Lógica Computacional
Notas para el curso Análisis Lógico

Mat. Favio Ezequiel Miranda Perea

Contenido

Capítulo 1

Sintáxis de la Lógica de Predicados

En este capítulo estudiaremos los preliminares sintácticos necesarios para el resto del libro, sólamente se supone un conocimiento de la lógica proposicional, tema que no trataremos aquí.

1.1 Conceptos Básicos

Definición 1.1 una *signatura* o *tipo de semejanza* es un conjunto Σ de símbolos de alguna de las siguientes formas:

- Símbolos de función o letras funcionales: $f_1^{(n_1)}, \ldots, f_k^{(n_k)}, \ldots$
- Símbolos de predicado o letras predicativas: $P_1^{(n_1)}, \ldots, P_k^{(n_k)}, \ldots$
- Símbolos de constante: $c_1, \ldots, c_k \ldots$

Cada símbolo tiene asociado un número natural, denotado entre paréntesis, que es su aridad o número de argumentos. Las constantes pueden considerarse como símbolos de aridad cero.

Las signaturas sirven para simbolizar y traducir algun enunciado en lenguaje natural, en nuestro caso el español, a un lenguaje formal como el que definimos enseguida.

Definición 1.2 Sea Σ una signatura. El *lenguaje formal de primer orden para* Σ, denotado $\mathcal{L}_\Sigma$, es un conjunto que consta de los siguientes conjuntos de símbolos:

- Símbolos no lógicos, que son los símbolos de Σ.
- Símbolos lógicos:
  - Conectivos: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
  - Cuantificadores: $\forall$, $\exists$.
- Variables: $VAR = \{x_1, \ldots, x_n, \ldots\}$
- Símbolos Auxiliares: paréntesis (,) y coma ,.

Si además $\mathcal{L}_\Sigma$ tiene el símbolo de igualdad $=$ decimos que el lenguaje es con igualdad. Nosotros utilizaremos lenguajes sin igualdad, a menos que se especifique lo contrario.

Los puros símbolos no bastan para poder traducir del español al lenguaje formal, para esto necesitamos combinar los diferentes símbolos de manera adecuada, formando términos y fórmulas. Los términos son cadenas de símbolos del lenguaje que representan individuos.

Definición 1.3 Sea Σ una signatura. Definimos recursivamente el conjunto de Σ-términos, denotado $TERM(\Sigma)$, como sigue:

- Los símbolos de variable son Σ-términos, es decir, si $x \in VAR$ entonces $x \in TERM(\Sigma)$.
- Los símbolos de constante son Σ-términos, es decir, si $c_k \in \Sigma$ entonces $c_k \in TERM(\Sigma)$.
- Si $t_1, \ldots, t_n$ son términos y $f \in \Sigma$ es un símbolo de función n-ario entonces $f(t_1, \ldots, t_n)$ es un término.
- Son todos.

las variables y constantes se llaman *términos simples* mientras que los símbolos de función aplicados a términos son los *términos compuestos*

Definición 1.4 Un Σ-término t es *cerrado* si en t no figuran variables. El conjunto de términos cerrados se denota con $TERM_0(\Sigma)$.

Finalmente, para poder lograr una traducción del lenguaje natural, presentamos las fórmulas, que son cadenas de símbolos que sirven para representar enunciados del español. Iniciamos con las fórmulas más simples que expresan propiedades y relaciones entre objetos.

Definición 1.5 Sean Σ una signatura y $t_1, \ldots, t_n \in TERM(\Sigma)$. Las *fórmulas atómicas* son cadenas de símbolos de $\mathcal{L}_\Sigma$ de la siguiente forma:

- $P(t_1, \ldots, t_n)$, donde $P \in \Sigma$ es un símbolo de predicado n-ario.
- Si el lenguaje es con igualdad entonces agregamos $t_i = t_j$.

El conjunto de fórmulas atómicas se denota $ATOM(\Sigma)$.

Por último definimos las fórmulas de una signatura.

Definición 1.6 Sea Σ una signatura. Definimos las Σ-fórmulas, recursivamente, como sigue:

- Las fórmulas atómicas son fórmulas.
- Si φ, ψ son fórmulas y $x \in VAR$ entonces $\neg\varphi,\ \varphi \vee \psi,\ \varphi \wedge \psi,\ \varphi \to \psi,\ \varphi \leftrightarrow \psi,\ \forall x\varphi,\ \exists x\varphi$ son fórmulas.
- Son todas.

Para evitar la presencia excesiva de paréntesis definimos el siguiente orden de precedencia entre los conectivos.

$\neg$ (Precedencia Mayor).

$\wedge,\ \vee$.

$\to$.

$\leftrightarrow$ (Precedencia Menor).

Así por ejemplo, la fórmula $p \to q \vee r \leftrightarrow s$ debe entenderse como $(p \to (q \vee r)) \leftrightarrow s$ y la fórmula $\neg\forall xPx \vee Qy \to \exists yRy$ como $(\neg(\forall xPx) \vee Qy) \to (\exists yRy)$.

Ejercicios

Sean $\Sigma_1 = \{P^{(2)}, Q^{(3)}, R^{(1)}, f^{(2)}, g^{(1)}, c, d\}$ y $\Sigma_2 = \{Q^{(2)}, P^{(3)}, R^{(1)}, g^{(2)}, f^{(3)}, d, e\}$.

1.1.1.- Dar 5 ejemplos de Σ_1-términos compuestos.

1.1.2.- Dar 5 ejemplos de Σ_2-términos compuestos.

1.1.3.- ¿ Cuales de los siguientes son Σ_1-términos o Σ_2-términos?

(a) $P(x, f(c, w))$.
(b) $g(f(x, d))$.
(c) $f(e, g(x, y), d)$.
(d) $g(d, f(c, w))$.
(e) $f(x, w, g(d, f(w, e, v))$.
(f) $f(g(y), f(c, d))$.
(g) $g(f(x, g(w)))$.
(h) $Q(d, f(d, e, y))$.
(i) $R(f(c, x))$.
(j) $P(e, g(x, d), f(z, g(e, w), w))$.

1.1.4.- Dar 5 ejemplos de Σ_1-fórmulas atómicas.

1.1.5.- Dar 5 ejemplos de Σ_2-fórmulas atómicas.

1.1.6.- ¿ Cuales de las siguientes son Σ_1-fórmulas o Σ_2-fórmulas?

(a) $Q(f(c, y), g(w), R(z))$.
(b) $\forall x P(x, Q(c, d, y))$.
(c) $\forall z Q(c, f(x, d), w) \to R(y)$.
(d) $R(g(v, d)) \lor \neg Q(x, e)$.
(e) $\neg R(d) \leftrightarrow \exists y P(f(c, x), y)$.
(f) $\neg Q(g(e, x), y) \land f(x, y, d)$.
(g) $\neg\neg P(f(x, y, w), g(w, v), d) \land Q(d, e)$.

(h) $\exists y \neg Q(y,y) \to R(w) \wedge P(x,y,e)$.

(i) $\neg(P(x,c) \vee R(g(c,w)))$.

(j) $\forall x \exists y \forall w (Q(x,w) \neg R(y) \to P(w,w))$.

1.2 Correspondencia con el lenguaje natural

En esta sección vamos a traducir enunciados del español al lenguaje formal, para lo cual necesitamos darle significado a los símbolos del lenguaje. Debe quedar claro que aquí solamente presentaremos algunos ejemplos y consejos para el proceso de traducción, el cual depende en gran medida de la habilidad del alumno. Vamos a analizar cada conectivo y cuantificador y su significado.

1.2.1 La Negación

La *negación* de la fórmula φ es la fórmula $\neg\varphi$.

Símbolo utilizado: $\neg$

Correspondencia con el español: No, no es cierto que , es falso que, etc.

Otros símbolos: $\sim \varphi$, $\overline{\varphi}$.

1.2.2 La Disyunción

La *disyunción* de las fórmulas φ, ψ es la fórmula $\varphi \vee \psi$. Las fórmulas φ, ψ se llaman *disyuntos*.

Símbolo utilizado: $\vee$

Correspondencia con el español: o.

Otros símbolos: $\varphi + \psi$, $\varphi | \psi$.

1.2.3 La Conjunción

La *conjunción* de las fórmulas φ, ψ es la fórmula $\varphi \wedge \psi$. Las fórmulas φ, ψ se llaman *conyuntos*.

Símbolo utilizado: $\wedge$

Correspondencia con el español: y, pero.

Otros símbolos: $\varphi \& \psi$, $\varphi \cdot \psi$.

1.2.4 La Implicación

La *implicación* o *condicional* de las fórmulas φ, ψ es la fórmula $\varphi \to \psi$. Las fórmulas φ es el *antecedente* y la fórmula ψ es *consecuente* de la implicación.

Símbolo utilizado: $\to$

Correspondencia con el español: $\varphi \to \psi$ significa: si φ entonces ψ; ψ, si φ; φ es condición suficiente para ψ; ψ es condición necesaria para φ, etc.

Otros símbolos: $\varphi \Rightarrow \psi$, $\varphi \supset \psi$.

1.2.5 La Equivalencia

La *equivalencia* o *bicondicional* de las fórmulas φ, ψ es la fórmula $\varphi \leftrightarrow \psi$.

Símbolo utilizado: $\leftrightarrow$

Correspondencia con el español: φ es equivalente a ψ; φ si y sólo si ψ; φ es condición necesaria y suficiente para ψ, etc.

Otros símbolos: $\varphi \Leftrightarrow \psi$, $\varphi \equiv \psi$.

1.2.6 La Cuantificación Universal

Una *cuantificación universal* de la fórmula φ es una fórmula del tipo $\forall x \varphi$, donde x es una variable, llamada la *variable del cuantificador*.

Símbolo utilizado: $\forall$

Correspondencia con el español: para todos, para cualquier, todos, cualquiera, etc.

Otros símbolos: $(x)\varphi$.

1.2.7 La Cuantificación Existencial

Una *cuantificación existencial* de la fórmula φ es una fórmula del tipo $\exists x \varphi$, donde x es una variable, llamada la *variable del cuantificador*.

Símbolo utilizado: $\exists$

Correspondencia con el español: para algún, existe un, existe algun, algun, etc.

Otros símbolos: $(Ex)\varphi$.

1.2.8 Algunos ejemplos de traducción

En esta sección vamos a dar algunos ejemplos de traducción, como ya se dijo, la forma de traducir no es única y dependerá en gran medida de la habilidad de cada estudiante.
Vamos a iniciar mostrando la traducción de las proposiciones categoricas de la lógica aristotélica, que sirven de base a traducciones más complicadas. Las proposiciones categóricas corresponden a alguno de los siguientes esquemas:

UA Todo S es P.

EA Algún S es P.

UN Ningún S es P.

EN Algún S no es P.

Para el proceso de traducción necesitamos la signatura $\Sigma = \{S, P\}$ donde los símbolos S y P son símbolos unarios de predicado.

UA Está proposición corresponde al juicio universal afirmativo y se traduce con la fórmula $\forall x\ (S(x) \to P(x))$.

EA Corresponde al juicio existencial afirmativo y se traduce mediante la fórmula $\exists x\ (S(x) \wedge P(x))$.

UN Corresponde al juicio universal negativo, la palabra “ningún” es una contracción de “no algún” y se traduce con $\forall x\ (S(x) \to \neg P(x))$ o equivalentemente con $\neg \exists x\ (S(x) \wedge P(x))$.

EN Esta proposición corresponde al juicio existencial afirmativo y se traduce con la fórmula $\exists x\ (S(x) \land \neg P(x))$.

Es común cometer errores en alguno de estos casos al traducir, si bien no hay una regla formal, es bueno recordar que los juicios universales se escriben con implicación y los existenciales con conjunción. Lo cual no quiere decir que no pueda aparecer un cuantificador universal con conjunción o un existencial con implicación, sino que no es usual, sobre todo el último caso. Como ejemplo obsérvese el caso de las fórmulas $\forall x(G(x) \to M(x))$ y $\forall x(G(x) \land M(x))$; si $G(x)$ significa "x es gato" y $M(x)$ significa "x maulla" entonces la primera fórmula traduce a la afirmación "todos los gatos maullan" mientras que la segunda nos dice algo mucho más fuerte que es "todos son gatos y maullan". El caso del existencial es más complicado una fórmula de la forma $\exists x\ (S(x) \to P(x))$ aunque sintácticamente es correcta, tiene problemas semánticos que veremos más adelante, así que siempre que se quiera traducir algo con un existencial debemos usar una conjunción y nunca una implicación.

El proceso de traducción también depende de la signatura elegida, pero lo más funcional es descomponer la oración en español en frases más simples que correspondan a fórmulas atómicas y traducir a partir de ellas. Pasemos a los ejemplos.

Ejemplo 1.1 Formar los cuatro juicios aristotelicos posibles con las propiedades "ser minotauro" y "ser troyano".

Elegimos la signatura $\Sigma = \{M, T\}$, donde el significado del símbolo M es "ser minotauro" y el de T es "ser troyano", siempre es adecuado elegir símbolos que recuerden a las propiedades o relaciones que se quieren representar. Los juicios son:

- Todos los minotauros son troyanos, $\forall x\ (M(x) \to T(x))$.
- Algún minotauro es troyano, $\exists x\ (M(x) \land T(x))$.
- Ningún minotauro es troyano, $\forall x\ (M(x) \to \neg T(x))$.
- Algún minotauro no es troyano, $\exists x\ (M(x) \land \neg T(x))$.

Ejemplo 1.2 *Hay una lanza que perfora a todos los escudos.* En este caso tenemos dos propiedades, "ser lanza", "ser escudo" y la relación binaria "perforar". La signatura es $\Sigma = \{L^{(1)}, E^{(1)}, P^{(2)}\}$.

