightarrow \{0, 1\}$*

Observación:

1. Dar una estructura es entonces elegir los elementos sobre los que queremos que las fórmulas digan algo (números, matrices, polinomios, personas, etc.), y una vez hecho esto, atribuir un significado a los símbolos de constante, de función y de relación. Hablaremos entonces de significado de constantes, de funciones y de predicados.
2. Si tenemos una estructura $\mathfrak{A}$, y c es un símbolo de constante, para no complicar mucho la notación escribiremos, si no hay lugar a la confusión, c en lugar de $c^{\mathfrak{A}}$. Así, por ejemplo, en una estructura en la que el dominio es $\mathbb{N}$ (números naturales), y al símbolo de constante c le asignamos el valor 1, escribiremos $c = 1$ en lugar de $c^{\mathfrak{A}} = 1$.
3. La misma observación anterior vale para los símbolos de función y los símbolos de predicado.
4. Si P es un símbolo de predicado 0-ario, entonces asignarle un significado a P significa elegir un valor de verdad (0 ó 1) para el predicado P. Esto es lo mismo que hacíamos cuando asignábamos un valor de verdad a una fórmula atómica en un lenguaje proposicional.
5. Si en nuestro lenguaje todos los símbolos de predicado son 0-arios, entonces en las fórmulas del lenguaje únicamente habrá símbolos de predicado y conectivas (no tienen cabida aquí los símbolos de constante, de variable y de función). En tal caso, lo que tenemos es un lenguaje proposicional.

   Dicho de otra forma, la lógica de predicados incluye como caso particular a la lógica proposicional. Concretamente como aquella en la que todos los símbolos de predicado son 0-arios.

Ejemplo 5.8 *Consideramos un lenguaje de primer orden cuyos símbolos de constante son s y a; los símbolos p y m son símbolos de función monarios y los símbolos de predicado son M, In, T monarios y MA binario.*

Sea $\mathfrak{A}$ la estructura siguiente:

- *El universo A es el conjunto de todas las personas.*
- *s = Sócrates y a = Aurora (notemos que habría que haber escrito $s^{\mathfrak{A}}$ = Sócrates e igual para a. Pero como acabamos de comentar, lo escribiremos tal y como hemos dicho para simplificar la escritura).*
- *$p(x)$ = el padre de x y $m(x)$ = la madre de x.*
- $$M(x) = \begin{cases} 1 & \textit{si } x \textit{ es mortal.} \\ 0 & \textit{en caso contrario.} \end{cases} \qquad In(x) = \begin{cases} 1 & \textit{si } x \textit{ es inteligente.} \\ 0 & \textit{en caso contrario.} \end{cases}$$

  $$T(x) = \begin{cases} 1 & \textit{si } x \textit{ es trabajador.} \\ 0 & \textit{en caso contrario.} \end{cases} \qquad MA(x,y) = \begin{cases} 1 & \textit{si } x \textit{ es más alto que } y. \\ 0 & \textit{en caso contrario.} \end{cases}$$

Aunque aún faltan por precisar algunos conceptos, con esta estructura tendríamos:

La fórmula $M(s)$ significaría Sócrates es mortal.

La fórmula $M(a)$ *significaría* Aurora es mortal.

La fórmula $\forall x M(x)$ *significaría* todas las personas son mortales.

La fórmula $M(p(s))$ *significaría* el padre de Sócrates es mortal.

La fórmula $M(p(m(a)))$ *significaría* el padre de la madre de Aurora es mortal.

La fórmula $\forall x M(p(x))$ *significaría* todos los padres son mortales.

La fórmula $\exists x T(x)$ *significaría* hay personas trabajadoras.

La fórmula $MA(a, s)$ *significaría* Aurora es más alta que Sócrates.

La fórmula $In(p(a))$ *significaría* el padre de Aurora es inteligente.

Ejemplo 5.9 *Sea ahora el lenguaje* L *en el que* $\mathcal{C} = \{a, b\}$, $\mathcal{F}_1 = \{s\}$, $\mathcal{F}_2 = \{m\}$ *y* $\mathcal{R}_1 = \{P, Pr\}$, $\mathcal{R}_2 = \{M\}$. *Sea* $\mathfrak{A}$ *la estructura siguiente:*

- *Universo:* $\mathbb{N}$.
- *Asignación de constantes:* $a = 0$, $b = 1$.
- *Asignación de funciones:* $s(x) = x + 1$, $m(x, y) = x \cdot y$.
- *Asignación de predicados:*

$$P(x) = \begin{cases} 1 & \text{si } x \text{ es par.} \\ 0 & \text{si } x \text{ es impar.} \end{cases} \quad Pr(x) = \begin{cases} 1 & \text{si } x \text{ es primo.} \\ 0 & \text{si } x \text{ no es primo.} \end{cases} \quad M(x, y) = \begin{cases} 1 & \text{si } x > y. \\ 0 & \text{si } x \leq y. \end{cases}$$

Con esta estructura tendríamos:

La fórmula $P(b)$ *significaría* uno es un número par.

La fórmula $M(b, a)$ *significaría* uno es mayor que cero.

La fórmula $Pr(s(b))$ *significaría* dos es un número primo.

La fórmula $m(a, b) \approx a$ *significaría* cero por uno es igual a cero.

Ejemplo 5.10 *Con el mismo lenguaje de este último ejemplo, consideramos la estructura* $\mathfrak{B}$ *siguiente:*

- *Dominio:* $\mathbb{Z}_4$.
- *Asignación de constantes:* $a = 0$, $b = 3$.
- *Asignación de funciones:* $s(x) = 2x + 1$, $m(x, y) = x \cdot y$.
- *Asignación de predicados:*

$$P(x) = \begin{cases} 1 & \text{si } x^2 = 0. \\ 0 & \text{si } x^2 \neq 0. \end{cases} \quad Pr(x) = \begin{cases} 1 & \text{si } x^2 = x. \\ 0 & \text{si } x^2 \neq x. \end{cases} \quad M(x, y) = \begin{cases} 1 & \text{si } x^2 + y = 1. \\ 0 & \text{si } x^2 + y \neq 1. \end{cases}$$

Con esta estructura tendríamos:

La fórmula $P(b)$ *significaría* tres al cuadrado es igual a cero.

La fórmula $M(b, a)$ *significaría* tres al cuadrado más cero es igual a uno.

La fórmula $Pr(s(b))$ *significaría* cero al cuadrado es igual a cero.

La fórmula $m(a, b) \approx a$ *significaría* cero por tres es igual a cero.

Observación:

La forma en que hemos asignado los predicados en estos dos últimos ejemplos puede resultar un poco engorrosa. Por tal motivo, emplearemos frecuentemente otras notaciones.

1. Una sería escribir $P(x_1, x_2, \cdots, x_n) \equiv$ [Condición], donde en lugar de [Condición] escribiremos la condición que deben cumplir $x_1, x_2, \cdots, x_n$ para que $P(x_1, x_2, \cdots, x_n)$ valga 1.

   Así, en la estructura del ejemplo 5.9 podríamos haber definido los predicados:

$$P(x) \equiv x \text{ es par.} \quad Pr(x) \equiv x \text{ es primo.} \quad M(x, y) \equiv x > y.$$

   Y en la del ejemplo 5.10:

$$P(x) \equiv x^2 = 0. \quad Pr(x) \equiv x^2 = x. \quad M(x, y) \equiv x^2 + y = 1.$$

2. Otra sería escribir $P =$ [Conjunto], donde en lugar de [Conjunto] iría el conjunto

$$\{(x_1, x_2, \cdots, x_n) \in D^n : P(x_1, x_2, \cdots, x_n) = 1\}.$$

   Esto, cuando podamos enumerar los elementos.

   Por ejemplo, en el caso de la estructura $\mathfrak{B}$ podríamos haber definido los predicados como:

$$P = \{0, 2\}; \quad M = \{(0, 1),\ (1, 0),\ (2, 1),\ (3, 0)\}; \quad Pr = \{0, 1\}.$$

   Para el caso de la estructura $\mathfrak{A}$ esto sería un poco más complicado, ya que los conjuntos de los que estamos hablando son infinitos. Aún así se podría decir:

   $P = \{0, 2, 4, 6, \cdots\}$ si con eso entendemos que nos referimos a todos los números pares.

   $Pr = \{2, 3, 5, 7, 11, 13, \cdots\}$ si con eso entendemos que nos referimos al conjunto de los números primos.

   Un poco más complicado es describir así el predicado M.

Asignaciones

Definición 5.15 (asignación) *Sea L un lenguaje de primer orden y $\mathfrak{A}$ una L-estructura. Una asignación en $\mathfrak{A}$ es una aplicación $\mathfrak{a} : \mathcal{V} \longrightarrow A$. Es decir, un elemento de $Ap(\mathcal{V}, A) = A^{\mathcal{V}}$.*

Ejemplo 5.11 *Consideramos un lenguaje de primer orden y una estructura en la que $A = \mathbb{Z}$. Sea $\mathfrak{a}$ una asignación que cumple $x \mapsto -2$, $y \mapsto 1$, $z \mapsto 5$.*

Nótese que decimos una asignación ya que hay infinitas que cumplen lo dado. En lugar de dar una especificación infinita de una, damos una especifificación finita de infinitas asignaciones y trabajamos con todas ellas a la vez.

Definición 5.16 (variante) *Sea $\mathfrak{a}$ una asignación en $\mathfrak{A}$, $x \in \mathcal{V}$ y $a \in A$. $\mathfrak{a}(x|a)$ es la asignación en $\mathfrak{A}$ que cumple:*

$$\mathfrak{a}(x|a)(z) = \begin{cases} a & \text{si } z = x \\ \mathfrak{a}(z) & \text{en otro caso} \end{cases}$$

y la llamamos una variante de $\mathfrak{a}$ en x.