La traducción es $\exists x\ (L(x) \land \forall y\ (E(y) \to P(x, y)))$.

Ejemplo 1.3 *Hay un escudo al cual ninguna lanza perfora.* Usamos la misma signatura del ejemplo anterior.
La traducción es: $\exists x\ (E(x) \wedge \forall y\ (L(y) \rightarrow \neg P(y, x)))$

Ejemplo 1.4 Considérese la siguiente signatura $\Sigma = \{L^{(2)},\ E^{(2)},\ A^{(1)},\ F^{(1)},\ M^{(1)},\ g, q, c, a, h, d\}$, con el siguiente significado:

$L(x, y)$	"x es más largo que y"
$E(x, y)$	"x es escrito por y"
$A(x)$	"x está escrito en alemán"
$M(x)$	"x es matemático"
$F(x)$	"x es filósofo"
g	"Grundlehren der Mathematik"
q	"El Quijote de la Mancha"
c	"La construcción lógica del mundo"
a	"Alicia en el país de las maravillas"
h	David Hilbert
d	Charles Dogdson.

Tenemos las siguientes traducciones.

- *"Grundlehren der Mathematik" es un libro escrito en alemán.*
$$A(g).$$
- *Hay un libro más largo que "El Quijote de la Mancha".*
$$\exists x L(x, q)$$
- *"La construcción lógica del mundo" es un libro escrito por un filósofo y no es mas largo que "Alicia en el país de las maravillas"*
$$\exists x(F(x) \wedge E(c, x)) \wedge \neg L(c, a)$$
- *"Alicia en el país de las maravillas" es un libro escrito por un matemático.*
$$\exists x(M(x) \wedge E(a, x))$$
- *Hay un libro escrito en alemán por un matemático que es más largo que "Grundlehren der Mathematik".*
$$\exists y(A(y) \wedge \exists x\ (M(x) \wedge E(y, x)) \wedge L(y, g))$$

- *Si "Alicia en el país de las maravillas" es un libro escrito por un matemático entonces es un libro escrito por un filósofo.*

$$\exists x(M(x) \wedge E(a,x)) \to \exists y(F(y) \wedge E(a,y))$$

- *No hay un libro escrito en alemán por un filósofo.*

$$\neg\exists x(A(x) \wedge \exists y(F(y) \wedge E(x,y)))$$

- *No es el caso que cualquier libro es escrito por un filósofo.*

$$\neg\forall x\exists y(F(y) \wedge E(x,y))$$

- *Para cualquier libro escrito por Dogdson, "Grundlehren der Mathematik" es un libro más largo escrito por Hilbert.*

$$\forall x(E(x,d) \to E(g,h) \wedge L(g,x))$$

- *Hay alguien que es filósofo y matemático y ha escrito un libro en alemán.*

$$\exists x(F(x) \wedge M(x) \wedge \exists y(E(x,y) \wedge A(y)))$$

Ejemplo 1.5 *Existe un único objeto con la propiedad P.* En este caso la signatura es $\Sigma = \{P\}$ y usaremos un lenguaje con igualdad. A veces es conveniente transformar la oración en español en una otra que diga lo mismo pero que tenga una forma más cercana a los conectivos lógicos que puede incluir variables. En nuestro caso la oración original se transforma en *Existe un objeto x con la propiedad P y cualquier objeto con la propiedad P es x.* La traducción es:

$$\exists x(P(x) \wedge \forall y(P(y) \to y = x))$$

Ejemplo 1.6 *Una condición necesaria y suficiente para que el sultán sea felíz es que tenga vino, mujeres y música.* Hacemos $\Sigma = \{F^{(1)}, V^{(1)}, M^{(1)}, Mu^{(1)}, s\}$ con el significado obvio. La traducción es:

$$F(s) \leftrightarrow V(s) \wedge M(s) \wedge Mu(s)$$

Como se ve esta es una fórmula proposicional, es decir, no se necesitaron cuantificadores.

Ejemplo 1.7 *No hay un barbero que rasura exactamente a aquellos hombres que no se rasuran a si mismos.* Esto es equivalente a *no existe un barbero que rasura a un hombre y si y sólo si y no se rasura a si mismo.* La signatura es $\Sigma = \{B^{(1)}, R^{(2)}\}$ y la traducción es:

$$\neg\exists x(B(x) \wedge \forall y(R(x,y) \leftrightarrow \neg R(y,y)))$$

Ejercicios

1.2.1.- Hacer las siguientes traducciones, dando previamente una signatura adecuada.

(a) Cualquiera que sea persistente puede aprender lógica.

(b) Ningún guajolote vuela.

(c) No todos los pachucos bailan.

(d) Alguna rumbera baila con algun pachuco.

(e) Los perros muerden a los carteros.

(f) Existe un perro que muerde a los carteros.

(g) Existe un cartero que es mordido por todos los perros.

(h) Hay un perro que no muerde carteros.

(i) Hay un cartero que no es mordido por perros.

(j) Hay un perro que es cartero y se muerde a si mismo.

1.2.2.- Tenemos seis cubos de color amarillo, azul o verde. Un cubo puede estar uno sobre otro o en el piso. Considerese la signatura $\Sigma = \{S^{(2)}, A^{(1)}, Az^{(1)}, V^{(1)}, L^{(1)}, p\}$ donde $S(x,y)$ significa "x está sobre y", $A(x), Az(x), V(x)$ representan los colores, $L(x)$ significa que "x esta libre" es decir que ningún cubo está sobre x y p representa al piso.
Simbolizar los siguiente:

(a) Hay un cubo azul sobre el piso con un cubo amarillo sobre él y un cubo verde sobre el amarillo.

(b) Ningún cubo amarillo está libre.

(c) Hay un cubo azul libre y un cubo verde y libre.

(d) Cualquier cubo amarillo tiene un cubo sobre él.

(e) No todos los cubos azules están libres.

(f) Cualquier cubo verde está libre.

(g) Todos los cubos sobre el piso son azules.

(h) Cualquier cubo que esté sobre un cubo amarillo es verde o azul.

(i) Hay un cubo verde sobre un cubo verde.

(j) Hay un cubo amarillo libre sobre el piso.

(k) Ningún cubo está sobre el piso.

(l) Hay un cubo amarillo que está sobre uno azul y hay un cubo azul sobre él.

(m) Todos los cubos están sobre algo.

1.3 Principios de Inducción y Recursión

A lo largo de todo el libro necesitaremos hacer demostraciones acerca de propiedades de los términos y las fórmulas. Dado que tales objetos se han definido por recursión es claro que las demostraciones se harán, en la gran mayoría de los casos, mediante una inducción. A continuación enunciamos algunos principios de inducción que utilizaremos cuando sea conveniente.

- Principio de Inducción sobre Términos.
  Sea $\mathbb{P}$ una propiedad acerca de términos. Para demostrar que $\mathbb{P}$ es válida para todos los términos, basta probar que:
  - $\mathbb{P}$ es válida para las variables.
  - $\mathbb{P}$ es válida para las constantes.
  - Suponiendo que $\mathbb{P}$ es válida para $t_1, \ldots, t_n$ probar que $\mathbb{P}$ es válida para $f(t_1, \ldots, t_n)$, donde $f \in \Sigma$ es un símbolo de función n-ario.
- Principio de Inducción Fuerte sobre la lóngitud de los términos.
  Sea $\mathbb{P}$ una propiedad acerca de términos. Para demostrar que $\mathbb{P}$ es válida para todos los términos, basta suponer $\mathbb{P}$ para todos los términos con menos símbolos que t y probar para t.
- Principio de Inducción sobre Fórmulas.
  Sea $\mathbb{P}$ una propiedad acerca de fórmulas. Para demostrar que $\mathbb{P}$ es válida para todas las fórmulas, basta probar que:

  - $\mathbb{P}$ es válida para las fórmulas atómicas.
  - Suponiendo que $\mathbb{P}$ es válida para φ y ψ probar que $\mathbb{P}$ es válida para $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$, $\forall x\varphi$, $\exists x\varphi$.
    Por supuesto que se pude tomar un conjunto completo de conevtivos en lugar de probar todos los casos, cualquier combinación del símbolo de negación, un conectivo binario y un cuantificador bastan.

- Principio de Inducción Fuerte sobre la lóngitud de las fórmulas.
  Sea $\mathbb{P}$ una propiedad acerca de fórmulas. Para demostrar que $\mathbb{P}$ es válida para todas las fórmulas, basta suponer $\mathbb{P}$ para todas las fórmulas con menos símbolos que φ y probar para φ.

- Principio de inducción sobre el peso de una fórmula.
  Sea $\mathbb{P}$ una propiedad acerca de fórmulas. Para demostrar que $\mathbb{P}$ es válida para todas las fórmulas, basta suponer $\mathbb{P}$ para todas las fórmulas de menor peso que φ y probar para φ. El *peso* de una fórmula se define como el número de símbolos lógicos que figuran en ella.

En algunas ocasiones necesitaremos definir funciones o conjuntos relacionados con términos o fórmulas, para lo cual nos serán utiles los siguientes principios de recursión.

- Definición Recursiva para Términos.
  Para definir una función h sobre el conjunto $TERM(\Sigma)$, basta definir como sigue:
  - Definir $h(x)$ para $x \in VAR$.
  - Definir $h(c_k)$ para cada constante $c_k \in \Sigma$.
  - Suponiendo que $h(t_1), \ldots, h(t_n)$ están definidas, definir $h(f(t_1, \ldots, t_n))$ para cada símbolo de función $f \in \Sigma$ de aridad n, utilizando $h(t_1), \ldots, h(t_n)$.

- Definición Recursiva para Fórmulas.
  Para definir una función h sobre el conjunto $FORM(\Sigma)$, basta definir como sigue:
  - Definir $h(P(t_1, \ldots, t_n))$ para cada fórmula atómica $P(t_1, \ldots, t_n)$.

– Suponiendo definidas $h(\varphi)$ y $h(\psi)$, definir a partir de éllas, $h(\neg\varphi)$, $h(\varphi \vee \psi)$, $h(\varphi \wedge \psi)$, $h(\varphi \rightarrow \psi)$, $h(\varphi \leftrightarrow \psi)$, $h(\forall x\varphi)$, $h(\exists x\varphi)$. Por supuesto que basta definir para un conjunto completo de conectivos.

Ejemplo 1.8 Definimos recursivamente el *peso* de una fórmula como sigue:

- $p(P(t_1, \ldots, t_n)) = 0$.
- $p(\neg\varphi) = p(\varphi) + 1$.
- $p(\varphi \wedge \psi) = p(\varphi) + p(\psi) + 1$
- $p(\forall x\varphi) = p(\varphi) + 1$

Definimos el conjunto de subtérminos de un término t recursivamente como sigue:

- $Sub(x_n) = \{x_n\}$
- $Sub(c_k) = \{c_k\}$
- $Sub(f(t_1, \ldots, t_n)) = \{f(t_1, \ldots, t_n)\} \cup Sub(t_1) \cup \ldots \cup Sub(t_n)$

Ejercicios

1.3.1.- Definir recursivamente el conjunto de subfórmulas de la fórmula φ.

1.3.2.- Demuestre que cualquier fórmula tiene el mismo número de paréntesis izquierdos que derechos.

1.3.3.- Demuestre que cualquier término tiene el mismo número de paréntesis izquierdos que derechos.

1.3.4.- Definir recursivamente el conjunto $Var(t) = \{x \in VAR \mid x \text{ figura en } t\}$.

1.3.5.- Definir recursivamente la función longitud $long : TERM(\Sigma) \longrightarrow \mathbb{N}$ donde $long(t)$ es el número de símbolos de que figuran en t, sin contar los paréntesis.

1.3.6.- Sean t un término y n_i el número de presencias de símbolos de función i-arios en t. Sea $ncv(t)$ el número de presencias de variables y constantes en t. Muestre que

$$ncv(t) = 1 + \sum_i (i-1)n_i$$

1.4 Conceptos Sintácticos Importantes

Los siguientes conceptos serán de utilidad durante todo el libro.

Definición 1.7 Dada una cuantificación $\forall x\varphi$ o $\exists x\varphi$, la fórmula φ se llama el *alcance* del cuantificador.

Definición 1.8 Una presencia de la variable x en la fórmula φ está *acotada* syss es la variable de un cuantificador de φ o figura en el alcance de un cuantificador de φ.
Si una presencia de la variable x en la fórmula φ no está acotada, decimos que está *libre*.

Ejemplo 1.9 Sea $\varphi = \forall x\exists z(Q(y,z) \vee R(z,x,y)) \wedge P(z,x)$.
El alcance del cuantificador $\forall$ es la fórmula $\exists z(Q(y,z) \vee R(z,x,y))$; el alcance del cuantificador $\exists$ es la fórmula $Q(y,z) \vee R(z,x,y)$. En φ hay tres presencias de x, las dos primeras acotadas y la última libre; las presencias de z son cuatro, acotadas las tres primeras y libre la última; finalmente las dos presencias de y son libres.

Definición 1.9 Sea φ una fórmula. El conjunto de variables libres de φ, se denota $Vl(\varphi)$. Es decir, $Vl(\varphi) = \{x \in VAR \mid x \text{ figura libre en } \varphi\}$. La notación $\varphi(x_1,\ldots,x_n)$ quiere decir que $Vl(\varphi) = \{x_1,\ldots,x_n\}$.

Definición 1.10 Una fórmula φ es *cerrada* si no tiene variables libres, es decir, si $Vl(\varphi) = \varnothing$. Una fórmula cerrada también se conoce como *enunciado*. El conjunto de Σ-fórmulas cerradas se denota con $FORM_0(\Sigma)$.