Ejemplo 5.12 *Consideramos un lenguaje de primer orden y una estructura en la que $A = \mathbb{Z}$. Sea $\mathfrak{a}$ la asignación $x \mapsto -2$, $y \mapsto 1$, $z \mapsto 5$. A continuación damos explícitamente las asignaciones $\mathfrak{a}$, $\mathfrak{a}(x|2)$, $\mathfrak{a}(x|5)$, $\mathfrak{a}(y|-1)$, $\mathfrak{a}(z|3)$ y $\mathfrak{a}(z|5)$.*

$\mathfrak{a}$	$\mathfrak{a}(x\|2)$	$\mathfrak{a}(x\|5)$	$\mathfrak{a}(y\|-1)$	$\mathfrak{a}(z\|3)$	$\mathfrak{a}(z\|5)$
$x \mapsto -2$	$x \mapsto 2$	$x \mapsto 5$	$x \mapsto -2$	$x \mapsto -2$	$x \mapsto -2$
$y \mapsto 1$	$y \mapsto 1$	$y \mapsto 1$	$y \mapsto -1$	$y \mapsto 1$	$y \mapsto 1$
$z \mapsto 5$	$z \mapsto 5$	$z \mapsto 5$	$z \mapsto 5$	$z \mapsto 3$	$z \mapsto 5$

Nótese que $\mathfrak{a}(z|5) = \mathfrak{a}$. En general, se tiene que $\mathfrak{a}(x|e) = \mathfrak{a}$ cuando $\mathfrak{a}(x) = e$.

Semántica de términos.

Definición 5.17 (significado de un término) *Dados* $t \in Ter(L)$ *y* $\mathfrak{A} \in \mathcal{E}(L)$ *definimos la aplicación*

$$t^{\mathfrak{A}} : A^{\mathcal{V}} \longrightarrow A$$

que cumple:

1. $c^{\mathfrak{A}}(\mathfrak{a}) = c^{\mathfrak{A}}$ *para toda* $c \in \mathcal{C}$.
2. $x^{\mathfrak{A}}(\mathfrak{a}) = \mathfrak{a}(x)$ *para toda* $x \in \mathcal{V}$.
3. $f(t_1, t_2, \ldots, t_n)^{\mathfrak{A}}(\mathfrak{a}) = f^{\mathfrak{A}}(t_1^{\mathfrak{A}}(\mathfrak{a}), t_2^{\mathfrak{A}}(\mathfrak{a}), \ldots, t_n^{\mathfrak{A}}(\mathfrak{a}))$ *para todo* $f \in \mathcal{F}_n$ *y todo* $t_1, t_2, \ldots, t_n \in Ter(L)$.

Al elemento de A $t^{\mathfrak{A}}(\mathfrak{a})$ *lo llamamos significado del término* t *en la estructura* $\mathfrak{A}$ *para la asignación* $\mathfrak{a}$.

Veamos algunos ejemplos:

Ejemplo 5.13
Consideramos el lenguaje de primer orden dado por $\mathcal{C} = \{a, b\}$, $\mathcal{F}_1 = \{s\}$, $\mathcal{F}_2 = \{m\}$, $\mathcal{R}_1 = \{P, Pr\}$ *y* $\mathcal{R}_2 = \{M\}$. *Sea la estructura* $\mathfrak{A}$ *que vimos en el ejemplo 5.9, y la asignación* $\mathfrak{a} : \mathcal{V} \to \mathbb{N}$ *dada por* $\mathfrak{a}(x) = 2$, $\mathfrak{a}(y) = 5$, $\mathfrak{a}(z) = 1$. *Vamos a considerar algunos términos y calcular su significado en* $\mathfrak{A}$ *para la asignación* $\mathfrak{a}$.

- $t = s(a)$.
  $t^{\mathfrak{A}}(\mathfrak{a}) = s^{\mathfrak{A}}(a^{\mathfrak{A}}) = s^{\mathfrak{A}}(0) = 0 + 1 = 1$. *Para abreviar podemos escribir* $t = s(a) = s(0) = 1$.

- $t = m(s(a), x)$.
  $t^{\mathfrak{A}}(\mathfrak{a}) = m^{\mathfrak{A}}(s^{\mathfrak{A}}(a^{\mathfrak{A}}), \mathfrak{a}(x)) = m^{\mathfrak{A}}(1, 2) = 2$. *Para abreviar podemos escribir* $m(s(a), x) = m(0 + 1, 2) = m(1, 2) = 2$.

- $t = s(y)$.
  $t^{\mathfrak{A}}(\mathfrak{a}) = s^{\mathfrak{A}}(\mathfrak{a}(y)) = s^{\mathfrak{A}}(5) = 5 + 1 = 6$. *Para abreviar podemos escribir* $s(y) = s(5) = 5 + 1 = 6$.

- $t = m(m(s(a), x), s(y))$.
  $t^{\mathfrak{A}}(\mathfrak{a}) = m^{\mathfrak{A}}(m^{\mathfrak{A}}(s^{\mathfrak{A}}(0), 2), s^{\mathfrak{A}}(5)) = m^{\mathfrak{A}}(m^{\mathfrak{A}}(1, 2)), 6) = m^{\mathfrak{A}}(2, 6) = 12$.
  Abreviando, $m(m(s(0), 2)), s(5)) = m(m(1, 2)), 6) = m(2, 6) = 12$.

- *Sea* $t = m(s(s(b)), m(s(x), m(b, m(x, y))))$.
  Vamos a calcular $t^{\mathfrak{A}}(\mathfrak{a})$.
  $$\begin{array}{rcl} m(s(s(b)), m(s(x), m(b, m(x, y)))) & = & m(s(s(1)), m(s(2), m(1, m(2, 5)))) \\ & = & m(s(2), m(3, m(1, 10))) \\ & = & m(3, m(3, 10)) \\ & = & m(3, 30) = 90 \end{array}.$$
  Es decir, $t^{\mathfrak{A}}(\mathfrak{a}) = 90$.

Ejemplo 5.14
Con el mismo lenguaje, la estructura $\mathfrak{B}$ *del ejemplo 5.10 y la asignación dada por* $\mathfrak{a}(x) = 3$, $\mathfrak{a}(y) = 1$, $\mathfrak{a}(z) = 2$ *vamos a calcular el significado de los mismos términos.*

- $s(a) = s(0) = 2 \cdot 0 + 1 = 1$.
- $m(s(a), x) = m(1, 3) = 1 \cdot 3 = 3$.
- $s(y) = s(1) = 2 \cdot 1 + 1 = 3$.
- $m(m(s(a), x), s(y)) = m(3, 3) = 3 \cdot 3 = 1$.
- $$\begin{array}{rcl} m(s(s(b)), m(s(x), m(b, m(x, y)))) & = & m(s(s(3)), m(s(3), m(3, m(3, 1)))) \\ & = & m(s(3), m(3, m(3, 3))) \\ & = & m(3, m(3, 1)) \\ & = & m(3, 3) = 1. \end{array}$$

Ejemplo 5.15
Dado el lenguaje de primer orden definido por $\mathcal{C} = \{a, c\}$, $\mathcal{F}_1 = \{f\}$, $\mathcal{F}_2 = \{g\}$, $\mathcal{R}_1 = \{R\}$, $\mathcal{R}_2 = \{Q\}$, *consideramos la estructura siguiente que llamaremos* $\mathfrak{C}$*:*

Dominio $\mathbb{Z}_3$.

Constantes $a = 0,\ c = 2$.

Funciones $f(x) = x + 1,\ g(x, y) = x + y$.

Predicados $R = \{0\},\ Q = \{(0, 1),\ (1, 2),\ (2, 0)\}$.

Consideramos la asignación $x \mapsto 0$. *Entonces:*

- $x = 0$.
- $a = 0$.
- $c = 2$.
- $f(x) = f(0) = 1$.
- $f(c) = f(2) = 0$.
- $g(a, f(x)) = g(0, f(0)) = g(0, 1) = 1$.
- $f(g(a, f(x))) = f(g(0, f(0))) = f(g(0, 1)) = f(1) = 2$.
- $g(a, c) = g(0, 2) = 2$.
- $g(g(a, c), f(g(a, f(x)))) = g(g(0, 2), f(g(0, f(0)))) = g(2, f(g(0, 1))) = g(2, f(1)) = g(2, 2) = 1$.
- $g(f(c), f(x)) = g(f(2), f(0)) = g(0, 1) = 1$.
- $$\begin{aligned} g(f(f(x)), g(f(a), g(a, f(c)))) &= f(f(x)) + g(f(a), g(a, f(c))) \\ &= f(x) + 1 + f(a) + g(a, f(c)) \\ &= x + 1 + 1 + a + 1 + a + f(c) \\ &= x + 1 + 1 + a + 1 + a + c + 1 \\ &= 0 + 1 + 1 + 0 + 1 + 0 + 2 + 1 \\ &= 0. \end{aligned}$$

Semántica de fórmulas.

Definición 5.18 (conjunto de satisfacibilidad) *Para cada* $\varphi \in For(L)$ *y* $\mathfrak{A} \in \mathcal{E}(L)$ *definimos el conjunto de satisfacibilidad de* φ *en* $\mathfrak{A}$ *como el subconjunto* $\varphi^{\mathfrak{A}} \subseteq A^{\mathcal{V}}$ *que cumple:*

1. $(t_1 \approx t_2)^{\mathfrak{A}} = \{\mathfrak{a} \in A^{\mathcal{V}} :\ t_1^{\mathfrak{A}}(\mathfrak{a}) = t_2^{\mathfrak{A}}(\mathfrak{a})\}$.
2. $R(t_1, t_2, \ldots, t_n)^{\mathfrak{A}} = \{\mathfrak{a} \in A^{\mathcal{V}} :\ \langle t_1^{\mathfrak{A}}(\mathfrak{a}), t_2^{\mathfrak{A}}(\mathfrak{a}), \ldots, t_n^{\mathfrak{A}}(\mathfrak{a})\rangle \in R^{\mathfrak{A}}\}$.
3. $(\neg\varphi)^{\mathfrak{A}} = A^{\mathcal{V}} \setminus \varphi^{\mathfrak{A}}$.
4. $(\varphi \vee \psi)^{\mathfrak{A}} = \varphi^{\mathfrak{A}} \cup \psi^{\mathfrak{A}}$.
5. $(\varphi \wedge \psi)^{\mathfrak{A}} = \varphi^{\mathfrak{A}} \cap \psi^{\mathfrak{A}}$.
6. $(\varphi \to \psi)^{\mathfrak{A}} = (A^{\mathcal{V}} \setminus \varphi^{\mathfrak{A}}) \cup \psi^{\mathfrak{A}}$.
7. $(\varphi \leftrightarrow \psi)^{\mathfrak{A}} = (\varphi \to \psi)^{\mathfrak{A}} \cap (\psi \to \varphi)^{\mathfrak{A}}$.
8. $(\forall x\varphi)^{\mathfrak{A}} = \{\mathfrak{a} \in A^{\mathcal{V}} :\ V_x(\mathfrak{a}) \subseteq \varphi^{\mathfrak{A}}\}$.
9. $(\exists x\varphi)^{\mathfrak{A}} = \{\mathfrak{a} \in A^{\mathcal{V}} :\ \text{existe } a \in A \text{ tal que } \mathfrak{a}(x|a) \in \varphi^{\mathfrak{A}}\}$.

Definiciones 5.19 (satisfacibilidad) *Sean* $\Gamma \cup \{\varphi\} \subseteq For(L)$, $\mathfrak{A}$ *una* L*-estructura y* $\mathfrak{a}$ *una asignación en* $\mathfrak{A}$.

- $\mathfrak{a}$ *satisface a* φ *en* $\mathfrak{A}$ *si* $\mathfrak{a} \in \varphi^{\mathfrak{A}}$. *También diremos que* φ *es verdadera para* $\mathfrak{a}$ *en* $\mathfrak{A}$.
- *El conjunto de satisfacibilidad de* Γ *en* $\mathfrak{A}$ *es* $\Gamma^{\mathfrak{A}} = \bigcap\{\gamma^{\mathfrak{A}} : \gamma \in \Gamma\}$.
- Γ *es satisfacible en* $\mathfrak{A}$ *si* $\Gamma^{\mathfrak{A}} \neq \emptyset$.
- Γ *es satisfacible si existe* $\mathfrak{A}$ *tal que* Γ *es satisfacible en* $\mathfrak{A}$. *Este hecho lo notaremos* $\mathbb{S}\mathrm{at}(\Gamma)$.

Ejemplo 5.16 *Sea* L *el lenguaje y* $\mathfrak{A}$ *la estructura del ejemplo 5.9. Vamos a hallar el conjunto de satisfacibilidad de algunas fórmulas de* L *en la estructura* $\mathfrak{A}$.

- $P(a)^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ 0 \text{ es par } \} = \mathbb{N}^{\mathcal{V}}$.
- $P(b)^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ 1 \text{ es par } \} = \emptyset$.
- $P(x)^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ \mathfrak{a}(x) \text{ es par } \}$.
- $(x \approx y)^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ \mathfrak{a}(x) = \mathfrak{a}(y)\}$.
- $(z \not\approx y)^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ \mathfrak{a}(z) \neq \mathfrak{a}(y)\}$.
- $P(s(z))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ \mathfrak{a}(z) + 1 \text{ es par } \} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ \mathfrak{a}(z) \text{ es impar } \}$.
- $Pr(s(b))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ 2 \text{ es primo } \} = \mathbb{N}^{\mathcal{V}}$.
- $M(m(x, s(b)), s(s(y)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ 2\mathfrak{a}(x) > \mathfrak{a}(y) + 2\}$.
- $(s(z) \approx m(x, s(x)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ \mathfrak{a}(z) + 1 = \mathfrak{a}(x)^2 + \mathfrak{a}(x)\}$.

Ejemplo 5.17 *Consideramos el mismo lenguaje, pero la estructura* $\mathfrak{B}$ *del ejemplo 5.10. En tal caso:*

- $P(a)^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ 0^2 = 0 \ \} = \mathbb{Z}_4^{\mathcal{V}}$.
- $P(b)^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ 3^2 = 0 \ \} = \emptyset$.
- $P(x)^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ \mathfrak{a}(x)^2 = 0 \ \}$.
- $(x \approx y)^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ \mathfrak{a}(x) = \mathfrak{a}(y)\}$.
- $(z \not\approx y)^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ \mathfrak{a}(z) \neq \mathfrak{a}(y)\}$.
- $P(s(z))^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ (2\mathfrak{a}(z) + 1)^2 = 1 = 0 \ \} = \emptyset$.
- $Pr(s(b))^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ (2 \cdot 3 + 1)^2 = 1 = (2 \cdot 3 + 1) = 3 \ \} = \emptyset$.
- $M(m(x, s(b)), s(s(y)))^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ (3\mathfrak{a}(x))^2 + 2(2\mathfrak{a}(y) + 1) + 1 = \mathfrak{a}(x)^2 + 3 = 1\} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ \mathfrak{a}(x)^2 = 2 \ \} = \emptyset$.
- $(s(z) \approx m(x, s(x)))^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} :\ 2\mathfrak{a}(z) + 1 = 2\mathfrak{a}(x)^2 + \mathfrak{a}(x) \ \}$.

Ejemplo 5.18 *Si consideramos las estructuras el lenguaje* $L = \mathcal{R}_2 = \{R\}$ *y las estructuras* $\mathfrak{N} = \langle \mathbb{N}, \leq \rangle$ *y* $\mathfrak{Z} = \langle \mathbb{Z}, \leq \rangle$ *tenemos que:*

- $(R(y, x) \wedge x \not\approx y)^{\mathfrak{N}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ \mathfrak{a}(y) < \mathfrak{a}(x)\}$
- $(\exists y(R(y, x) \wedge x \not\approx y))^{\mathfrak{N}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} :\ \mathfrak{a}(x) \neq 0\}$
- $(\forall x \exists y(R(y, x) \wedge x \not\approx y))^{\mathfrak{N}} = \emptyset$
- $(R(y, x) \wedge x \not\approx y)^{\mathfrak{Z}} = \{\mathfrak{a} \in \mathbb{Z}^{\mathcal{V}} :\ \mathfrak{a}(y) < \mathfrak{a}(x)\}$
- $(\exists y(R(y, x) \wedge x \not\approx y))^{\mathfrak{Z}} = \mathbb{Z}^{\mathcal{V}}$
- $(\forall x \exists y(R(y, x) \wedge x \not\approx y))^{\mathfrak{Z}} = \mathbb{Z}^{\mathcal{V}}$

Ejemplo 5.19 *Sean L y $\mathfrak{A}$ el lenguaje y la estructura del ejemplo 5.9. Consideremos las siguientes asignaciones*

$$\begin{array}{cccccc}
\mathfrak{a} & \mathfrak{b} & \mathfrak{c} & \mathfrak{d} & \mathfrak{e} & \mathfrak{f} \\
x \mapsto 3 & x \mapsto 2 & x \mapsto 5 & x \mapsto 4 & x \mapsto 1 & x \mapsto 5 \\
y \mapsto 5 & y \mapsto 1 & y \mapsto 1 & y \mapsto 2 & y \mapsto 1 & y \mapsto 2 \\
z \mapsto 11 & z \mapsto 5 & z \mapsto 5 & z \mapsto 4 & z \mapsto 3 & z \mapsto 2
\end{array}$$

- $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$, $\mathfrak{d}$, $\mathfrak{e}$ *y* $\mathfrak{f}$ *satisfacen a* $P(a)$ *en* $\mathfrak{A}$.
- $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$, $\mathfrak{d}$, $\mathfrak{e}$ *y* $\mathfrak{f}$ *no satisfacen a* $P(b)$ *en* $\mathfrak{A}$.
- $\mathfrak{b}$ *y* $\mathfrak{d}$ *satisfacen a* $P(x)$ *en* $\mathfrak{A}$ *y* $\mathfrak{a}$, $\mathfrak{c}$, $\mathfrak{e}$ *y* $\mathfrak{f}$ *no satisfacen a* $P(x)$ *en* $\mathfrak{A}$.
- $\mathfrak{e}$ *satisface a* $(x \approx y)$ *en* $\mathfrak{A}$ *y* $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$, $\mathfrak{d}$ *y* $\mathfrak{f}$ *no satisfacen a* $(x \approx y)$ *en* $\mathfrak{A}$.
- $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$, $\mathfrak{d}$ *y* $\mathfrak{e}$ *satisfacen a* $(z \not\approx y)$ *en* $\mathfrak{A}$ *y* $\mathfrak{f}$ *no satisface a* $(z \not\approx y)$ *en* $\mathfrak{A}$.
- $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$ *y* $\mathfrak{e}$ *satisfacen a* $P(s(z))$ *en* $\mathfrak{A}$ *y* $\mathfrak{d}$ *y* $\mathfrak{f}$ *no satisfacen a* $P(s(z))$ *en* $\mathfrak{A}$.
- $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$, $\mathfrak{d}$, $\mathfrak{e}$ *y* $\mathfrak{f}$ *satisfacen a* $Pr(s(b))$ *en* $\mathfrak{A}$.
- $\mathfrak{b}$, $\mathfrak{c}$, $\mathfrak{d}$ *y* $\mathfrak{f}$ *satisfacen a* $M(m(x,s(b)),s(s(y)))$ *en* $\mathfrak{A}$ *y* $\mathfrak{a}$ *y* $\mathfrak{e}$ *no satisfacen a* $M(m(x,s(b)),s(s(y)))$ *en* $\mathfrak{A}$.
- $\mathfrak{a}$, $\mathfrak{b}$, ~~$\mathfrak{c}$, $\mathfrak{d}$, $\mathfrak{e}$ *y* $\mathfrak{f}$ *no*~~ *satisfacen a* $(s(z) \approx m(x,s(x)))$ *en* $\mathfrak{A}$. **y c,d,e,f no satisfacen**

Ejemplo 5.20 *Consideramos el mismo lenguaje, pero la estructura del ejemplo 5.10 y la asignación* $\mathfrak{a}(x) = 2$, $\mathfrak{a}(y) = 1$, $\mathfrak{a}(z) = 0$. *En tal caso:*

- $\mathfrak{a}$ *satisface a* $P(a)$ *en* $\mathfrak{B}$.
- $\mathfrak{a}$ *no satisface a* $Pr(s(b))$ *en* $\mathfrak{B}$.
- $\mathfrak{a}$ *no satisface a* $M(m(x,s(b)),s(s(y)))$ *en* $\mathfrak{B}$, *pues* $m(x,s(b)) = 2 \cdot 3 = 2$, $s(s(y)) = 3$ *y* $2^2 + 3 \neq 1$.
- $\mathfrak{a}$ *no satisface a* $s(z) \approx m(x,s(x))$ *en* $\mathfrak{B}$, *pues* $s(z) = 1$, $m(x,s(x)) = 2 \cdot 1 = 2$ *y* $1 \neq 2$.