Definición 1.11 Sea $\varphi(x_1,\ldots,x_n)$ una fórmula. La *cerradura universal* de φ, denotada $\forall\varphi$, es la fórmula $\forall x_1 \ldots \forall x_n\ \varphi$. La *cerradura existencial* de φ, denotada $\exists\varphi$ es la fórmula $\exists x_1 \ldots \exists x_n\ \varphi$.

Obsérvese que una cerradura se obtiene cuantificando todas las variables libres de una fórmula.

Ejercicios

1.4.1.- Defina recursivamente el conjunto $Vl(\varphi)$.

1.4.2.- Defina el conjunto $Vac(\varphi) = \{x \in VAR \mid \; x \text{ figura acotada en } \varphi\}$ mediante recursión para fórmulas.

1.4.3.- De ejemplos de fórmulas φ, ψ tales que $Vl(\varphi) \cap Vac(\varphi) = \varnothing$ y $Vl(\psi) \cap Vac(\psi) \neq \varnothing$.

1.4.4.- Obtener $Vl(\varphi)$ y $Vac(\varphi)$ para cada una de las siguientes fórmulas:

(a) $\forall x \forall y (Q(x, y) \wedge R(w)) \to \neg R(y, x)$.

(b) $\forall x \exists y R(y, x, z) \wedge \exists z Q(f(z), y)$.

(c) $T(c, x, v) \to \neg L(y, w, d) \vee \exists y \forall z R(y, z)$.

(d) $S(x, z) \to \forall z T(z, w, a) \vee \exists y R(y, w)$.

(e) $\forall x \forall y \exists v \exists w (Q(x, w) \wedge R(y, v) \to L(x, y, v))$.

1.4.5.- De ejemplos de fórmulas φ que cumplan lo siguiente:

(a) φ es un enunciado que es cerradura universal de una implicación en la que el consecuente es una cerradura existencial.

(b) φ no es un enunciado pero es cerradura existencial de una conjunción.

(c) φ es un fórmula con al menos tres presencias libres de exactamente dos variables distintas.

(d) φ es un enunciado que es disyunción de un enunciado atómico con un predicado ternario y una cerradura universal.

(e) φ no es un enunciado pero tiene dos cuantificadores universales con variables distintas, un cuantificador existencial y además se vuelve enunciado cuando se cierra universalmente con un sólo cuantificador más.

1.5 Sustituciones

La noción de sustitución es una herramienta indispensable para el lógico, sin élla no se podría hacer prácticamente nada.

Definición 1.12 Una *sustitución* es una función $\sigma : VAR \longrightarrow TERM(\Sigma)$ tal que $\sigma(x) = x$ excepto para un número finito de variables. Como una sustitución solamente mueve un número finito de variables, usaremos la notación $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ para denotar a la sustitución σ tal que $\sigma(x_i) = t_i$ para $1 \leq i \leq n$ y $\sigma(y) = y$ para $y \neq x_i$. Para sustituciones usaremos notación sufija y eliminaremos paréntesis, es decir, escribiremos $x\sigma$ en lugar de $\sigma(x)$. Si $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ entonces decimos que σ actúa sobre $x_1, \ldots, x_n$.

Definición 1.13 Una sustitución $\sigma : VAR \longrightarrow TERM(\Sigma)$ es una *sustitución cerrada* si $ran(\sigma) \subseteq TERM_0(\Sigma)$. Es decir, si el rango de σ consta de términos cerrados.

Las sustituciones hasta ahora definidas no son de gran utilidad puesto que sólo actúan sobre variables, a continuación hacemos extensiones a términos y fórmulas.

Definición 1.14 Sea $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ una sustitución, definimos la extensión $\sigma^\star : TERM(\Sigma) \longrightarrow TERM(\Sigma)$ recursivamente, como sigue:

$x\sigma^\star = x\sigma$, para toda $x \in VAR$.

$c\sigma^\star = c$, para cualquier símbolo de constante $c \in \Sigma$.

$f(t_1, \ldots, t_n)\sigma^\star = f(t_1\sigma^\star, \ldots, t_n\sigma^\star)$

En adelante cometeremos el abuso de denotar a $\sigma^\star$ como σ

Ejemplo 1.10 Sea $\sigma = \{x/y,\ y/f(z),\ z/c\}$.

$w\sigma = w$.

$t\sigma = f(z)$.

$c\sigma = c$.

$f(x, y, g(v, z))\sigma = f(x\sigma, y\sigma, g(v, z)\sigma) = f(y, f(z), g(v, c))$

Esta extensión ya es bastante útil, pero aún no basta, pues también necesitamos substituir variables en fórmulas, para lo cual definimos otra extensión.

Definición 1.15 Sea $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ una sustitución, definimos la extensión $\sigma^\star : FORM(\Sigma) \longrightarrow FORM(\Sigma)$ como sigue, $\varphi\sigma^\star$ es la fórmula que se obtiene a partir de φ reemplazando simultaneamente todas las presencias libres de x_i por t_i, para $1 \leq i \leq n$. Abusando de la notación escribimos $\varphi\sigma$ en lugar de $\varphi\sigma^\star$. A $\varphi\sigma$ se le llama una *instancia* de φ. Si $\varphi\sigma$ es un enunciado entonces decimos que σ es una *instancia cerrada*.

Ejemplo 1.11 Sean $\varphi = \forall x P(x) \vee \exists z Q(x, y, f(z))$. $\psi = \exists x Q(x, y, z) \wedge \forall z R(z, y)$ y $\sigma = \{x/f(x),\ y/d,\ z/g(c)\}$. Entonces

$$\varphi\sigma = \forall x P(x) \vee \exists z Q(f(x), d, f(z)).$$

$$\psi\sigma = \exists x Q(x, d, g(c)) \wedge \forall z R(z, d)$$

Definición 1.16 Sean $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ y $\varphi \in FORM(\Sigma)$. σ es una *instancia admisible* de φ si ninguna variable de t_i, para $1 \leq i \leq n$ se acota en φ al aplicar σ.

Veamos porque es necesario el concepto de instancia admisible. En el siguiente capítulo daremos un significado formal a todas las nociones sintácticas que hemos introducido hasta ahora, sin embargo, ya les hemos dado un significado intuitivo que usaremos ahora. Considerese la fórmula $\forall x\varphi$, con la cual representamos que la fórmula se cumple para todo valor posible de la variable x, de manera que nos gustaría poder concluir, a partir de la fórmula $\forall x\varphi$, la instancia $\varphi\{x/t\}$ para cualquier término t. Veamos que pasa si consideramos la fórmula $\forall x \exists y \neg(x = y)$, está fórmula expresa el hecho de que para cualquier individuo existe otro distinto de él y es cierta si nuestro universo de discurso tiene al menos dos elementos, pero al momento de obtener el caso particular para y obtenemos la fórmula $\exists y \neg(y = y)$ que claramente es falsa. '? Qué es lo está mal?, obsérvese que la presencia libre de x en la fórmula $\exists y \neg(x = y)$ se acotó al aplicar la sustitución $\{x/y\}$, es decir la instancia $(\exists y \neg(x = y))\{x/y\}$ no es admisible. La condición de admisibilidad prohibe que sucedan situaciones erroneas como la anterior.

Ejemplo 1.12 Considerense las siguientes sustituciones, $\sigma = \{x/f(x), y/c\}$, $\tau = \{x/f(y), y/z\}$, $\rho = \{x/c, z/g(x, y)\}$, entonces:

$(\forall x P(x, y))\sigma = \forall x P(x, c)$ es admisible.

$(\exists y R(x, z, y))\tau = \exists y R(f(y), z, y)$ no es admisible pues la presencia de y en el término $f(y)$ se vuelve acotada.

$(\forall x \exists y Q(z,x,y) \vee P(c,x))\rho = \forall x \exists y Q(g(x,y),x,y) \vee P(c,c)$ no es admisible pues las presencias de x, y en $g(x,y)$ se vuelven acotadas.

El siguiente lema nos muestra la equivalencia entre las distintas formas de ver una sustitución.

Lema 1.1 *Sean σ, ρ dos sustituciónes sobre la misma signatura Σ. Las siguientes condiciones son equivalentes.*

(a). $\sigma = \rho$, vistos como conjuntos finitos.

(b). $t\sigma = t\rho$ para cualquier término t.

(c). $x\sigma = x\rho$ para cualquier $x \in VAR$.

dem:

- $a) \Rightarrow b)$. Vamos a demostrar b mediante inducción para los términos.
  - Base de la inducción. El caso de las constantes es trivial, pues $c\sigma = c = c\rho$. Si $x \in VAR$ y $x \neq x_i$ entonces $x\sigma = x = x\rho$, aquí estamos usando la hipótesis de que $\sigma = \rho$ como conjuntos finitos de pares. Análogamente, $x_i\sigma = t_i$ implica que $x_i/t_i \in \sigma = \rho$. Por lo tanto $x_i\rho = t_i$.
  - Hipótesis de inducción $t_j\sigma = t_j\rho$ para $1 \leq j \leq m$.
  - Vamos a demostrar que $f(t_1, \ldots, t_m)\sigma = f(t_1, \ldots, t_m)\rho$. $f(t_1, \ldots, t_m)\sigma = f(t_1\sigma, \ldots, t_m\sigma) \underset{HI}{=} f(t_1\rho, \ldots, t_m\rho) = f(t_1, \ldots, t_m)\rho$

  Por lo tanto se cumple $b)$.
- $b) \Rightarrow c)$. Esto es inmediato pues las variables son términos.
- $c) \Rightarrow a)$. Veamos que $\sigma \subseteq \rho$. Sea $x/t \in \sigma$. esto significa que $x\sigma = t$, pero mediante $c)$ tenemos que $x\sigma = x\rho$, por lo tanto $x\rho = t$, es decir, $x/t \in \rho$. Análogamente se prueba que $\rho \subseteq \sigma$.

$\dashv$

El siguiente concepto de importancia es el de composición de sustituciones.

Definición 1.17 Sean $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$, $\rho = \{y_1/s_1, \ldots, y_n/s_m\}$ dos sustituciones sobre la misma signatura Σ. La composición $\sigma\rho$ queda definida de la siguiente manera:

$$\sigma\rho = \{x_i/t_i\rho \mid 1 \leq i \leq n \text{ y } x_i \neq t_i\rho\} \cup \{y_j/s_j \mid 1 \leq j \leq m \text{ y } y_j \notin \{x_1, \ldots, x_n\}\}$$

Ejemplo 1.13

Si $\sigma = \{y/x\}$ y $\rho = \{x/a\}$ entonces $\sigma\rho = \{x/a, y/a\}$.

Si $\sigma = \{x/f(y), y/z\}$ y $\rho = \{x/a, y/b, z/y\}$ entonces $\sigma\rho = \{x/f(b), z/y\}$.

Si $\sigma = \{x/f(x), y/z\}$ y $\rho = \{x/f(a), z/b\}$ entonces $\sigma\rho = \{x/f(f(a)), y/b, z/b\}$.

Si $\sigma = \{x/f(y), y/g(z), w/v\}$ y $\rho = \{x/a, y/b, z/f(y), v/w, u/c\}$ entonces $\sigma\rho = \{x/f(b),\ y/g(f(y)),\ z/f(y),\ v/w,\ u/c\}$.

El siguiente lema muestra algunas propiedades de la composición de sustituciones.

Lema 1.2 *(Lema de Composición)*
Sean σ, ρ, τ sustituciones sobre la misma signatura Σ, $t \in TERM(\Sigma)$ y $\varphi \in FORM(\Sigma)$. Entonces:

(a). $t(\sigma\rho) = (t\sigma)\rho$.

(b). $(\sigma\rho)\tau = \sigma(\rho\tau)$.

(c). Si $\varphi\sigma$ y $(\varphi\sigma)\rho$ son admisibles entonces $\varphi(\sigma\rho)$ es admisible.

(d). Si $\varphi\sigma$ y $(\varphi\sigma)\rho$ son admisibles entonces $\varphi(\sigma\rho) = \varphi(\sigma\rho)$

dem:

Sean $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$, $\rho = \{y_1/s_1, \ldots, y_n/s_m\}$

(a). Vamos a hacer inducción sobre los términos.

- Sea $z \in VAR$, hay que analizar tres casos
  - $z = x_i$. Entonces $z(\sigma\rho) = x_i(\sigma\rho) = t_i\rho$, por definición de composición. Por otra parte, $(z\sigma)\rho = (x_i\sigma)\rho = t_i\rho$. Por lo tanto $z(\sigma\rho) = (z\sigma)\rho$.
  - $z = y_j$ y $y_j \notin \{x_1, \dots, x_n\}$. En tal caso $z(\sigma\rho) = y_j(\sigma\rho) = y_j\rho = s_j$, por definición de composición. Por otra parte, $(z\sigma)\rho = (y_j\sigma)\rho = y_j\rho = s_j$.Por lo tanto $z(\sigma\rho) = (z\sigma)\rho$.
  - $z \neq x_i$ y $z \neq y_j$ para toda i, j. En este caso tenemos $z(\sigma\rho) = z$ y $(z\sigma)\rho = z\rho = z$.Por lo tanto $z(\sigma\rho) = (z\sigma)\rho$.
- El caso de las constantes es trivial. $c(\sigma\rho)$ y $(c\sigma)\rho = c\rho = c$.
- Hipótesis de inducción: $t_k(\sigma\rho) = (t_k\sigma)\rho$, con $1 \leq k \leq l$.
- $f(t_1, \dots, t_l)(\sigma\rho) = f(t_1(\sigma\rho), \dots, t_l(\sigma\rho)) \underset{HI}{=} f((t_1\sigma)\rho, \dots, (t_l\sigma)\rho) = f((t_1\sigma), \dots, (t_l\sigma))\rho = (f(t_1, \dots, t_l)\sigma)\rho$.

(b). Por el lema 1.1, basta ver que $z[(\sigma\rho)\tau] = z[\sigma(\rho\tau)]$, para $z \in VAR$.

$$z[(\sigma\rho)\tau] = [z(\sigma\rho)]\tau = [(z\sigma)\rho]\tau = (z\sigma)[\rho\tau] = z[\sigma(\rho\tau)]$$

donde todas las igualdades se obtuvieron utilizando (a).

(c). Queremos mostrar que $\varphi(\sigma\rho)$ es admisible.
Recordemos la definición de $\sigma\rho$.

$$\begin{aligned} \sigma\rho \;=\; & \{x_i/t_i\rho \mid 1 \leq i \leq n \text{ y } x_i \neq t_i\rho\} \cup \\ & \{y_j/s_j \mid 1 \leq j \leq m \text{ y } y_j \notin \{x_1, \dots, x_n\}\} \end{aligned}$$

Para ver que $\varphi(\sigma\rho)$ es admisible hay que checar que ninguna de las variables en $t_i\rho$ y s_j se acote al aplicar $\sigma\rho$ a φ. Tenemos dos casos:

- Sea $y_j/s_j \in \sigma\rho$, por lo tanto $y_j \notin \{x_1, \dots, x_n\}$.
  Suponga que hay una presencia libre de y_j en φ, dado que $y_j \notin \{x_1, \dots, x_n\}$ entonces la misma presencia de y_j está libre en $\varphi\sigma$, puesto que las variables que se cambiaron fueron las x_i. Por la admisibilidad de $(\varphi\sigma)\rho$ tenemos que ninguna varaible z de s_j se acota en $\varphi\sigma$ al reemplazar la presencia de y_j por s_j y si no se da el acotamiento de z en $\varphi\sigma$ tampoco se dará en φ puesto que σ no afecta a las y_j. Por lo tanto las variables de s_j permanecen libres.

- Sea $x_i/t_i\rho \in \sigma\rho$, por lo tanto $x_i \neq t_1\rho$.
  Suponga que hay una presencia libre de x_i en φ. Como $\varphi\sigma$ es admisible esta presencia de x_i no cae dentro del alcance de algún cuantificador $\forall z$ o $\exists z$, donde z es una variable de t_i. Por lo tanto, la subsecuente aplicación de ρ a $\varphi\sigma$ reemplaza t_i por $t_i\rho$. Además como $(\varphi\sigma)\rho$ es admisible, ninguna z que figure en $t_i\rho$ y que no haya estado presente en t_i cae dentro del alcance de un cuantificador $\forall z$ o $\exists z$. Por lo tanto las variables de $t_i\rho$ permanecen libres.

(d). Se le deja al lector.

$\dashv$

Ejercicios

1.5.1.- Reescriba la definición 1.15 mediante recursión para fórmulas.

1.5.2.- Defina la admisibilidad de $\varphi\sigma$ mediante recursión sobre φ.

1.5.3.- Obtener las composiciones $\sigma\rho$ y $\rho\sigma$ en cada caso:

(a) $\sigma = \{x/a\}$ $\rho = \{y/f(x)\}$

(b) $\sigma = \{x/a, y/f(z), z/u\}$ $\rho = \{u/y, x/f(x), v/g(a), z/y\}$

(c) $\sigma = \{y/f(g(a,v)), z/f(v), u/h(w)\}$ $\rho = \{v/f(u), w/a, y(h(v)\}$

(d) $\sigma = \{y/v, v/z\}$ $\rho = \{v/y, z/v\}$

(e) $\sigma = \{y/v, v/z\}$ $\rho = \{u/a, w/f(v)\}$

1.5.4.- Defina la admisibilidad de $\varphi\sigma$ mediante recursión sobre φ.

1.5.5.- Obtener la instancia $\varphi\sigma$ en cada caso y analizar su admisibilidad.

(a) $\varphi = \forall x\forall y P(x,y,z)$ $\sigma = \{x/a, y/f(a), z/f(y)\}$

(b) $\varphi = \forall x\exists z(Q(x,y) \wedge \exists y R(x, f(x)))$ $\sigma = \{x/g(a), y/f(b), z/b\}$

(c) $\varphi = \forall x\exists y T(x,y,z) \wedge \exists z P(f(x), g(y), z)$ $\sigma = \{x/f(a), y/g(b), z/f(x)\}$

(d) $\varphi = P(x,a,y) \vee \exists y(P(x,y,a) \wedge R(a,z))$ $\sigma = \{z/g(a,b), y/f(z)\}$

(e) $\varphi = W(f(x,a), g(b)) \wedge \forall x\exists y S(f(x,a), g(z))$ $\sigma = \{x/a, y/f(z,z), z/f(y,x)\}$

1.5.6.- Demuestre el lema 1.2(d).

1.5.7.- Muestre que si c es una constante entonces $Vl(\varphi\{x/c\}) = Vl(\varphi) - \{x\}$.

1.5.8.- Sean ϑ_1, ϑ_2 sustituciones tales que $\vartheta_1 = \vartheta_2\sigma_1$ y $\vartheta_2 = \vartheta_1\sigma_2$ para algunas σ_1, σ_2. Muestre que existe una sustitución ϱ tal que $ran(\varrho) \subseteq VAR$ y $\vartheta_1 = \vartheta_2\varrho$.

1.5.9.- Muestre que $\varphi\{x/t\}\{z/c\} = \varphi\{z/c\}\{z/t\}$ dado que x y z son variables distintas, c es una constante y z no figura en t.

1.5.10.- Muestre que $\varphi\{x/t\}\sigma = \varphi\{x/t\sigma\}$ si σ no actua en las variables libres de φ.

1.5.11.- Pruebe que φ es un enunciado syss $\varphi\sigma = \varphi$ para toda sustitución σ.

1.5.12.- Una sustitución σ es *idempotente* si $\sigma\sigma = \sigma$. Sean $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ y $V = Var(t_1) \cup \ldots \cup Var(t_n)$. Pruebe que σ es idempotente syss $\{x_1, \ldots, x_m\} \cap V = \varnothing$.

1.6 Unificación

Los sistemas de programación lógica tienen 2 cimientos, la regla de resolución que analizaremos más adelante y el proceso de *unificación*, que consiste en encontrar, dado un conjunto de literales o términos W, una substitución σ de tal forma que el conjunto imagen $W\sigma$ conste de un solo elemento. En adición a sus aplicaciones en programación lógica, la unificación también es importante para los sistemas de reescritura de términos y para el razonamiento automático.

Definición 1.18 Una *literal* es una fórmula atómica o la negación de una fórmula atómica

Definición 1.19 Una *expresión* es una literal o un término.

Definición 1.20 Sea W un conjunto no vacío de expresiones. Un *unificador* de W es una substitución σ tal que $\mid W\sigma \mid = 1$. Si W tiene un unificador decimos que W es *unificable*.

Ejemplo 1.14 Sea $W = \{P(x, f(y)), P(x, f(x)), P(u, v)\}$ entonces la sustitución $\sigma = \{x/a, y/a, u/a, v/f(a)\}$ es un unificador de W, ya que $W\sigma = \{P(a, f(a))\}$

Un conjunto de fórmulas puede tener una infinidad de unificadores o ninguno, dado un conjunto finito de fórmulas W, es decidible mediante un algoritmo si W es unificable o no; si W es unificable el algoritmo proporciona un unificador llamado *unificador más general*.

Definición 1.21 Un unificador σ de un conjunto de fórmulas W, se llama *unificador más general* (abreviado *umg*) si para cada unificador τ de W, existe una sustitución ϑ, tal que $\sigma\vartheta = \tau$

La importancia de los unificadores más generales es que nos permiten representar de manera finita un número infinito de sustituciones de manera que no estamos restringidos a razonar solamente sobre dominios finitos.

Ejemplo 1.15 Sean $W = \{f(g(a, x), g(y, b)), f(z, g(u, v))\}$, $\tau = \{x/a, z/g(a, a), y/u, v/b\}, \sigma = \{z/g(a, x), y/u, v/b\}$. Entonces τ y σ son unificadores de W y
σ resulta ser umg; en particular si $\vartheta = \{x/a\}$ entonces $\sigma\vartheta = \tau$.

1.6.1 Un Algoritmo de Unificación

Antes de pasar a la descripción formal del algoritmo vamos a hacer un análisis intuitivo del problema. Sea $W = \{E_1, E_2\}$, queremos ver si el conjunto W es unificable. En primer lugar las expresiones de W deben ser del mismo tipo, es decir, ambas términos o ambas literales. Después hay que analizar varios casos:

- Las expresiones E_1 y E_2 son constantes. En este caso $E_i\sigma = E_i$ para cualquier sustitución σ, de manera que W será unificable syss $E_1 = E_2$.
- Alguna de las dos expresiones es una variable. Supongamos que$E_1 = x$. Si x figura en E_2 entonces W no es unificable, en caso contrario $\sigma = \{x/E_2\}$ unifica a W.
- E_1 y E_2 son expresiones compuestas, es decir, términos compuestos o literales. En este caso W es unificable syss se cumplen las siguientes dos condiciones:
  - Los símbolos principales, es decir los primeros, de E_1 y E_2 son el mismo.
  - Cada par correspondiente de subexpresiones de E_1 y E_2 deben ser unificables.

Ejemplo 1.16 El conjuntos $\{c, d\}$ no es unificable pues consta de dos constantes diferentes; el conjunto $\{x, f(y)\}$ es unificable mediante $\sigma = \{x/f(y)\}$; el conjunto $\{P(x, w), Q(y, a)\}$ no es unificable pues P y Q son símbolos distintos; $\{f(x, g(y), w), f(a, g(b), h(w))\}$ no es unificable pues las subexpresiones w y $h(w)$ no son unificables,

Ahora pasamos al trato formal del problema presentando un algoritmo de unificación.

Definición 1.22 Sea $W \neq \emptyset$ un conjunto de expresiones. Definimos el *conjunto discorde de W* como sigue:
localizar la posición del primer símbolo, es decir, el símbolo más a la izquierda en el que al menos dos expresiones de W difieren y extraer de cada expresión de W la subexpresión que inicia en esta posición; el conjunto D_W de dichas subexpresiones es el *conjunto discorde de W*.

Ejemplo 1.17 Si $W = \{f(x,y), g(a)\}$ entonces $D_W = W$.

Ejemplo 1.18 Si $W = \{P(x, f(y)), P(x, f(a)), P(x, f(h(y)))\}$ entonces $D_W = \{y, a, h(y)\}$.

A continuación presentamos el algoritmo de unificación AU:

Precondición: $k = 0,\ \sigma_0 = \emptyset$
  while $W\sigma_k$ no sea unitario **do**
    Hallar el conjunto discorde D_k de $W\sigma_k$
    if existen $v, t \in D_k$ tales que v no figura en t **then**
      $\sigma_{k+1} := \sigma_k\{v/t\}$
      $k := k + 1$
    else
      write('W no es unificable')
      halt
    end if
  end while

Postcondición: σ_k es umg de W ó W no es unificable.

Veamos algunos ejemplos.

Ejemplo 1.19 Sea $W = \{P(f(a), g(x)),\ \ P(y, y)\}$. Tomamos el primer conjunto discorde $D_0 = \{f(a), y\}$. Como y no figura en $f(a)$ hacemos $\sigma_1 = \{y/f(a)\}$. $W\sigma_1 = \{P(f(a), g(x),\ \ P(f(a), f(a))\}$. Como $W\sigma_1$ no es unitario obtenemos $D_1 = \{g(x), f(a)\}$ y dado que en D_1 no hay variables el algoritmo termina siendo W no unificable.

Ejemplo 1.20 Sea $W = \{f(a, x, h(g(z))),\ \ f(z, h(y), h(y))\}$. Veamos las iteraciones del algoritmo en la siguiente tabla:

Sustitución	Conjunto Discorde
$\sigma_0 = \varnothing$	$D_0 = \{a, z\}$
$\sigma_1 = \{z/a\}$	$D_1 = \{x, h(y)\}$
$\sigma_2 = \{x/h(y), z/a\}$	$D_2 = \{g(a), y\}$
$\sigma_3 = \{x/h(g(a)), y/g(a), z/a\}$	

σ_3 es un umg de W.

Veamos por último un ejemplo con tres expresiones.

Ejemplo 1.21 Sea $W = \{Q(x,a,z), Q(y,a,h(y)), Q(x,a,h(g(b)))\}$. Las iteraciones son las siguientes:

Sustitución	Conjunto Discorde
$\sigma_0 = \varnothing$	$D_0 = \{x, y\}$
$\sigma_1 = \{y/x\}$	$D_1 = \{z, h(x)\}$
$\sigma_2 = \{y/x, z/h(x)\}$	$D_2 = \{x, g(b)\}$
$\sigma_3 = \{x/g(b), y/g(b), z/h(g(b))\}$	

σ_3 es un umg de W.Este ejemplo también muestra el caracter no determinista del algoritmo, ya que la sustitución x/y también pudo ser usada para σ_1.

Para finalizar la sección se hará el análisis del algoritmo de unificación mostrando que la especificación

$$\{k = 0,\ \sigma_0 = \emptyset\}\ AU\ \{\sigma_k \text{ es umg de } W \text{ o } W \text{ no es unificable }\}$$

es correcta.