Ejemplo 5.21 *Consideramos nuevamente el lenguaje y la estructura del ejemplo 5.9. Halla los conjuntos de satisfacibilidad de las siguientes fórmulas.*

1. $\forall x P(m(x,s(x)))$
   Hallemos antes
   $P(m(x,s(x)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \ \mathfrak{a}(x)(\mathfrak{a}(x)+1) \text{ es par } \} = \mathbb{N}^{\mathcal{V}}$.
   Por tanto,
   $(\forall x P(m(x,s(x))))^{\mathfrak{A}} = \mathbb{N}^{\mathcal{V}}$.
   También podíamos haber traducido la fórmula $\forall x P(m(x,s(x)))$, *que en este caso viene a decir* $\forall x\ x(x+1)$ *es par, lo cual es cierto sea quien sea* $x \in \mathbb{N}$.

2. $\exists x(P(x) \wedge Pr(x))$
   Hallemos antes
   $(P(x) \wedge Pr(x))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \ \mathfrak{a}(x) \text{ es par } \} \cap \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \ \mathfrak{a}(x) \text{ es primo } \} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \ \mathfrak{a}(x) = 2\}$.
   Por tanto,
   $(\exists x(P(x) \wedge Pr(x)))^{\mathfrak{A}} = \mathbb{N}^{\mathcal{V}}$
   En este ejemplo, la fórmula se podría haber traducido como que existe x que es par y primo, *que sabemos que es cierto.*

3. $\exists x P(x) \wedge Pr(x)$
   $(\exists x P(x) \wedge Pr(x))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \ \text{ existe un natural par } \} \cap \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \ \mathfrak{a}(x) \text{ es primo}\} =$
   $\mathbb{N}^{\mathcal{V}} \cap \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \ \mathfrak{a}(x) \text{ es primo } \} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \ \mathfrak{a}(x) \text{ es primo } \}$

4. $\forall x(P(x) \wedge Pr(x) \rightarrow (x \approx s(b)))$
   Esta fórmula dice que si x es un número par y primo entonces $x = 2$, que sabemos que es cierto.
   $(P(x) \wedge Pr(x))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) \text{ par } \} \cap \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) \text{ primo } \} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) = 2 \}$
   $(x \approx s(b))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) = 2 \}$
   $(P(x) \wedge Pr(x) \rightarrow (x \approx s(b)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) \neq 2 \} \cup \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) = 2 \} = \mathbb{N}^{\mathcal{V}}$
   Por tanto,
   $(\forall x(P(x) \wedge Pr(x) \rightarrow (x \approx s(b))))^{\mathfrak{A}} = \mathbb{N}^{\mathcal{V}}$

5. $\forall x(P(x) \wedge Pr(x) \rightarrow M(x, s(b)))$
   Esta fórmula dice que si x es un número par y primo entonces $x > 2$, que sabemos que es falso.
   $(P(x) \wedge Pr(x))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) \text{ par } \} \cap \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) \text{ primo } \} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) = 2 \}$
   $(M(x, s(b)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) > 2 \}$
   $(P(x) \wedge Pr(x) \rightarrow M(x, s(b)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) \neq 2 \} \cup \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) > 2 \} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \mathfrak{a}(x) \neq 2 \}$
   Por tanto,
   $(\forall x(P(x) \wedge Pr(x) \rightarrow M(x, s(b))))^{\mathfrak{A}} = \emptyset$

Ejemplo 5.22
Sea ahora el lenguaje de primer orden definido por $\mathcal{C} = \{a, c\}$, $\mathcal{F}_1 = \{f\}$, $\mathcal{F}_2 = \{g\}$, $\mathcal{R}_1 = \{R\}$, $\mathcal{R}_2 = \{Q\}$ y la estructura $\mathfrak{C}$ del ejemplo 5.15.

Universo $\mathbb{Z}_3$.

Constantes $a = 0$, $c = 2$.

Funciones $f(x) = x + 1$, $g(x, y) = x + y$.

Predicados $R = \{0\}$, $Q = \{(0, 1),\ (1, 2),\ (2, 0)\}$,

Vamos a hallar el conjunto de satisfacibilidad de las siguientes fórmulas:

1. $\forall x Q(x, f(x))$
   $(Q(x, f(x)))^{\mathfrak{C}} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : (\mathfrak{a}(x), \mathfrak{a}(x) + 1) \in Q \} = \mathbb{Z}_3^{\mathcal{V}}$
   Por tanto,
   $(\forall x Q(x, f(x)))^{\mathfrak{C}} = \mathbb{Z}_3^{\mathcal{V}}$

2. $\forall x Q(x, c)$
   $(Q(x, c))^{\mathfrak{C}} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : (\mathfrak{a}(x), 2) \in Q \} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \mathfrak{a}(x) = 1 \}$
   Por tanto,
   $(\forall x Q(x, c))^{\mathfrak{C}} = \emptyset$

3. $\exists x Q(x, f(c))$
   $(Q(x, f(c))^{\mathfrak{C}} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : (\mathfrak{a}(x), 0) \in Q \} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \mathfrak{a}(x) = 2 \}$
   Por tanto,
   $(\exists x Q(x, f(c)))^{\mathfrak{C}} = \mathbb{Z}_3^{\mathcal{V}}$

4. $\exists x Q(x, f(f(x)))$
   $(Q(x, f(f(x))))^{\mathfrak{C}} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : (\mathfrak{a}(x), \mathfrak{a}(x) + 2) \in Q \} = \emptyset$
   Por tanto,
   $(\exists x Q(x, f(f(x))))^{\mathfrak{C}} = \emptyset$

5. $\exists x(R(x) \rightarrow \neg R(x))$
   $(R(x) \rightarrow \neg R(x))^{\mathfrak{C}} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \mathfrak{a}(x) \neq 0 \} \cup \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \mathfrak{a}(x) \neq 0 \} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \mathfrak{a}(x) \neq 0 \}$
   Por tanto,
   $(\exists x(R(x) \rightarrow \neg R(x)))^{\mathfrak{C}} = \mathbb{Z}_3^{\mathcal{V}}$

6. $\exists x R(x) \to \neg R(x)$
   $(R(x))^{\mathfrak{C}} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \ \mathfrak{a}(x) = 0\}$
   $(\exists x R(x))^{\mathfrak{C}} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \ \mathfrak{a}(x) = 0\} = \mathbb{Z}_3^{\mathcal{V}}$
   $(\neg R(x))^{\mathfrak{C}} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 0\}$
   Por tanto,
   $(\exists x R(x) \to \neg R(x))^{\mathfrak{C}} = \emptyset \cup \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 0\} = \{\mathfrak{a} \in \mathbb{Z}_3^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 0\}$

Hemos visto en estos ejemplos como la satisfacibilidad de una fórmula sólo depende de las variables que aparecen libres en la fórmula y de la estructura que consideremos. Por tanto para las sentencias la satisfacibilidad no depende de la asignación que escojamos, sino solo de la estructura. Este hecho queda reflejado en el siguiente lema:

Lema 5.3 (Lema de coincidencia) *Sea φ una fórmula de un lenguaje de primer orden, y sea $\mathfrak{A}$ una estructura. Supongamos que* $\mathrm{Lib}(\alpha) \subseteq \{x_1, x_2, \ldots, x_n\}$. *Sean $\mathfrak{a}_1, \mathfrak{a}_2$ dos valoraciones tales que*

$$\mathfrak{a}_1(x_i) = \mathfrak{a}_2(x_i) \text{ para } i = 1, 2, \ldots, n.$$

Entonces $\mathfrak{a}_1 \in \varphi^{\mathfrak{A}}$ si, y sólo si, $\mathfrak{a}_2 \in \varphi^{\mathfrak{A}}$.

Es decir, a la hora de interpretar una fórmula, no importa como actúe la valoración sobre las variables ligadas. Sólo tiene relevancia sobre las variables libres.

Como consecuencia, si la fórmula es una sentencia, es decir, no hay ninguna ocurrencia libre de ninguna variable, entonces la interpretación de la fórmula depende únicamente de la estructura. No depende de la valoración que tomemos.