Lema 1.3 *(Terminación de AU)*
Si m es el número de variables que figuran en W entonces el máximo k tal que σ_k es calculado en AU es menor o igual que m.

dem:

Obsérvese que todas las variables que figuran en $\sigma_k, W\sigma_k, D_k$ son variables de W, de manera que AU no introduce variables nuevas.
Al calcular $\sigma_{k+1} := \sigma_k\{v/t\}$, la variable v que figuraba en W se sustituye por un término que no la contiene, por lo que al calcular $W\sigma_k$ se reduce el número de variables en uno y como hay m variables este proceso puede efectuarse a lo más m veces. $\dashv$

Lema 1.4 *(Correctez parcial para un conjunto no unificable)*
Si W no es unificable, AU termina con el mensaje 'W no es unificable'.

dem:

Como $W\sigma_k$ no es nunca un unitario y AU siempre termina, por el lema anterior, necesariamente debe terminar con el mensaje requerido. $\dashv$

Lema 1.5 *(Las sustituciones intermedias son más generales)*
Sea ϑ un unificador de W. Si σ_k es la sustitución obtenida en la k-ésima iteración de AU entonces existe una sustitución τ_k tal que $\vartheta = \sigma_k \tau_k$.

dem:

Inducción sobre k.

- $k = 0$, es obvio, al tomar $\tau_0 = \vartheta$.
- Hipótesis de Inducción Existe un τ_k tal que $\vartheta = \sigma_k \tau_k$.
- Paso inductivo. Basta analizar el caso en que $W\sigma_k$ no es un unitario, pues si lo fuera el algoritmo habría terminado. Como $W\sigma_k$ no es unitario, AU produce el conjunto discorde D_k de $W\sigma_k$. Por HI, tenemos que $\vartheta = \sigma_k \tau_k$ y como ϑ unifica a W, se sigue que τ_k unifica a D_k. Por lo tanto debe existir una variable $v \in D_k$. Sea t cualquier otro término de D_k, v no figura en t puesto que $v\tau_k = t\tau_k$. Sea $\sigma_{k+1} = \sigma_k\{v/t\}$. Definimos $\tau_{k+1} = \tau_k \setminus \{v/v\tau_k\}$. Vamos a considerar 2 casos:

  (a). $v\tau_k \neq v$. Entonces

$$\begin{aligned} \tau_k &= \{v/v\tau_k\} \cup \tau_{k+1} \\ &\quad (v\tau_k = t\tau_k) \\ &= \{v/t\tau_k\} \cup \tau_{k+1} \\ &\quad (t\tau_k = t\tau_{k+1}) \\ &= \{v/t\}\tau_{k+1} \end{aligned}$$

  (b). $v\tau_k = v$. En este caso se tiene que D_k solo tiene variables como elementos, puesto que τ_k unifica a D_k; además $\tau_{k+1} = \tau_k$, por lo que $\tau_k = \{v/t\}\tau_{k+1}$.

  Así que en cualquier caso, $\tau_k = \{v/t\}\tau_{k+1}$, de donde se sigue que $\vartheta = \sigma_k \tau_k = \sigma_k\{v/t\}\tau_{k+1} = \sigma_{k+1}\tau_{k+1}$.

$\dashv$

Lema 1.6 *(Correctez Parcial para un conjunto unificable)*
Si W es unificable entonces la ultima sustitución generada por AU es un umg de W.

dem:

Sean ϑ un unificador de W y σ_m la última sustitución calculada por AU. Por el lema anterior existe τ_m tal que $\vartheta = \sigma_m \tau_m$. Como $W\vartheta$ es unitario, se sigue que τ_m unifica a $W\sigma_m$, es decir, se cumple la condición del ciclo *while*; pero como $\vartheta = \sigma_m \tau_m$ y ϑ era arbitrario, concluimos que σ_k es umg de W. $\dashv$

Teorema 1.1 *(Correctez total de AU)*
Dado un conjunto finito de expresiones W, el algoritmo AU termina, dando como resultado el mensaje 'W no es unificable', en el caso en que W no sea unificable, y un umg de W en el caso en que W sea unificable.

dem:

Es consecuencia inmediata de los lemas anteriores. $\dashv$

Para finalizar la sección veamos que, en cierto sentido, los unificadores más generales son únicos.

Definición 1.23 Una sustitución σ es una *variante* de ρ si existen τ_1, τ_2 tales que $\sigma = \rho\tau_1$ y $\rho = \sigma\tau_2$.

Ejemplo 1.22 $\sigma = \{x/a, y/z\}$ es una variante de $\rho = \{x/a, y/x, z/y\}$, ya que $\sigma = \rho\{x/z\}$ y $\rho = \sigma\{z/x\}$.

Proposición 1.1 *Sean μ_1, μ_2 unificadores más generales del conjunto W. Entonces μ_1 es variante de μ_2.*

dem:

Como μ_1 es umg de W, entonces, por la definición 1.21 existe τ_1 tal que $\mu_2 = \mu_1\tau_1$ y analogamente existe τ_2 tal que $\mu_1 = \mu_2\tau_1$. Por lo tanto μ_1 y μ_2 son variantes. $\dashv$

Ejercicios

1.6.1.- Decidir si los siguientes conjuntos son unificables mediante el algoritmo de unificación.

(a) $W = \{P(g(a,x), y), P(g(z, h(y), b)\}$

(b) $W = \{f(x, y, h(a, x, y)), f(g(a), g(b), h(a, g(z), w))\}$

(c) $W = \{f(x, f(x, z)), f(g(x), f(a, b))\}$

(d) $W = \{Q(a, x, f(g(y))), Q(z, h(z, w), f(w))\}$

(e) $W = \{R(a, f(b), x, w), R(a, f(b), y, g(a, w))\}$

1.6.2.- Pruebe que el umg obtenido mediante el algoritmo de unificación es idempotente.

1.6.3.- Sean S, T dos conjuntos finitos de términos tales que $Var(S) \cap Var(T) = \varnothing$. Pruebe que si $S \cup T$ es unificable, σ_S es un umg de S, σ_T es un umg de T y σ_{ST} es un umg de $S\sigma_S \cup T\sigma_T$ entonces $\sigma_{ST}\sigma_T\sigma_S$ es un umg de $S \cup T$.

1.6.4.- Sea ϑ un umg de un conjunto finito de términos S. Pruebe que ϑ es umg e idempotente syss para cada unificador σ de S, $\sigma = \vartheta\sigma$.

Capítulo 2

Semántica de la Lógica de Predicados

En este capítulo le daremos significado a las nociones sintácticas del capítulo anterior, presentaremos los grandes teoremas de la lógica clásica y dejaremos preparado el camino hacia los métodos de la programación lógica.

2.1 Interpretaciónes y Estados

Si bien en el capítulo anterior le dimos cierto sentido a las fórmulas al momento de hacer traducciones el tratamiento formal fue puramente sintáctico. Es hasta ahora que les daremos a las fórmulas un significado formal. Iniciamos con un ejemplo, sea $\varphi = \forall x P(x, x)$, para conocer el significado de φ necesitamos saber el significado del predicado P y además debemos conocer nuestro universo de discurso, es decir, cuales son los individuos a los que nos estamos refiriendo, de esta manera los cuantificadores varían sobre los individuos del universo de discurso, de aquí los lenguajes como los que hemos descrito se llamen lenguajes de primer órden. También existen *lenguajes de orden superior* cuya diferencia con los de primer orden radica en poder cuantificar no solamente sobre individuos sino sobre conjuntos de individuos, conjuntos de conjuntos de individuos, etc . Dependiendo del universo y el significado de los símbolos elegidos una misma fórmula puede ser cierta o falsa o en ocasiones ninguna de las dos cosas, por ejemplo, siguiendo con la fórmula φ de arriba, si interpretamos a P como la relación de divisibilidad en los enteros, claramente φ será cierta; sin embargo si P significa el orden

en los naturales, entonces φ será falsa.
Obsérvese que los únicos símbolos que cambian de significado son los símbolos no logicos del lenguaje, que son los símbolos de la signatura. Para fijar una manera de interpretar los símbolos de una signatura introducimos la noción de interpretación o estructura.

Definición 2.1 Sea Σ una signatura. Una Σ-*interpretación* o Σ-*estructura* $\mathfrak{A}$ consta de lo siguiente:

- Un conjunto $A \neq \emptyset$ llamado el *universo o dominio* de la estructura y denotado $\mid \mathfrak{A} \mid$ o $dom(\mathfrak{A})$.
- Para cada símbolo de constante $c \in \Sigma$, una elemento de A, denotado $c^{\mathfrak{A}}$ y llamado la *interpretación de la constante c*.
- Para cada símbolo de predicado n-ario $P \in \Sigma$, una relación n-aria sobre $\mid \mathfrak{A} \mid$, $P^{\mathfrak{A}} \subseteq A^n$. $P^{\mathfrak{A}}$ es la *interpretación de P*.
- Para cada símbolo de función n-ario $f \in \Sigma$, una función $f^{\mathfrak{A}} : A^n \longrightarrow A$, $f^{\mathfrak{A}}$ es la *interpretación de f*.

Usaremos letras góticas mayúsculas $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}$, etc. para denotar estructuras. Si $\Sigma = \{P_i, f_j, c_k\}$ entonces es usual denotar a una Σ-estructura con $\mathfrak{A} = \langle A, P_i^{\mathfrak{A}}, f_j^{\mathfrak{A}}, c_k^{\mathfrak{A}} \rangle$.

Así que cada lenguaje $\mathcal{L}_\Sigma$ puede tener asociada una infinidad de estructuras todas diferentes entre si, mientras que una estructura solo puede simbolizarse mediante una estructura única, salvo isomorfismo.

Veamos algunos ejemplos usuales.

Ejemplo 2.1 Una gráfica es un conjunto de vértices V junto con una relación binaria $A \subseteq V \times V$ que representa a las aristas. Si tenemos la signatura $\Sigma = \{Q^{(2)}\}$, entonces la Σ-estructura $\mathfrak{A} = \langle V, A \rangle$, donde $Q^{\mathfrak{A}} = A$ es una gráfica.

Ejemplo 2.2 Un conjunto total o parcialmente ordenado puede verse como una Σ-estructura $\mathfrak{B} = \langle X, < \rangle$, donde Σ es la misma que en el ejemplo anterior y $Q^{\mathfrak{B}} =<$.

Ejemplo 2.3 Veamos ahora las estructuras algebraicas más usuales. Sean $\Sigma_1 = \{f^{(2)}\}$, $\Sigma_2 = \{f^{(2)}, c\}$, $\Sigma_3 = \{f^{(2)}, g^{(2)}, c\}$, $\Sigma_4 = \{f^{(2)}, g^{(2)}, c, d\}$.

(a). Estructuras con una operación binaria.

(i) Semigrupo. Un *semigrupo* es una Σ_1-estructura $\mathfrak{A} = \langle S, \otimes \rangle$ donde se cumple que $\forall x \forall y \forall z(f(x, f(y, z)) = f(f(x, y), z))$, es decir, $f^{\mathfrak{A}} = \otimes$ es asociativa. Como ejemplo tenemos $\mathfrak{A} = \langle \mathbb{Z}^+, + \rangle$.

(ii) Monoide. Un *monoide* es una Σ_2-estructura $\mathfrak{A} = \langle M, \otimes, e \rangle$ donde se cumple que $\forall x \forall y \forall z(f(x, f(y, z)) = f(f(x, y), z)) \wedge \forall x(f(e, x) = x = f(x, e))$, es decir, $f^{\mathfrak{A}} = \otimes$ es asociativa y $c^{\mathfrak{A}} = e$ es un neutro para $\otimes$. Como ejemplo tenemos $\mathfrak{A} = \langle \mathbb{N}, +, 0 \rangle$.

(iii) Grupo. Un *grupo* es una Σ_2-estructura $\mathfrak{A} = \langle G, \otimes, e \rangle$ donde se cumple que $\forall x \forall y \forall z(f(x, f(y, z)) = f(f(x, y), z)) \wedge \forall x(f(e, x) = x = f(x, e)) \wedge \forall x \exists y(f(x, y) = e)$, es decir, $f^{\mathfrak{A}} = \otimes$ es asociativa, $c^{\mathfrak{A}} = e$ es un neutro para $\otimes$ y cada elemento de G tiene un inverso respecto a $f^{\mathfrak{A}}$. Como ejemplo tenemos $\mathfrak{A} = \langle \mathbb{Z}, +, 0 \rangle$.

(iv) Grupo Abeliano. Un *grupo abeliano* es una Σ_2-estructura $\mathfrak{A} = \langle G, \otimes, e \rangle$ tal que $\mathfrak{A}$ es un grupo y además se cumple que $\forall x \forall y(f(x, y) = f(y, x))$, es decir, $f^{\mathfrak{A}} = \otimes$ es conmutativa. Como ejemplo tenemos $\mathfrak{A} = \langle \mathbb{Q}, +, 0 \rangle$

(b). Estructuras con dos operaciones binarias.

(i) Anillo. Un *anillo* es una Σ_3-estructura $\mathfrak{A} = \langle A, \oplus, \otimes, e \rangle$ tal que $\langle A, \oplus, e \rangle$ es un grupo abeliano, $\otimes^{\mathfrak{A}}$ es asociativa y se cumple que $\forall x \forall y \forall z(g(x, f(y, z)) = f(g(x, y), g(x, z)) \wedge \forall x \forall y \forall z(g(f(x, y), z) = f(g(x, z), g(y, z)))$, es decir, $\oplus$ distribuye a $\otimes$.

(ii) Anillo Conmutativo. Un *anillo conmutativo* es una Σ_3-estructura $\mathfrak{A} = \langle A, \oplus, \otimes, e \rangle$ tal que $\mathfrak{A}$ es un anillo y $g^{\mathfrak{A}} = \otimes$ es conmutativa.