Corolario 5.4 *Si $\sigma \in \mathbb{S}\mathrm{ent}(L)$, entonces para cada $\mathfrak{A}$ se cumple $\sigma^{\mathfrak{A}} = \emptyset$ o $\sigma^{\mathfrak{A}} = A^{\mathcal{V}}$.*

Observación:

De ahora en adelante, en muchas ocasiones en que queramos dar algún ejemplo y necesitemos un lenguaje de primer orden, no especificaremos los elementos del lenguaje, sino que daremos las fórmulas que necesitemos y supondremos que en nuestro lenguaje tenemos todos los símbolos que aparecen en las fórmulas. La escritura de las fórmulas debe ser coherente. Así, si aparece varias veces un símbolo de función o de predicado, en todas las ocasiones debe tener la misma aridad.

Por ejemplo, si decimos:

> Sea $\forall x(Q(x, f(a)) \to \exists y(P(y) \wedge Q(f(x), y)))$ una fórmula de un lenguaje de primer orden, entenderemos que estamos hablando de un lenguaje de primer orden en el que entre los símbolos de constante está a (puede que algunos más), entre los símbolos de función está f (y que es un símbolo (unario) y entre los símbolos de predicado está P (unario) y Q (binario).

5.2.2 Clasificación semántica de las fórmulas

Al igual que en el lenguaje proposicional, vamos a dar una clasificación de las fórmulas atendiendo a su satisfacibilidad en determinadas estructuras.

Definiciones 5.20 *Sea φ una fórmula de un lenguaje de primer orden L y $\mathfrak{A}$ una L-estructura.*

5.4 se hace con eso

1. *φ es* válida o cierta *en $\mathfrak{A}$ si $\varphi^{\mathfrak{A}} = A^{\mathcal{V}}$. Es decir, todas las asignaciones en $\mathfrak{A}$ satisfacen a φ. También diremos que $\mathfrak{A}$ es modelo de φ y lo notaremos $\mathfrak{A} \models \varphi$.*

2. *φ es* satisfacible *en $\mathfrak{A}$ si $\varphi^{\mathfrak{A}} \neq \emptyset$. Es decir, hay una asignación $\mathfrak{a}$ en $\mathfrak{A}$ que satisface a φ.*

3. *φ es* refutable *en $\mathfrak{A}$ si $\varphi^{\mathfrak{A}} \neq A^{\mathcal{V}}$. Es decir, hay una asignación $\mathfrak{a}$ en $\mathfrak{A}$ que no satisface a φ.*

4. *φ es* no válida o falsa *en $\mathfrak{A}$ si $\varphi^{\mathfrak{A}} = \emptyset$. Es decir, ninguna asignación $\mathfrak{a}$ en $\mathfrak{A}$ satisface a φ.*

Observación:

Si σ es una sentencia y $\mathfrak{A}$ es una estructura, puesto que el valor de verdad de σ no depende de la asignación, si es satisfacible en $\mathfrak{A}$ entonces es válida en $\mathfrak{A}$ y si es refutable en $\mathfrak{A}$ entonces es falsa en $\mathfrak{A}$.

Ejemplo 5.23 *Sea* $\alpha = \forall x P(m(x, s(x)))$ *si utilizamos las estructuras* $\mathfrak{A}$ *y* $\mathfrak{B}$ *de los ejemplos 5.9 y 5.10 tendríamos:*
$P(m(x, s(x)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{N}^{\mathcal{V}} : \ \mathfrak{a}(x)(\mathfrak{a}(x)+1) \text{ es par } \} = \mathbb{N}^{\mathcal{V}}$.
Por tanto,
$(\forall x P(m(x, s(x))))^{\mathfrak{A}} = \mathbb{N}^{\mathcal{V}}$.
y,
$P(m(x, s(x)))^{\mathfrak{B}} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} : \ (2\mathfrak{a}(x)^2 + \mathfrak{a}(x))^2 = 0 \ \} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} : \ \mathfrak{a}(x)^2 = 0 \ \} = \{\mathfrak{a} \in \mathbb{Z}_4^{\mathcal{V}} : \ \mathfrak{a}(x) = 0 \text{ o } \mathfrak{a}(x) = 2 \ \}$.
Por tanto,
$(\forall x P(m(x, s(x))))^{\mathfrak{B}} = \emptyset$.

Ejemplo 5.24
Sea L el lenguaje de primer orden $\mathcal{C} = \{b\}$, $\mathcal{F}_1 = \{f\}$, $\mathcal{F}_2 = \{g\}$, $\mathcal{R}_1 = \{P\}$ *y* $\mathcal{R}_2 = \{Q\}$. *Sea la estructura siguiente:*

Universo $\mathbb{Z}_5$.

Constantes $b = 2$.

Funciones $f(x) = x^2 + 1$, $g(x, y) = x \cdot y$.

Predicados $P(x) \equiv x^4 = 1$, $Q(x, y) \equiv x^2 + y = 0$.

Vamos a tomar varias fórmulas, y estudiar si son válidas, satisfacibles, refutables o no válidas en $\mathfrak{A}$.

1. $\forall x(P(x) \to \exists y(b \approx g(x, y)))$. *Esta fórmula es una sentencia. Por tanto, o es válida en* $\mathfrak{A}$ *o es no válida en* $\mathfrak{A}$
   $(b \approx g(x, y))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x)\mathfrak{a}(y) = 2\}$
   $(\exists y(b \approx g(x, y)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 0\}$
   $(P(x))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x)^4 = 1\} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x)^4 \neq 0\}$
   $(P(x) \to \exists y(b \approx g(x, y)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x) = 0\} \cup \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 0\} = \mathbb{Z}_5^{\mathcal{V}}$
   Por tanto,
   $(\forall x(P(x) \to \exists y(b \approx g(x, y))))^{\mathfrak{A}} = \mathbb{Z}_5^{\mathcal{V}}$
   Con esto tenemos que $\forall x(P(x) \to \exists y(b \approx g(x, y)))$ *es válida en* $\mathfrak{A}$.

2. $\exists x \forall y(Q(x, b) \vee P(g(x, f(y))))$. *También esta fórmula es una sentencia.*
   $(Q(x, b))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x)^2 = 3\} = \emptyset$
   $(P(g(x, f(y))))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x)(\mathfrak{a}(y)+1) \neq 0\} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 0, \ \mathfrak{a}(y) \neq 2, \ \mathfrak{a}(y) \neq 3 \ \}$
   $(Q(x, b) \vee P(g(x, f(y))))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 0, \ \mathfrak{a}(y) \neq 2, \ \mathfrak{a}(y) \neq 3 \ \}$
   Por tanto,
   $(\forall y(Q(x, b) \vee P(g(x, f(y)))))^{\mathfrak{A}} = \emptyset$
   $(\exists x \forall y(Q(x, b) \vee P(g(x, f(y)))))^{\mathfrak{A}} = \emptyset$
   Con esto tenemos que $\exists x \forall y(Q(x, b) \vee P(g(x, f(y))))$ *no es válida en* $\mathfrak{A}$.

3. $\forall x(Q(x, y) \to (x \not\approx b))$. *Esta fórmula no es una sentencia. La variable y aparece libre en la fórmula. Por tanto, debemos ver como se interpreta la fórmula dependiendo de los valores de y.*
   $(Q(x, y))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x)^2 + \mathfrak{a}(y) = 0\}$
   $(x \not\approx b)^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 2 \ \}$
   $(Q(x, y) \to (x \not\approx b))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x)^2 + \mathfrak{a}(y) \neq 0\} \cup \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 2 \ \}$
   Por tanto,
   $(\forall x(Q(x, y) \to (x \not\approx b)))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(y) \neq 1\}$
   Con esto tenemos que $\exists x \forall y(Q(x, b) \vee P(g(x, f(y))))$ *es satisfacible en* $\mathfrak{A}$ *por las asignaciones que cumplan* $\mathfrak{a}(y) \neq 1$ *y es refutable por las que cumplan* $\mathfrak{a}(y) = 1$.

4. $(x \approx f(b)) \vee P(x)$
   $(x \approx f(b))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x) = 0\}$
   $(P(x))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \ \mathfrak{a}(x) \neq 0 \ \}$

$((x \approx f(b)) \vee P(x))^{\mathfrak{A}} = \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \mathfrak{a}(x) = 0\} \cup \{\mathfrak{a} \in \mathbb{Z}_5^{\mathcal{V}} : \mathfrak{a}(x) \neq 0\} = \mathbb{Z}_5^{\mathcal{V}}$
Con esto tenemos que $(x \approx f(b)) \vee P(x)$ es válida en $\mathfrak{A}$.

Definición 5.21 (universalmente válida, satisfacible, refutable, contradicción) *Sea φ una fórmula de un lenguaje de primer orden.*

1. *φ es* universalmente válida *si φ es válida en todas las estructuras $\mathfrak{A}$.*
2. *φ es* satisfacible *si φ es satisfacible en alguna estructura $\mathfrak{A}$.*
3. *φ es* refutable *si φ es refutable en alguna estructura $\mathfrak{A}$.*
4. *φ es* contradicción *si φ es no válida en todas las estructuras $\mathfrak{A}$.*

Definición 5.22 *Sea $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\}$ un conjunto de fórmulas. Se dice que Γ es satisfacible si existe una interpretación $I = (\mathcal{E}, v)$ tal que $I^v(\gamma_1) = I^v(\gamma_2) = \cdots = I^v(\gamma_n) = 1$ (es decir, existe un modelo para todas las fórmulas de Γ).*

Un conjunto que no es satisfacible se dice insatisfacible.

Observación:

Con lo que hemos visto hasta ahora no es fácil comprobar que un conjunto de fórmulas es insatisfacible. Más adelante estudiaremos técnicas para comprobar que un conjunto de fórmulas es insatisfacible.

5.3 Implicación semántica

Definición 5.23 (implicación semántica) *Sea L un lenguaje de primer orden y $\Gamma \cup \{\varphi\} \subseteq For(L)$. Γ* implica semánticamente *φ, o φ es* consecuencia lógica *de Γ si para toda L-estructura $\mathfrak{A}$ se cumple $\Gamma^{\mathfrak{A}} \subseteq \varphi^{\mathfrak{A}}$. Este hecho lo notaremos $\Gamma \models \varphi$. En el caso de que $\Gamma = \emptyset$ escribiremos $\models \varphi$ en lugar de $\emptyset \models \varphi$.*

Observación:

Si $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\}$ y $\Gamma \models \varphi$, escribiremos $\gamma_1, \gamma_2, \cdots, \gamma_n \models \varphi$.

Pero ahora, para ver si $\gamma_1, \gamma_2, \cdots, \gamma_n \models \varphi$ no podremos hacer como en el caso de la lógica proposicional de tomar todas las asignaciones en todas las L-estructuras posibles, ya que el conjunto de tales asignaciones es infinito.

Lo que sí es cierto es que si encontramos una asignación en una L-estructura $\mathfrak{A}$ que satisfaga todas las fórmulas de Γ y no satisfaga φ, entonces podemos afirmar que $\Gamma \models \varphi$ no se cumple.

Cuando el conjunto Γ sea vacío, la expresión $\Gamma \models \varphi$ significa que φ es universalmente válida. En tal caso, escribiremos $\models \varphi$.

Teorema 5.5 . *Sea Γ un conjunto de fórmulas y φ otra fórmula. Son equivalentes:*

1. $\Gamma \models \varphi$
2. $\Gamma \cup \{\neg\varphi\}$ *es insatisfacible.*

Teorema 5.6 (Teorema de la Deducción)

Sea Γ un conjunto de fórmulas (que podría ser vacío) de un lenguaje de primer orden, y φ, ψ, otras dos fórmulas. Entonces las siguientes afirmaciones son equivalentes:

1. $\Gamma \models \varphi \to \psi$
2. $\Gamma \cup \{\varphi\} \models \psi$

La demostración es igual a la que se hizo en el caso de la lógica proposicional.

Al igual que en el tema anterior, el problema fundamental de la lógica de predicados es estudiar cuando, dado un conjunto de fórmulas Γ y una fórmula α se tiene que $\Gamma \models \alpha$. Entonces vimos que podíamos resolverlo mediante tablas de verdad o planteando ecuaciones en $\mathbb{Z}_2$. Estas dos técnicas para resolver este problema no pueden trasladarse a la lógica de predicados, así que tenemos que intentar otros métodos. Conocemos, para la lógica proposicional el algoritmo de Davis-Putnam y el método de resolución. Ambos nos servían para demostrar que un conjunto de fórmulas era insatisfacible, lo que, junto con el teorema **??** nos permitía responder a la pregunta de si $\Gamma \models \alpha$ o $\Gamma \not\models \alpha$. Ahora, con el teorema 5.5, también podemos transformar un problema de implicación semántica en un problema de estudiar si un conjunto de fórmulas es o no insatisfacible. Pero para poder aplicar estos dos métodos necesitábamos, dada una fórmula, transformarla en otra que tuviera una forma determinada (lo que en su momento llamamos la forma clausular). Estas transformaciones se conseguían a partir de unas equivalencias lógicas básicas.

Ahora vamos a necesitar algo parecido. Por tanto, en lo que sigue, vamos a extender las reglas que teníamos de equivalencias lógicas a fórmulas en las que intervienen cuantificadores.

5.4 Equivalencia lógica

Definición 5.24 (equivalencia lógica) *Dadas dos fórmulas φ y ψ de un lenguaje L. φ y ψ son lógicamente equivalentes si para toda L-estructura $\mathfrak{A}$ se cumple que $\varphi^{\mathfrak{A}} = \psi^{\mathfrak{A}}$. Es decir, si sus conjuntos de satisfacibilidad coinciden en todas las estructuras. Este hecho lo notaremos $\varphi \equiv \psi$*

5.5 Algunas equivalencias lógicas.

En una primera lectura saltar al resumen.

5.5.1 Negación y cuantificadores

5.5.2 Inclusión de $\vee$ e $\wedge$ en radio de acción de un cuantificador (I)

5.5.3 Eliminación de cuantificadores

5.5.4 Agrupación de cuantificadores

5.5.5 Inclusión de $\vee$ e $\wedge$ en radio de acción de un cuantificador (II)

5.5.6 Intercambio de cuantificadores

5.5.7 Resumen

Lógica proposicional.

1. $\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$
2. $\varphi \to \psi \equiv \neg\varphi \vee \psi$
3. $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$
4. $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$
5. $\neg\neg\varphi \equiv \varphi$
6. $\varphi \vee \varphi \equiv \varphi$
7. $\varphi \wedge \varphi \equiv \varphi$
8. $\varphi \vee \psi \equiv \psi \vee \varphi$
9. $\varphi \wedge \psi \equiv \psi \wedge \varphi$
10. $\varphi \vee (\psi \vee \chi) \equiv (\varphi \vee \psi) \vee \chi$
11. $\varphi \wedge (\psi \wedge \chi) \equiv (\varphi \wedge \psi) \wedge \chi$
12. $\varphi \wedge (\psi \vee \chi) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$
13. $\varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$

Lógica de predicados.

1. $\forall x\, \varphi \equiv \varphi$ si x no está libre en φ.
2. $\exists x\, \varphi \equiv \varphi$ si x no está libre en φ.
3. $\neg\forall x\, \varphi \equiv \exists x\, \neg\varphi$.
4. $\neg\exists x\, \varphi \equiv \forall x\, \neg\varphi$.
5. $\forall x\, \varphi \equiv \forall y\, \varphi(x|y)$ si y ni ningún cuantificador de y aparecen en φ.
6. $\exists x\, \varphi \equiv \exists y\, \varphi(x|y)$ si y ni ningún cuantificador de y aparecen en φ.
7. $\forall x\, \varphi \to \psi \equiv \exists x(\varphi \to \psi)$ si x no está libre en ψ.
8. $\exists x\varphi \to \psi \equiv \forall x(\varphi \to \psi)$ si x no está libre en ψ.
9. $\varphi \to \forall x\psi \equiv \forall x(\varphi \to \psi)$ si x no está libre en φ.
10. $\varphi \to \exists x\psi \equiv \exists x(\varphi \to \psi)$ si x no está libre en φ.
11. $\forall x\varphi \vee \psi \equiv \forall x(\varphi \vee \psi)$ si x no está libre en ψ.
12. $\forall x\varphi \wedge \psi \equiv \forall x(\varphi \wedge \psi)$ si x no está libre en ψ.
13. $\exists x\varphi \vee \psi \equiv \exists x(\varphi \vee \psi)$ si x no está libre en ψ.
14. $\exists x\varphi \wedge \psi \equiv \exists x(\varphi \wedge \psi)$ si x no está libre en ψ.
15. $\forall x\varphi \to \exists x\psi \equiv \exists x(\varphi \to \psi)$.
16. $\forall x\varphi \wedge \forall x\psi \equiv \forall x(\varphi \wedge \psi)$.
17. $\exists x\varphi \vee \exists x\psi \equiv \exists x(\varphi \vee \psi)$.
18. $\forall x\forall y\, \varphi \equiv \forall y\forall x\, \varphi$.
19. $\exists x\exists y\, \varphi \equiv \exists y\exists x\, \varphi$.

5.6 Formas normales.

Dado un conjunto de fórmulas, vamos a aplicarles ciertas transformaciones de manera que éstas no alteren el carácter de satisfacibilidad o insatisfacibilidad del conjunto considerado y con la ventaja adicional de que las transformaciones darán lugar a fórmulas más "simples" que las del conjunto de partida.

5.6.1 Forma normal prenexa

Definición 5.25 (forma normal prenexa) *Una fórmula φ es una **forma normal prenexa** si es de la forma*

$$\varphi = C_1 x_1 \ldots C_n x_n \Phi$$

donde $C_i = \forall$ ó $\exists$ y Φ es una fórmula sin cuantificadores.

Teorema 5.7 *Para toda fórmula φ existe otra fórmula φ^* en forma normal prenexa lógicamente equivalente a φ.*

La demostración se hace aplicando las propiedades de la chuleta oficial.

Ejemplo 5.25

1. $\forall x S(x) \to \exists z \forall y R(z,y)$
Cuatro posibilidades:

(a) $\exists x(S(x) \to \exists z \forall y R(z,y))$
$\exists x \exists z(S(x) \to \forall y R(z,y))$
$\exists x \exists z \forall y(S(x) \to R(z,y))$

(b) $\exists z(\forall x S(x) \to \forall y R(z,y))$
$\exists z \exists x(S(x) \to \forall y R(z,y))$
$\exists z \exists x \forall y(S(x) \to R(z,y))$

(c) $\exists z(\forall x S(x) \to \forall y R(z,y))$
$\exists z \forall y(\forall x S(x) \to R(z,y))$
$\exists z \forall y \exists x(S(x) \to R(z,y))$

(d) $\forall x S(x) \to \exists x \forall y R(x,y)$
$\exists x(S(x) \to \forall y R(x,y))$
$\exists x \forall y(S(x) \to R(x,y))$

2. $\forall x(R(x,y) \wedge \neg \forall y R(x,y))$
$\forall x(R(x,y) \wedge \exists y \neg R(x,y))$
$\forall x(R(x,y) \wedge \exists z \neg R(x,z))$
$\forall x \exists z(R(x,y) \wedge \neg R(x,z))$

3. $\exists x R(x,y) \vee (\forall x S(x) \wedge \neg \exists z R(a,z))$
$\exists x R(x,y) \vee (\forall x S(x) \wedge \forall z \neg R(a,z))$
$\exists x R(x,y) \vee (\forall x S(x) \wedge \forall x \neg R(a,x))$
$\exists x R(x,y) \vee \forall x(S(x) \wedge \neg R(a,x))$
$\exists z R(z,y) \vee \forall x(S(x) \wedge \neg R(a,x))$
$\exists z(R(z,y) \vee \forall x(S(x) \wedge \neg R(a,x)))$
$\exists z \forall x(R(z,y) \vee (S(x) \wedge \neg R(a,x)))$