(iii) Anillo con Unitario. Un *anillo con unitario* es una Σ_4-estructura $\mathfrak{A} = \langle A, \oplus, \otimes, e, 1 \rangle$ tal que $\langle A, \oplus, \otimes, e \rangle$ es un anillo y $d^{\mathfrak{A}} = 1$ es un neutro para $g^{\mathfrak{A}} = \otimes$.

(iv) Dominio Entero. Un *dominio entero* es una Σ_4-estructura $\mathfrak{A} = \langle A, \oplus, \otimes, e, 1 \rangle$ tal que $\mathfrak{A}$ es un anillo conmutativo con unitario y se cumple que $\forall x \forall y(g(x, y) = e \rightarrow x = e \vee y = e)$. Como ejemplo tenemos $\mathfrak{A} = \langle \mathbb{Z}, +, \cdot, 0, 1 \rangle$

(v) Anillo con División. Un *anillo con división* es una Σ_4-estructura $\mathfrak{A} = \langle A, \oplus, \otimes, e, 1 \rangle$ tal que $\mathfrak{A}$ es un anillo con unitario y se cumple que $\forall x(x \neq e \rightarrow \exists y(g(x,y) = d))$, es decir, todo elemento no cero tiene un inverso multiplicativo.

(vi) Campo. Un *campo* es una Σ_4-estructura $\mathfrak{A} = \langle A, \oplus, \otimes, e, 1\rangle$ tal que $\mathfrak{A}$ es un anillo conmutativo con división. Como ejemplo tenemos $\mathfrak{A} = \langle \mathbb{R}, +, \cdot, 0, 1\rangle$

Ya hemos dicho que para conocer el significado de una fórmula cualquiera necesitamos saber como se interpretan sus símbolos no lógicos. Sin embargo a veces esto no es suficiente, considerese por ejemplo la fórmula $\varphi \equiv \exists x(f(x,y) = z)$, y preguntemonos si será cierta o falsa en determinada estructura, por ejemplo en $\mathfrak{A} = \langle \mathbb{N}, +\rangle$. Aún sabiendo que $f^{\mathfrak{A}} = +$ no podemos contestar la pregunta pues no hay manera de saber cuanto valen y y z. Por ejemplo si y vale 2 y z vale 5 entonces la fórmula es cierta haciendo que x valga 3. Sin embargo si y vale 10 y z vale 7 entonces la fórmula es falsa en $\mathfrak{A}$ pues $-3 \notin \mathbb{N}$. De manera que con lo visto hasta ahora no podemos todavia encontrar el valor de verdad de una fórmula. Para esto necesitamos saber el valor de las variables libres que figuran en élla, cosa que formalizamos mediante el siguiente concepto.

Definición 2.2 Sea $\mathfrak{A}$ una Σ-interpretación. Un *estado de las variables* es una función $s : VAR \longrightarrow \mid \mathfrak{A} \mid$.

Lo que un estado hace es informarnos acerca del valor de cada variable en la estructura dada, es decir, un estado nos dice como interpretar las variables libres de una fórmula. ¿ Será necesario interpretar las variables acotadas ?.

Ejercicios

2.1.1.- Sea $A \neq \varnothing$ un conjunto. De ejemplos de estructuras cuyo universo sea el conjunto potencia de A, $\mathcal{P}(A)$.

2.1.2.- Sea V un espacio vectorial sobre el campo K. Represente a V mediante una estructura. Es más fácil tomar como universo el conjunto de vectores V y encontrar una manera adecuada para representar la multiplicación por escalares.

2.1.3.- Para cada uno de los siguientes conjuntos dar una estructura cuyo universo sea el conjunto dado y que tenga al menos una relación, una función y una constante.

(a) $\mathbb{Q}[x]$. El Anillo de polinomios con coeficientes en $\mathbb{Q}$.

(b) $\mathbb{R}_{\mathbb{R}}$. El conjunto de funciones $f : \mathbb{R} \longrightarrow \mathbb{R}$.

(c) $Suc = \{\langle x_n \rangle_{n\in\mathbb{N}} \mid \; x_n \in \mathbb{R}\}$, El conjunto de sucesiones de números reales.

2.1.4.- Sea $\mathbb{P}$ un conjunto de símbolos de proposición. Construir una estructura que represente a la lógica proposicional cuyas proposiciones atómicas sean los elementos de $\mathbb{P}$.

2.2 Evaluación de Términos

En la sección anterior aprendimos a interpretar los símbolos no lógicos de un lenguaje así como las variables en una estructura dada, pero para saber el valor de verdad de una fórmula necesitamos interpretar o evaluar los términos que figuran en élla, cosa que haremos mediante la siguiente definición.

Definición 2.3 Sean $\mathfrak{A}$ una Σ-estructura, $s : VAR \longrightarrow \mid \mathfrak{A} \mid$ un estado de las variables y $t \in TERM(\Sigma)$. Definimos la *interpretación de* t *en* $\mathfrak{A}$ *según* s, denotada $t^{\mathfrak{A}}[s]$, recursivamente como sigue:

$$c^{\mathfrak{A}}[s] = c^{\mathfrak{A}}.$$

$$x^{\mathfrak{A}}[s] = s(x).$$

$$f(t_1, \ldots, t_n)^{\mathfrak{A}}[s] = f^{\mathfrak{A}}(t_1^{\mathfrak{A}}[s], \ldots, t_n^{\mathfrak{A}}[s]).$$

Veamos un ejemplo.

Ejemplo 2.4 Sean $\mathfrak{A} = \langle \mathbb{Q}, \cdot, g^{\mathfrak{A}}, \frac{8}{5} \rangle$, donde $f^{\mathfrak{A}} = \cdot$, $g^{\mathfrak{A}}(\frac{p}{q}) = p$, $c^{\mathfrak{A}} = \frac{8}{5}$, $s : VAR \longrightarrow \mathbb{Q}$ tal que $s(x_i) = \frac{i}{4}$ Vamos a interpretar algunos términos en $\mathfrak{A}$.

(a). $x_{23}^{\mathfrak{A}}[s] = \frac{23}{4}$

(b). $c^{\mathfrak{A}}[s] = c^{\mathfrak{A}} = \frac{8}{5}$

(c). $f(x_4, x_{15})^{\mathfrak{A}}[s] = f^{\mathfrak{A}}(x_4^{\mathfrak{A}}[s], x_{15}^{\mathfrak{A}}[s]) = \cdot(s(x_4), s(x_{15})) = \frac{4}{4} \cdot \frac{15}{4} = \frac{15}{4}$

(d). $g(f(c, x_7))^{\mathfrak{A}}[s] = g^{\mathfrak{A}}(f(c, x_7)^{\mathfrak{A}}[s]) = g^{\mathfrak{A}}(f^{\mathfrak{A}}(c^{\mathfrak{A}}[s], x_7^{\mathfrak{A}}[s])) = g^{\mathfrak{A}}(\frac{8}{5} \cdot s(x_7)) = g^{\mathfrak{A}}(\frac{8}{5} \cdot \frac{7}{4}) = g^{\mathfrak{A}}(\frac{14}{5}) = 14$

Los siguientes lemas serán de gran utilidad más adelante pues optimizan la evaluación de términos.

Lema 2.1 *(Lema de Coincidencia para Términos)*
Sean $\mathfrak{A}$ una Σ-interpretación, $t \in TERM(\Sigma)$ y $s, s' : VAR \longrightarrow \mid \mathfrak{A} \mid$ dos estados de las variables. Si $s(x) = s'(x)$ para toda x que figure en t entonces $t^{\mathfrak{A}}[s] = t^{\mathfrak{A}}[s']$.

dem:

Por Inducción sobre los términos.

- Base de la Inducción.
  - $t = c$. Entonces $c^{\mathfrak{A}}[s] = c^{\mathfrak{A}} = c^{\mathfrak{A}}[s']$
  - $t = x$. Entonces $x^{\mathfrak{A}}[s] = s(x) \underset{Hip.}{=} s'(x) = x^{\mathfrak{A}}[s']$.
- Hipótesis de inducción. $t_i^{\mathfrak{A}}[s] = t_i^{\mathfrak{A}}[s']$ para $1 \leq i \leq n$.
- $t = f(t_1, \ldots, t_n)$. Entonces $f(t_1, \ldots, t_n)^{\mathfrak{A}}[s] = f^{\mathfrak{A}}(t_1^{\mathfrak{A}}[s], \ldots, t_n^{\mathfrak{A}}[s]) \underset{HI}{=} f^{\mathfrak{A}}(t_1^{\mathfrak{A}}[s'], \ldots, t_n^{\mathfrak{A}}[s']) = f(t_1, \ldots, t_n)^{\mathfrak{A}}[s']$

$\dashv$

Definición 2.4 Sean $\mathfrak{A}$ una Σ-interpretación, $s : VAR \longrightarrow \mid \mathfrak{A} \mid$, $x \in VAR$ y $a \in \mid \mathfrak{A} \mid$. Definimos el *estado modificado en x a partir de s*, denotado $s(x/a)$, como sigue:

$$s(x/a)(y) = \begin{cases} a & \text{si } y = x \\ \\ s(y) & \text{si } y \neq x \end{cases}$$

Obsérvese que $s(x/a) : VAR \longrightarrow \mid \mathfrak{A} \mid$ es un estado que difiere de s únicamente en el valor de x. y que el símbolo $s(x/a)$ es sólo el nombre del estado, la aplicación a una variable es $s(x/a)(y)$. Como puede observarse esta notación recuerda a una sustitución, esto no es conicidencia, sino que los dos conceptos están relacionados como veremos a continuación.
Supóngase que tenemos un término t que tiene muchas presencias de la variable x y que queremos evaluar $t\{x/r\}$ en algún estado de cierta interpretación,

para esto aplicamos la definición recursiva y cada vez que encontramos una presencia del subtérmino r en $t\{x/r\}$ en debemos calcular el valor de r en el mismo estado; esta situación es poco práctica, pensemos por ejemplo en que r es un término complicado y que hay miles de presencias de r en t. Una mejor manera de evaluar $t\{x/r\}$ sería evaluar r en el estado dado s, lo cual nos da cierto valor a y evaluar el término $t\{x/r\}$ en el estado modificado $s(x/a)$. El resultado será el mismo, como lo asegura el siguiente lema.

Lema 2.2 *(Lema de Sustitución para Términos).*
Sean $\mathfrak{A}$ una Σ-interpretación, s un estado de las variables, $x \in VAR$ y $t, r \in TERM(\Sigma)$. Si $r^{\mathfrak{A}}[s] = a$ entonces

$$(t\{x/r\})^{\mathfrak{A}}[s] = t^{\mathfrak{A}}[s(x/a)]$$

dem:

Inducción sobre el término t.

- Base de la Inducción.
  - $t = c$. Entonces $(c\{x/r\})^{\mathfrak{A}}[s] = (c)^{\mathfrak{A}}[s] = c^{\mathfrak{A}} = c^{\mathfrak{A}}[s(x/a)]$
  - $t \in VAR$
    - $t = y \neq x$. Entonces $(y\{x/r\})^{\mathfrak{A}}[s] = (y)^{\mathfrak{A}}[s] = s(y) = s(x/a)(y) = y^{\mathfrak{A}}[s(x/a)]$
    - $t = x$. Entonces $(x\{x/r\})^{\mathfrak{A}}[s] = (r)^{\mathfrak{A}}[s] = a = s(x/a)(x) = x^{\mathfrak{A}}[s(x/a)]$
- H.I. $(t_i\{x/r\})^{\mathfrak{A}}[s] = t_i^{\mathfrak{A}}[s(x/a)]$ para $1 \leq i \leq n$.
- $t = f(t_1, \ldots, t_n)$. Entonces

$$\begin{aligned}
(f(t_1, \ldots, t_n)\{x/r\})^{\mathfrak{A}}[s] &= \\
f(t_1\{x/r\}, \ldots, t_n\{x/r\})^{\mathfrak{A}}[s] &= \\
f^{\mathfrak{A}}(t_1\{x/r\}^{\mathfrak{A}}[s], \ldots, t_n\{x/r\}^{\mathfrak{A}}[s]) &\underset{HI}{=} \\
f^{\mathfrak{A}}(t_1^{\mathfrak{A}}[s(x/a)], \ldots, t_n^{\mathfrak{A}}[s(x/a)]) &= \\
f(t_1, \ldots, t_n)^{\mathfrak{A}}[s(x/a)] &
\end{aligned}$$

$\dashv$

Ejercicios

2.2.1.- Sean $\mathfrak{A} = \langle \mathbb{Q}, +, \cdot, h^{\mathfrak{A}}, 1, \frac{1}{2} \rangle$ y $\mathfrak{B} = \langle \mathbb{Z}, \cdot, +, h^{\mathfrak{A}}, -1, 3 \rangle$. Donde $f^{\mathfrak{A}} = + = g^{\mathfrak{B}}$, $g^{\mathfrak{A}} = \cdot = f^{\mathfrak{B}}$, $a^{\mathfrak{A}} = 1$, $a^{\mathfrak{B}} = -1$, $b^{\mathfrak{A}} = \frac{1}{2}$, $b^{\mathfrak{B}} = 3$, $h^{\mathfrak{A}}$ cumple $h^{\mathfrak{A}}(\frac{p}{q}) = \frac{q}{p}$, $h^{\mathfrak{A}}(0) = 0$ y $h^{\mathfrak{B}}$ cumple $h^{\mathfrak{B}}(m) = -m$. Sean $s : VAR \longrightarrow \mathbb{Q}$ tal que $s(x) = \frac{1}{3}, s(y) = 2, s(z) = \frac{1}{4}$ y $s' : VAR \longrightarrow \mathbb{Z}$ tal que $s'(x) = -2$, $s'(y) = -4$, $s'(z) = 5$.
Evaluar los siguientes términos en $\mathfrak{A}$ según s y en $\mathfrak{B}$ según s'.

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

(b) $g(h(a), g(b, x))$.

(c) $h(f(x, y))$.

(d) $f(h(z), g(z, b))$.

(e) $h(g(a, x))$.

2.3 La definición de Satisfacción de Tarski

Ahora que ya sabemos evaluar términos es posible dar la definición formal de verdad, pero como ya hemos visto una fórmula no necesariamente tiene que ser verdadera o falsa en un estado dado, como sucedia en el caso de la lógica proposicional, de manera que el criterio de verdad es más complicado en este caso y está basado en la noción de *satisfacción* que es más simple.

Definición 2.5 (Definición de satisfacción de Tarski 1936).
Sean $\mathfrak{A}$ una Σ-interpretación, s un estado de las variables y $\varphi \in FORM(\Sigma)$. Definimos recursivamente cuando φ *es satisfacible en* $\mathfrak{A}$ *mediante* s, lo cual se denota $\mathfrak{A} \models \varphi[s]$, como sigue:

- $\mathfrak{A} \models P(t_1, \ldots, t_n)[s]$ syss $P^{\mathfrak{A}}(t_1^{\mathfrak{A}}[s], \ldots, t_n^{\mathfrak{A}}[s])$[1]

- Si el lenguaje tiene igualdad entonces $\mathfrak{A} \models (t_1 = t_2)[s]$ syss $t_1^{\mathfrak{A}}[s] = t_2^{\mathfrak{A}}[s]$.

[1] $P^{\mathfrak{A}}(t_1^{\mathfrak{A}}[s], \ldots, t_n^{\mathfrak{A}}[s])$ significa que $(t_1^{\mathfrak{A}}[s], \ldots, t_n^{\mathfrak{A}}[s]) \in P^{\mathfrak{A}}$, es decir, que la n-ada de términos evaluados está en la relación $P^{\mathfrak{A}}$.

- $\mathfrak{A} \models \neg\psi[s]$ syss $\mathfrak{A} \not\models \psi[s]$.
- $\mathfrak{A} \models (\psi \vee \chi)[s]$ syss $\mathfrak{A} \models \psi[s]$ o $\mathfrak{A} \models \chi[s]$
- $\mathfrak{A} \models (\psi \wedge \chi)[s]$ syss $\mathfrak{A} \models \psi[s]$ y $\mathfrak{A} \models \chi[s]$
- $\mathfrak{A} \models (\psi \rightarrow \chi)[s]$ syss $\mathfrak{A} \not\models \psi[s]$ o $\mathfrak{A} \models \chi[s]$
- $\mathfrak{A} \models (\psi \leftrightarrow \chi)[s]$ syss $\mathfrak{A} \models \psi[s]$ y $\mathfrak{A} \models \chi[s]$, o $\mathfrak{A} \not\models \psi[s]$ y $\mathfrak{A} \not\models \chi[s]$.
- $\mathfrak{A} \models \forall x\psi[s]$ syss $\mathfrak{A} \models \psi[s(x/a)]$, para toda $a \in\mid \mathfrak{A} \mid$.
- $\mathfrak{A} \models \exists x\psi[s]$ syss $\mathfrak{A} \models \psi[s(x/a)]$, para alguna $a \in\mid \mathfrak{A} \mid$.

Decimos que φ es *satisfacible* si existen una interpretación $\mathfrak{A}$ y un estado s tales que $\mathfrak{A} \models \varphi[s]$.

Ejemplo 2.5 La fórmula atómica $P(x, y)$ es satisfacible. Tomando $\mathfrak{A} = \langle \mathbb{Z}, < \rangle$ y $s(x) = -2, s(y) = 5$ tenemos que $P^{\mathfrak{A}}(x^{\mathfrak{A}}[s], y^{\mathfrak{A}}[s])$ syss $< (s(x), s(y))$ syss $-2 < 5$ y esto último es claramente cierto. Por lo tanto $\mathfrak{A} \models P(x, y)[s]$. Sin embargo, también la fórmula $\neg P(x, y)$ es satisfacible en $\mathfrak{A}$ mediante el estado s' tal que $s'(x) = 4, s'(x) = 1$. En este caso tenemos que $(x^{\mathfrak{A}}[s], y^{\mathfrak{A}}[s]) = (s(x), s(y)) = (4, 1) \notin P^{\mathfrak{A}}$ pues $4 \not< 1$. Por lo tanto $\mathfrak{A} \not\models P(x, y)[s']$ es decir, $\mathfrak{A} \models \neg P(x, y)[s']$.

A continuación mostramos los lemas de coincidencia y sustitución en el caso de fórmulas.

Lema 2.3 *(Lema de Coincidencia para Fórmulas).*
Sean $\mathfrak{A}$ una Σ-interpretación, s, s' cualesquiera dos estados de las variables y $\varphi \in FORM(\Sigma)$. Si $s(x) = s'(x)$ para toda $x \in Vl(\varphi)$ entonces

$$\mathfrak{A} \models \varphi[s] \textit{ syss } \mathfrak{A} \models \varphi[s'].$$

dem:

Por inducción sobre las fórmulas.

- Base de la Inducción $\varphi = P(t_1, \ldots, t_n)$. Tenemos que $\mathfrak{A} \models P(t_1, \ldots, t_n)[s]$ syss $P^{\mathfrak{A}}(t_1^{\mathfrak{A}}[s], \ldots, t_n^{\mathfrak{A}}[s])$. Como $Var(t_i) \subseteq Vl(\varphi)$ entonces $s(x) = s'(x)$ p.t. $x \in Var(t_i)$. Por lo tanto podemos aplicar el lema 2.1 con lo que obtenemos $t_i^{\mathfrak{A}}[s] = t_i^{\mathfrak{A}}[s']$. Por lo tanto $\mathfrak{A} \models P(t_1, \ldots, t_n)[s]$ syss $P^{\mathfrak{A}}(t_1^{\mathfrak{A}}[s], \ldots, t_n^{\mathfrak{A}}[s])$ syss $P^{\mathfrak{A}}(t_1^{\mathfrak{A}}[s'], \ldots, t_n^{\mathfrak{A}}[s'])$ syss $\mathfrak{A} \models P(t_1, \ldots, t_n)[s']$.

- H.I. Para cualesquiera dos estados s, s', si $s(x) = s'(x)$ para toda $x \in Vl(\chi)$ y para toda $x \in Vl(\psi)$ entonces $\mathfrak{A} \models \chi[s]$ syss $\mathfrak{A} \models \chi[s']$ y $\mathfrak{A} \models \psi[s]$ syss $\mathfrak{A} \models \psi[s']$.

- Paso Inductivo.

  - $\varphi = \neg\psi$. Tenemos $\mathfrak{A} \models \neg\psi[s]$ syss $\mathfrak{A} \not\models \psi[s] \underset{HI}{syss} \mathfrak{A} \not\models \psi[s']$ syss $\mathfrak{A} \models \neg\psi[s']$.
  - $\varphi = \psi \wedge \chi$. Tenemos $\mathfrak{A} \models (\psi \wedge \chi)[s]$ syss $\mathfrak{A} \models \psi[s]$ y $\mathfrak{A} \models \chi[s] \underset{HI}{syss}$ $\mathfrak{A} \models \psi[s']$ y $\mathfrak{A} \models \chi[s']$ syss $\mathfrak{A} \models (\psi \wedge \chi)[s']$.
  - $\varphi = \forall x\psi$. Primero observemos que $s(x/a)(y) = s'(x/a)(y)$ p.t. $y \in Vl(\psi)$, $a \in |\mathfrak{A}|$, porque $Vl(\psi) = Vl(\varphi) - \{x\}$. Por lo tanto tenemos $\mathfrak{A} \models \forall x\chi[s]$ syss p.t. $a \in |\mathfrak{A}|$, $\mathfrak{A} \models \psi[s(x/a)] \underset{HI}{syss}$ p.t. $a \in |\mathfrak{A}|$, $\mathfrak{A} \models \psi[s'(x/a)]$. Es decir, $\mathfrak{A} \models \forall x\psi[s]$ syss $\mathfrak{A} \models \forall x\psi[s']$.

$\dashv$

Como consecuencia del lema de coincidencia para fórmulas tenemos el siguiente resultado.

Proposición 2.1 *Sea φ un enunciado. Entonces se cumple una y sólo una de las siguientes propiedades.*

(a). $\mathfrak{A} \models \varphi[s]$ para todo estado $s : VAR \longrightarrow |\mathfrak{A}|$.

(b). $\mathfrak{A} \not\models \varphi[s]$. para todo estado $s : VAR \longrightarrow |\mathfrak{A}|$.

dem:

Sea s un estado cualquiera. Si $\mathfrak{A} \models \varphi[s]$ entonces, como φ no tiene varaibles libres, cualquier otro estado s' coincide con s en $Vl(\varphi) = \varnothing$. Por lo tanto $\mathfrak{A} \models \varphi[s']$ y se cumple el caso (a). En caso de que $\mathfrak{A} \not\models \varphi[s]$ se procede de igual forma para obtener (b). $\dashv$

Lema 2.4 *(Lema de Sustitución para Fórmulas).*
Sean $\mathfrak{A}$ una Σ-interpretación, s un estado de las variables, $t \in TERM(\Sigma)$, $\varphi \in FORM(\Sigma)$ y $a = t^{\mathfrak{A}}[s]$. Si $\varphi\{x/t\}$ es admisible entonces

$$\mathfrak{A} \models \varphi\{x/t\}[s] \textit{ syss } \mathfrak{A} \models \varphi[s(x/a)]$$

<u>dem:</u>

Si x no figura libre en φ entonces $\varphi\{x/t\} = \varphi$ y $s(y) = s(x/a)(y)$ para toda $y \in Vl(\varphi)$. Por lo tanto, usando el lema 2.3 obtenemos que $\mathfrak{A} \models \varphi[s]$ syss $\mathfrak{A} \models \varphi[s(x/a)]$ y terminamos. El caso en el que $x \in Vl(\varphi)$ se hace por inducción sobre φ.

- Base de la inducción. Es consecuencia inmediata del lema 2.2 y queda como ejercicio.

- H.I. Supóngase que el lema es válido para ψ y χ.

- Paso Inductivo. Los casos en que $\varphi = \neg\psi$ y $\varphi = \psi \wedge \chi$ son inmediatos de la HI y se dejan como ejercicio.
  Sea $\varphi = \forall z \psi$. Obsérvese que $x \neq z$ pues x figura libre en φ. Como $\varphi\{x/t\}$ es admisible entonces z no figura en t. Además se cumple lo siguiente:

  (a). $\psi\{x/t\}$ es admisible pues $\varphi\{x/t\}$ lo es.
  (b). $\varphi\{x/t\} = (\forall z\psi)\{x/t\} = \forall z(\psi\{x/t\})$ pues $z \neq x$.
  (c). $a = t^{\mathfrak{A}}[s] = t^{\mathfrak{A}}[s(z/b)]$ p.t. $b \in |\mathfrak{A}|$ pues z no figura en t.

  De donde obtenemos que $\mathfrak{A} \models \psi\{x/t\}[s]$ $\underset{(b)}{\textit{syss}}$ $\mathfrak{A} \models \forall z(\psi\{x/t\})[s]$ syss $\mathfrak{A} \models \psi\{x/t\}[s(z/b)]$ p.t. $b \in |\mathfrak{A}|$ $\underset{HI,(c)}{\textit{syss}}$ $\mathfrak{A} \models \psi[s(z/b)(x/a)]$ p.t. $b \in |\mathfrak{A}|$ $\underset{z \neq x}{\textit{syss}}$ $\mathfrak{A} \models \psi[s(x/a)(z/b)]$ p.t. $b \in |\mathfrak{A}|$ syss $\mathfrak{A} \models \forall z\psi[s(x/a)]$ syss $\mathfrak{A} \models \varphi[s(x/a)]$

Corolario 2.1 *Si $\varphi\{x/t\}$ es admisible entonces para cualquier Σ-interpretación $\mathfrak{A}$ y estado s se cumple que $\mathfrak{A} \models (\forall x\varphi \to \varphi\{x/t\})[s]$. Más aún, si φ es una cerradura universal no es necesario que $\varphi\{x/t\}$ sea admisible.*

dem:

Tenemos que mostrar que $\mathfrak{A} \not\models \forall x\varphi[s]$ o $\mathfrak{A} \models \varphi\{x/t\}[s]$. Si $\mathfrak{A} \not\models \forall x\varphi[s]$, terminamos, así que supongamos que $\mathfrak{A} \models \forall x\varphi[s]$ y demostremos que $\mathfrak{A} \models \varphi\{x/t\}[s]$. Por hipótesis tenemos que $\mathfrak{A} \models \varphi[s(x/b)]$ p.t. $b \in |\mathfrak{A}|$. En particular si $b = t^{\mathfrak{A}}[s]$ entonces $\mathfrak{A} \models \vartheta[s(x/a)]$ que es equivalente, mediante el lema 2.4, a $\mathfrak{A} \models \varphi\{x/t\}[s]$.
Supongamos ahora que φ es una cerradura universal, digamos $\varphi = \forall x_1 \ldots \forall x_n\psi$ donde ψ está libre de cuantificadores, $Vl(\psi) = \{x_1, \ldots, x_n\}$ y spg $x \neq x_i$. Vamos a demostrar que $\mathfrak{A} \models \forall x\varphi \to \varphi\{x/t\}[s]$. Nuevamente suponemos $\mathfrak{A} \models \forall x\varphi[s]$ y demostramos que $\mathfrak{A} \models \varphi\{x/t\}[s]$. Tenemos que $\mathfrak{A} \models \varphi\{x/t\}[s]$ syss $\mathfrak{A} \models \psi[s(x_1/b_1) \ldots (x_n/b_n)]$ p.t. $b_1, \ldots, b_n \in |\mathfrak{A}|$. Sea $b = t^{\mathfrak{A}}[s]$, como $s(x_1/b_1) \ldots (x_n/b_n)$ y $s(x_1/b_1) \ldots (x_n/b_n)(x/b)$ coinciden en $Vl(\psi)$ entonces por el lema 2.3 tenemos que $\mathfrak{A} \models \psi[s(x_1/b_1) \ldots (x_n/b_n)]$ syss $\mathfrak{A} \models \psi[s(x_1/b_1) \ldots (x_n/b_n)(x/b)]$ p.t. $b_1, \ldots, b_n \in |\mathfrak{A}|$. Aplicando ahora el lema 2.4 con $\psi, s(x_1/b_1) \ldots (x_n/b_n)$ y t obtenemos que $\mathfrak{A} \models \psi\{x/t\}[s(x_1/b_1) \ldots (x_n/b_n)]$ p.t. $b_1, \ldots, b_n \in |\mathfrak{A}|$ (obsérvese que $\psi\{x/t\}$ es admisible pues ψ es libre de cuantificadores). Pero esto último es equivalente a $\mathfrak{A} \models \forall x\varphi \to \varphi\{x/t\}[s]$. $\dashv$

Ejercicios

2.3.1.- Considerense las siguientes interpretaciones.