4. $\forall x \forall z(\forall z P(x,z) \wedge \forall x P(x,z)) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$(\forall x \forall z \forall z P(x,z) \wedge \forall x \forall z \forall x P(x,z)) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$(\forall x \forall z P(x,z) \wedge \forall x \forall z P(x,z)) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$\forall x \forall z P(x,z) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$\forall x \forall z P(x,z) \to (\forall x \exists y P(x,y) \wedge \forall x Q(x))$
$\forall x \forall z P(x,z) \to \forall x \exists y(P(x,y) \wedge Q(x))$
$\forall x \forall z P(x,z) \to \forall y \exists z(P(y,z) \wedge Q(y))$
$\exists x(\forall z P(x,z) \to \forall y \exists z(P(y,z) \wedge Q(y)))$
$\exists x \forall y(\forall z P(x,z) \to \exists z(P(y,z) \wedge Q(y)))$
$\exists x \forall y \exists z(P(x,z) \to (P(y,z) \wedge Q(y)))$

Teorema 5.8 *Dado un conjunto de fórmulas* Γ*, considérese el conjunto* Γ^* *formado al escoger una fórmula en forma normal prenexa para cada fórmula de* Γ*. Entonces* Γ *es insatisfacible si y sólo si* Γ^* *es insatisfacible.*

5.6.2 Forma normal de Skolem

Definición 5.26 (forma normal de Skolem) *Una fórmula* φ *es una* ***forma normal de Skolem*** *si es de la forma*

$$\varphi = \forall x_1 \forall x_2 \ldots \forall x_n \Phi$$

donde Φ *es una fórmula sin cuantificadores.*

Partimos ya de una fórmula φ en forma normal prenexa. A grosso modo una forma normal de Skolem para φ será obtenida reemplazando las variables cuantificadas existencialmente en φ por ciertos términos apropiados y “eliminando” los cuantificadores existenciales que aparecen en φ.

Dada $\varphi \in Form(L)$ en forma normal prenexa, una forma normal de Skolem de φ es una fórmula obtenida al ir substituyendo cada variable x_i cuantificada existencialmente por $f(x_{i_1}, \ldots, x_{i_m})$, donde f es un símbolo de función con ariedad m que no aparece en el lenguaje que estemos utilizando, ni ha sido usado hasta el momento para la substitución de otra variable, y $x_{i_1}, \ldots, x_{i_m}$ son las variables cuantificadas universalmente que preceden a $\exists x_i$ en la escritura de φ, de no haber ninguna, la substitución se hace por un símbolo de constante que no aparezca en el lenguaje, ni haya sido utilizado para la substitución de otra variable.

Además cuando transformemos fórmulas que ya están en forma normal prenexa a forma normal de Skolem, los nuevos símbolos introducidos para cada una de las fórmulas (ya sean de constante o de función) no pueden aparecer ni pueden haber sido utilizados en ninguna de las otras fórmulas dadas.

Al "skolemizar" de esta forma perdemos la equivalencia lógica, pero la satisfacibilidad, que es lo que nos interesa ahora, se conserva.

Teorema 5.9 *Sea Γ^* un conjunto de fórmulas en forma normal prenexa. Tomamos una forma normal de Skolem de cada elemento de Γ^*, y obtenemos el conjunto Γ^{**}. Entonces Γ^* es insatisfacible si, y sólo si, Γ^{**} es insatisfacible.*

Ejemplo 5.26

1. $\exists x \forall y(\neg S(x) \vee R(x, y))$
   $\forall y(\neg S(a) \vee R(a, y))$
2. $\forall x \exists z(R(x, y) \wedge \neg R(x, z))$
   $\forall x(R(x, y) \wedge \neg R(x, f(x)))$
3. $\forall x \exists y \exists z R(x, y, z)$
   $\forall x \exists z R(x, f(x), z)$
   $\forall x R(x, f(x), g(x))$
4. $\forall x \exists y \forall z \exists u(R(x, y, u) \vee S(y, f(z)))$
   $\forall x \forall z \exists u(R(x, g(x), u) \vee S(g(x), f(z)))$
   $\forall x \forall z(R(x, g(x), h(x, z)) \vee S(g(x), f(z)))$
5. $\forall x \forall y \exists u(R(x, y, u) \vee S(y, f(u)))$
   $\forall x \forall y(R(x, y, g(x, y)) \vee S(y, f(g(x, y))))$

Ejemplo 5.27

1. $\forall x S(x) \to \exists z \forall y R(z, y)$
   Cuatro posibilidades:
   (a) $\exists x(S(x) \to \exists z \forall y R(z, y))$
       $\exists x \exists z(S(x) \to \forall y R(z, y))$
       $\exists x \exists z \forall y(S(x) \to R(z, y))$ *Forma prenexa*
       $\exists z \forall y(S(a) \to R(z, y))$
       $\forall y(S(a) \to R(b, y))$ *Forma de Skolem*
   (b) $\exists z(\forall x S(x) \to \forall y R(z, y))$
       $\exists z \exists x(S(x) \to \forall y R(z, y))$
       $\exists z \exists x \forall y(S(x) \to R(z, y))$ *Forma prenexa*
       $\exists x \forall y(S(x) \to R(a, y))$
       $\forall y(S(b) \to R(a, y))$ *Forma de Skolem*
   (c) $\exists z(\forall x S(x) \to \forall y R(z, y))$
       $\exists z \forall y(\forall x S(x) \to R(z, y))$
       $\exists z \forall y \exists x(S(x) \to R(z, y))$ *Forma prenexa*
       $\forall y \exists x(S(x) \to R(a, y))$
       $\forall y(S(f(y)) \to R(a, y))$ *Forma de Skolem*

(d) $\forall x S(x) \to \exists x \forall y R(x,y)$
$\exists x(S(x) \to \forall y R(x,y))$
$\exists x \forall y(S(x) \to R(x,y))$ *Forma prenexa*
$\forall y(S(a) \to R(a,y))$ *Forma de Skolem*

2. $\forall x(R(x,y) \wedge \neg\forall y R(x,y)$
$\forall x(R(x,y) \wedge \exists y \neg R(x,y)$
$\forall x(R(x,y) \wedge \exists z \neg R(x,z)$
$\forall x \exists z(R(x,y) \wedge \neg R(x,z)$ *Forma prenexa*
$\forall x(R(x,y) \wedge \neg R(x,f(x))$ *Forma de Skolem*

3. $\exists x R(x,y) \vee (\forall x S(x) \wedge \neg\exists z R(a,z))$
$\exists x R(x,y) \vee (\forall x S(x) \wedge \forall z \neg R(a,z))$
$\exists x R(x,y) \vee (\forall x S(x) \wedge \forall x \neg R(a,x))$
$\exists x(R(x,y) \vee \forall x(S(x) \wedge \neg R(a,x)))$
$\exists z(R(z,y) \vee \forall x(S(x) \wedge \neg R(a,x)))$
$\exists z \forall x(R(z,y) \vee (S(x) \wedge \neg R(a,x)))$ *Forma prenexa*
$\forall x(R(b,y) \vee (S(x) \wedge \neg R(a,x)))$ *Forma de Skolem*

4. $\forall x \forall z(\forall z P(x,z) \wedge \forall x P(x,z)) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$(\forall x \forall z \forall z P(x,z) \wedge \forall x \forall z \forall x P(x,z)) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$(\forall x \forall z P(x,z) \wedge \forall x \forall z P(x,z)) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$\forall x \forall z P(x,z) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$\forall x \forall z P(x,z) \to (\forall x \exists y P(x,y) \wedge \forall x Q(x))$
$\forall x \forall z P(x,z) \to \forall x \exists y(P(x,y) \wedge Q(x))$
$\forall x \forall z P(x,z) \to \forall y \exists z(P(y,z) \wedge Q(y))$
$\exists x(\forall z P(x,z) \to \forall y \exists z(P(y,z) \wedge Q(y)))$
$\exists x \forall y(\forall z P(x,z) \to \exists z(P(y,z) \wedge Q(y)))$
$\exists x \forall y \exists z(P(x,z) \to (P(y,z) \wedge Q(y)))$ *Forma prenexa*
$\forall y \exists z(P(a,z) \to (P(y,z) \wedge Q(y)))$
$\forall y(P(a,f(y)) \to (P(y,f(y)) \wedge Q(y)))$ *Forma de Skolem*

5.6.3 Forma clausulada

Nos restringirenos a fórmulas que sean sentencias (recuérdese que una fórmula es sentencia si todas las ocurrencias de sus variables son ligadas). Así pues, vamos a partir sentencias ya en forma de Skolem y vamos a escribirlas como una conjunción de cláusulas.

Definiciones 5.27 (literal, cierre universal, cláusula)

1. *Un **literal** es una fórmula atómica o la negación de una fórmula atómica.*
2. *El **cierre universal** de una fórmula $\varphi(x_1,\ldots,x_n)$ cuyas variables con ocurrencias libres son $x_1,\ldots,x_n$, es*

$$\forall x_1 \cdots \forall x_n \varphi(x_1,\ldots,x_n).$$

3. *Una **cláusula** es el cierre universal de una disyunción de literales en los que no aparece un literal y su contrario.*

Así pues, si C es una cláusula, entonces C es de la forma $\forall x_1 \cdots \forall x_n(L_1 \vee \ldots \vee L_m)$, con L_i literales. De ahora en adelante al escribir las cláusulas omitiremos los cuantificadores universales que las preceden, escribiendo de esta forma $L_1 \vee \ldots \vee L_m$ en vez de C, sin olvidarnos de que todas las variables están cuantificadas universalmente.

Ejemplo 5.28 *Son literales las siguientes fórmulas:*

1. $P(a)$.
2. $\neg Q(x,a)$.
3. $\neg H(x, g(x,a), y)$.
4. $P(f(x))$.
5. $Q(g(y, f(b)), x)$.

Los contrarios de estos literales son:

1. $\neg P(a)$.
2. $Q(x,a)$.
3. $H(x, g(x,a), y)$.
4. $\neg P(f(x))$.
5. $\neg Q(g(y, f(b)), x)$.

Ejemplo 5.29

1. *El cierre universal de* $P(a)$ *es ella misma.*
2. *El cierre universal de* $\neg H(x, g(x,a), y)$ *es* $\forall x \forall y \neg H(x, g(x,a), y)$.
3. *El cierre universal de* $P(a) \vee Q(x, g(y,b)) \vee \neg H(x, g(x,a), y)$ *es* $\forall x \forall y (P(a) \vee Q(x, g(y,b)) \vee \neg H(x, g(x,a), y))$.

Ejemplo 5.30 *Son ejemplos de cláusulas:*

1. $P(a)$.
2. $\forall x \forall y (P(a) \vee Q(x, g(y,b)) \vee \neg H(x, g(x,a), y))$.
3. $\forall x \forall y (x \not\approx y)$.
4. $\forall x \forall y \neg H(x, g(x,a), y)$.

Dada σ una sentencia, sea σ^* una sentencia equivalente a ella en forma prenexa. Sea σ^{**} una forma de Skolem asociada a σ^*. Podemos encontrar cláusulas $C_1, \ldots, C_m$ verificando

$$\sigma^{**} \equiv \bigwedge_{i=1}^{m} C_i.$$

A esta conjunción de cláusulas la llamamos una **forma clausulada** de σ.

La obtención de la forma clausulada a partir de una forma de Skolem de una sentencia se realiza olvidándose de los cuantificadores y trabajando con la fórmula sin cuantificadores como si fuese una fórmula proposicional (donde el lugar de las proposiciones atómicas lo ocupan fórmulas atómicas). Al final del proceso se distribuyen los cuantificadores universales sobre la $\wedge$ y tenemos la forma clausulada.

Ejemplo 5.31

1. $\forall x S(x) \to \exists z \forall y R(z,y)$
   Cuatro posibilidades:
   (a) $\exists x (S(x) \to \exists z \forall y R(z,y))$
   $\exists x \exists z (S(x) \to \forall y R(z,y))$
   $\exists x \exists z \forall y (S(x) \to R(z,y))$ *Forma prenexa*
   $\exists z \forall y (S(a) \to R(z,y))$
   $\forall y (S(a) \to R(b,y))$ *Forma de Skolem*
   $\forall y (\neg S(a) \vee R(b,y))$ *Forma clausulada (1 cláusula)*
   (b) $\exists z (\forall x S(x) \to \forall y R(z,y))$
   $\exists z \exists x (S(x) \to \forall y R(z,y))$
   $\exists z \exists x \forall y (S(x) \to R(z,y))$ *Forma prenexa*
   $\exists x \forall y (S(x) \to R(a,y))$
   $\forall y (S(b) \to R(a,y))$ *Forma de Skolem*
   $\forall y (\neg S(b) \vee R(a,y))$ *Forma clausulada (1 cláusula)*
   (c) $\exists z (\forall x S(x) \to \forall y R(z,y))$
   $\exists z \forall y (\forall x S(x) \to R(z,y))$
   $\exists z \forall y \exists x (S(x) \to R(z,y))$ *Forma prenexa*
   $\forall y \exists x (S(x) \to R(a,y))$
   $\forall y (S(f(y)) \to R(a,y))$ *Forma de Skolem*
   $\forall y (\neg S(f(y)) \vee R(a,y))$ *Forma clausulada (1 cláusula)*

(d) $\forall x S(x) \to \exists x \forall y R(x,y)$
$\exists x(S(x) \to \forall y R(x,y))$
$\exists x \forall y(S(x) \to R(x,y))$ *Forma prenexa*
$\forall y(S(a) \to R(a,y))$ *Forma de Skolem*
$\forall y(\neg S(a) \vee R(a,y))$ *Forma clausulada (1 cláusula)*

2. $\forall x \forall z(\forall z P(x,z) \wedge \forall x P(x,z)) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$(\forall x \forall z \forall z P(x,z) \wedge \forall x \forall z \forall x P(x,z)) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$(\forall x \forall z P(x,z) \wedge \forall x \forall z P(x,z)) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$\forall x \forall z P(x,z) \to \forall x(\exists y P(x,y) \wedge \forall x Q(x))$
$\forall x \forall z P(x,z) \to (\forall x \exists y P(x,y) \wedge \forall x Q(x))$
$\forall x \forall z P(x,z) \to \forall x \exists y(P(x,y) \wedge Q(x))$
$\forall x \forall z P(x,z) \to \forall y \exists z(P(y,z) \wedge Q(y))$
$\exists x(\forall z P(x,z) \to \forall y \exists z(P(y,z) \wedge Q(y)))$
$\exists x \forall y(\forall z P(x,z) \to \exists z(P(y,z) \wedge Q(y)))$
$\exists x \forall y \exists z(P(x,z) \to (P(y,z) \wedge Q(y)))$ *Forma prenexa*
$\forall y \exists z(P(a,z) \to (P(y,z) \wedge Q(y)))$
$\forall y(P(a,f(y)) \to (P(y,f(y)) \wedge Q(y)))$ *Forma de Skolem*
$\forall y(\neg P(a,f(y)) \vee (P(y,f(y)) \wedge Q(y)))$
$\forall y((\neg P(a,f(y)) \vee P(y,f(y))) \wedge (\neg P(a,f(y)) \vee Q(y)))$
$\forall y(\neg P(a,f(y)) \vee P(y,f(y))) \wedge \forall y(\neg P(a,f(y)) \vee Q(y))$ *Forma clausulada (2 cláusulas)*

Teorema 5.10 *Dado un conjunto de sentencias Σ. Supongamos que para $\sigma_i \in \Sigma$ su forma clausulada es $C_{i1} \wedge C_{i2,} \wedge \ldots \wedge C_{im_i}$. Sea Σ^{**} el conjunto formado por todas las cláusulas que aparecen en la forma clausulada de cada una de las sentencias de Σ, es decir,*

$$\Sigma^{**} = \{C_{11}, C_{12}, \ldots, C_{1m_1}, C_{21}, C_{22}, \ldots, C_{2m_2}, \ldots, C_{n1}, C_{n2}, \ldots, C_{nm_n}\}.$$

Entonces Σ es insatisfacible si, y sólo si, Σ^{**} es insatisfacible.

Ejemplo 5.32 *Supongamos que tenemos las sentencias:*
$\sigma_1 = \forall x(P(x) \wedge \neg Q(x) \to \exists y(R(x,y) \wedge S(y)))$
$\sigma_2 = \exists y \forall x((R(y,x) \to T(x)) \wedge T(y) \wedge P(y))$
$\sigma_3 = \exists x(T(x) \wedge (Q(x) \vee S(x)))$
Queremos saber si
$\{\sigma_1, \sigma_2\} \models \sigma_3$
Este problema se reduce a saber si el conjunto $\{\sigma_1, \sigma_2, \neg\sigma_3\}$ es insatisfacible o satisfacible.
Calculamos una forma clausulada de σ_1, σ_2 y $\neg\sigma_3$

- $\forall x(P(x) \wedge \neg Q(x) \to \exists y(R(x,y) \wedge S(y)))$
  $\forall x \exists y(P(x) \wedge \neg Q(x) \to (R(x,y) \wedge S(y)))$ *FNP*
  $\forall x(P(x) \wedge \neg Q(x) \to (R(x,f(x)) \wedge S(f(x))))$ *FNS*
  $\forall x(\neg(P(x) \wedge \neg Q(x)) \vee (R(x,f(x)) \wedge S(f(x))))$
  $\forall x(\neg P(x) \vee Q(x) \vee (R(x,f(x)) \wedge S(f(x))))$
  $\forall x((\neg P(x) \vee Q(x) \vee R(x,f(x)) \wedge (\neg P(x) \vee Q(x) \vee S(f(x))))$
  $\forall x(\neg P(x) \vee Q(x) \vee R(x,f(x)) \wedge \forall x(\neg P(x) \vee Q(x) \vee S(f(x)))$ *FC*

- $\exists y \forall x((R(y,x) \to T(x)) \wedge T(y) \wedge P(y))$ *FNP*
  $\forall x((R(a,x) \to T(x)) \wedge T(a) \wedge P(a))$ *FNS*
  $\forall x((\neg R(a,x) \vee T(x)) \wedge T(a) \wedge P(a))$
  $\forall x(\neg R(a,x) \vee T(x)) \wedge T(a) \wedge P(a)$ *FC*

- $\neg\exists x(T(x) \wedge (Q(x) \vee S(x)))$
  $\forall x\neg(T(x) \wedge (Q(x) \vee S(x)))$ *FNP y FNS*
  $\forall x(\neg T(x) \vee \neg(Q(x) \vee S(x)))$
  $\forall x(\neg T(x) \vee (\neg Q(x) \wedge \neg S(x)))$
  $\forall x((\neg T(x) \vee \neg Q(x)) \wedge (\neg T(x) \vee \neg S(x)))$
  $\forall x(\neg T(x) \vee \neg Q(x)) \wedge \forall x(\neg T(x) \vee \neg S(x))$ *FC (2 cláusulas)*

El problema queda reducido a estudiar la insatisfacibilidad o satisfacibilidad del conjunto de cláusulas:

$$\Sigma = \{\neg P(x)\vee Q(x)\vee R(x,f(x)),\ \neg P(x)\vee Q(x)\vee S(f(x)),\ \neg R(a,x)\vee T(x),\ T(a),\ P(a),\ \neg T(x)\vee\neg Q(x),\ \neg T(x)\vee\neg S(x)\}$$