> $\mathfrak{A} = \langle \mathbb{Z}, |, Q^{\mathfrak{A}}, +, \cdot, -, -1, 0\rangle$ donde $P^{\mathfrak{A}} =|$ (la relación de divisibilidad), $Q^{\mathfrak{A}}$ es la relación "ser primo", $f^{\mathfrak{A}} = +$, $G^{\mathfrak{A}} = \cdot$, $h^{\mathfrak{A}} = -$, $a^{\mathfrak{A}} = -1$, $b^{\mathfrak{A}} = 0$.
>
> $\mathfrak{B} = \langle \mathbb{Q}, >, Q^{\mathfrak{A}}, +, \cdot, /, \frac{1}{3}, \frac{2}{5}\rangle$ donde $P^{\mathfrak{B}} = >$, $Q^{\mathfrak{B}}$ es la relación "tener denominador primo", $f^{\mathfrak{B}} = \cdot$, $g^{\mathfrak{B}} = /$, $h^{\mathfrak{B}} = +$, $a^{\mathfrak{B}} = \frac{1}{3}$, $b^{\mathfrak{B}} = \frac{2}{5}$.
>
> $\mathfrak{C} = \langle \mathbb{R}, P^{\mathfrak{A}}, Q^{\mathfrak{A}}, +, \cdot, /, \pi, 4\rangle$ donde $P^{\mathfrak{A}}(x, y)$ syss $x + y \in \mathbb{Q}$, $Q^{\mathfrak{A}}(x)$ syss x es irracional, $f^{\mathfrak{A}} = /$, $g^{\mathfrak{A}} = +$, $h^{\mathfrak{A}} = \cdot$, $a^{\mathfrak{A}} = \pi$, $b^{\mathfrak{A}} = 4$,

Contestar con cierto o falso, demostrando o dando un contraejemplo.

(a) $\mathfrak{A} \models \forall x\forall y(Pxy \to Pyx)[s]$ Para cualquier s.

(b) $\mathfrak{A} \models \forall y(P(f(a, a), y) \to \exists z(h(b, y) = f(z, z)))[s]$. Para cualquier s.

(c) $\mathfrak{A} \models \forall x \forall y \forall z(Qx \wedge P(x, g(y, z)) \rightarrow P(x, y) \vee P(x, z))[s]$ Para cualquier s.

(d) $\mathfrak{A} \models (Q(x) \wedge P(x, u))[s]$ para $s(x) = 3, s(u) = 17$.

(e) $\mathfrak{A} \models (Q(h(a, y)) \vee P(a, h(x, y)))[s]$ para $s(x) = 16,\ s(y) = 3$.

(f) $\mathfrak{B} \models \forall x \forall z(Q(x) \wedge z = g(f(a, f(a, a)), x) \rightarrow Q(z)))[s]$ Para cualquier s.

(g) $\mathfrak{B} \models \exists y P(f(b, b), y)[s]$ para cualquier s.

(h) $\mathfrak{B} \models Q(h(a, f(a, b)))[s]$ para cualquier s.

(i) $\mathfrak{B} \models (Q(x) \rightarrow Q(h(x, h(a, h(a, a)))))[s]$ para $s(x) = \frac{7}{9}$.

(j) $\mathfrak{B} \models \exists y \exists x(Q(x) \wedge Q(y) \wedge \neg Q(g(x, y)))[s]$ para cualquier s.

(k) $\mathfrak{C} \models \exists y \exists z \neg P(y, h(z, z))[s]$ para cualquier s.

(l) $\mathfrak{C} \models \exists x \forall y P(x, y)[s]$ para cualquier s.

(m) $\mathfrak{C} \models (P(a, b) \vee Q(f(b, z)))[s]$ para $s(x) = \sqrt{(}2)$.

(n) $\mathfrak{C} \models \exists x P(a, x)[s]$ para cualquier s.

(o) $\mathfrak{C} \models \exists y(Q(y) \wedge \neg P(y, g(x, x)))[s]$ para $s(x) = e$.

2.3.2.- Demuestre que las siguientes condiciones son equivalentes:

(a) $\mathfrak{A} \models (\varphi \rightarrow \psi)[s]$

(b) Si $\mathfrak{A} \models \varphi[s]$ entonces $\mathfrak{A} \models \psi[s]$.

2.3.3.- Demuestre que las siguientes condiciones son equivalentes:

(a) $\mathfrak{A} \models (\varphi \leftrightarrow \psi)[s]$

(b) $\mathfrak{A} \models \varphi[s]$ si y sólo si $\mathfrak{A} \models \psi[s]$.

2.3.4.- Pruebe que $\mathfrak{A} \not\models \neg\varphi[s]$ syss $\mathfrak{A} \models \varphi[s]$.

2.3.5.- Pruebe que φ es satisfacible syss $\exists\varphi$ es satisfacible.

2.4 Modelos y Consecuencia Lógica

Definición 2.6 Sean $\mathfrak{A}$ una Σ-interpretación, φ una fórmula y M un conjunto de fórmulas. Decimos que $\mathfrak{A}$ es *modelo* de φ o que φ es *verdadera* en $\mathfrak{A}$ si $\mathfrak{A} \models \varphi[s]$ para todo estado de las variables s, esto lo denotamos con $\mathfrak{A} \models \varphi$. Decimos que $\mathfrak{A}$ es *modelo* de M si $\mathfrak{A} \models \psi$ para toda $\psi \in M$. Si $\mathfrak{A} \models \neg\varphi$ entonces decimos que φ es *falsa* en $\mathfrak{A}$.

Debe quedar claro que no se le puede dar el valor de verdadero o falso a cualquier fórmula, sin embargo si φ es un enunciado entonces φ si toma necesariamente alguno de los dos valores.

Ejemplo 2.6 Sean $\mathfrak{A} = \langle \mathbb{N}, < \rangle$ y $\mathfrak{B} = \langle \mathbb{Q}, < \rangle$ y considérese la fórmula $\varphi = \forall x \forall y (P(x,y) \to \exists z (P(x,z) \wedge P(z,y))$ que representa la propiedad de densidad de $P^{\mathfrak{A}}$. Por lo tanto tenemos que $\mathfrak{A} \not\models \varphi$ puesto que el orden usual de los naturales no es denso, mientras que $\mathfrak{B} \models \varphi$, pues para cualquier estado s, si $\mathfrak{B} \models P(x,y)[s(x/a)(y/b)]$ donde $a, b \in \mathbb{Q}$ entonces al tomar $d = \frac{a+b}{2}$ se tiene que $\mathfrak{B} \models (P(x,z) \wedge P(z,y))[s(x/a)(y/b)(z/d)]$.

También debemos aclarar que la relación de verdad no tiene las mismas propiedades que la relación de satisfacción, esto se debe a que la relación de verdad involucra una cuantificación implicita de las variables libres de φ. Debemos insistir en esto ya que es error común aplicar ciertas propiedades de la relación de satisfacción a la noción de verdad lo cual resulta incorrecto, como en el siguiente ejemplo.

Ejemplo 2.7 Sean $\mathfrak{A} = \langle \mathbb{N}, P^{\mathfrak{A}}, Q^{\mathfrak{A}} \rangle$ donde $P^{\mathfrak{A}}$ es la relación "ser par" y $Q^{\mathfrak{A}}$ es la relación "ser impar". Considerense las fórmulas $\varphi = P(x)$ y $\psi = Q(x)$. Entonces $\mathfrak{A} \models \varphi \vee \psi$ puesto que cualquier número natural es par o es impar. Sin embargo no se cumple $\mathfrak{A} \models \varphi$ ni $\mathfrak{A} \models \psi$. Puesto que el valor de x no puede ser siempre par o siempre impar.

Otro error común sucede con la negación, si se tiene que $\mathfrak{A} \not\models \varphi$ entonces no se puede concluir que $\mathfrak{A} \models \neg\varphi$. La misma fórmula φ del ejemplo anterior sirve como contraejemplo, pues el hecho de que el valor de x no sea siempre par no implica que siempre sea impar. Tomense estados s, s' con $s(x) = 3$ y $s'(x) = 8$ entonces $\mathfrak{A} \not\models \varphi$ puesto que $\mathfrak{A} \not\models \varphi[s]$ y tampoco se cumple $\mathfrak{A} \models \neg\varphi$ puesto que $\mathfrak{A} \models \varphi[s']$, es decir $\mathfrak{A} \not\models \neg\varphi[s']$.

Veamos ahora algunas propiedades de la relación de verdad.

Proposición 2.2 *Sean $\mathfrak{A}$ una Σ-interpretación y φ una fórmula. Entonces*

$$\mathfrak{A} \models \varphi \textit{ syss } \mathfrak{A} \models \forall\varphi.$$

dem:

$\Rightarrow$) Supogamos que $\mathfrak{A} \models \varphi$, es decir,$\mathfrak{A} \models \varphi[s]$ p.t. $s : VAR \longrightarrow |\mathfrak{A}|$. En particular si $Vl(\varphi) = \{x_1, \ldots, x_n\}$ entonces $\mathfrak{A} \models \varphi[s(x_1/a_1)\ldots(x_n/a_n)]$ p.t.s y cualesquiera $a_1, \ldots, a_n \in |\mathfrak{A}|$. Pero $\mathfrak{A} \models \varphi[s(x_1/a_1)\ldots(x_n/a_n)]$ syss $\mathfrak{A} \models \forall x_n\varphi[s(x_1/a_1)\ldots(x_{n-1}/a_{n-1})]$ syss ... syss $\mathfrak{A} \models \forall x_1 \ldots \forall x_n\varphi[s]$ p.t. s. Por lo tanto $\mathfrak{A} \models \forall\varphi$.
$\Leftarrow$) Supongamos que $\mathfrak{A} \models \forall\varphi$, es decir, $\mathfrak{A} \models \forall\varphi[s]$ p.t. $s : VAR \longrightarrow |\mathfrak{A}|$, por lo tanto tenemos que $\mathfrak{A} \models \varphi[s(x_1/a_1)\ldots(x_n/a_n)]$ p.t. s y para cualesquiera $a_1, \ldots, a_n \in |\mathfrak{A}|$, donde $Vl(\varphi) = \{x_1, \ldots, x_n\}$. En particular se tiene que $\mathfrak{A} \models \varphi[s(x_1/s(x_1))\ldots(x_n/s(x_n))]$ p.t. s, pues $s(x_i) \in |\mathfrak{A}|$. Pero $s(x_1/s(x_1))\ldots(x_n/s(x_n)) = s$. Por lo tanto $\mathfrak{A} \models \varphi[s]$ p.t. s, es decir, $\mathfrak{A} \models \varphi$.
$\dashv$

Proposición 2.3 *Sean $\mathfrak{A}$ una Σ-interpretación y φ, ψ fórmulas. Si $\mathfrak{A} \models \varphi$ y $\mathfrak{A} \models (\varphi \to \psi)$ entonces $\mathfrak{A} \models \psi$.*

dem:

Supongamos que $\mathfrak{A} \models \varphi$ y $\mathfrak{A} \models (\varphi \to \psi)$ y sea $s : VAR \longrightarrow |\mathfrak{A}|$. Como $\mathfrak{A} \models \varphi \to \psi[s]$ entonces $\mathfrak{A} \not\models \varphi[s]$ o $\mathfrak{A} \models \psi[s]$, pero además $\mathfrak{A} \models \varphi[s]$ porque $\mathfrak{A} \models \varphi$. Por lo tanto se tiene que $\mathfrak{A} \models \psi[s]$ y como s era cualquiera concluimos que $\mathfrak{A} \models \psi$. $\dashv$
Uno de los problemas fundamentales en lógica es la busqueda de modelos de un conjunto dado M, tanto desde el punto de vista puramente teorico, al buscar y clasificar modelos de un conjunto en especial y estudiar como son estos modelos entre si, problemas de los que se encarga la rama de la lógica llamada *teoría de modelos*, como desde el punto de vista computacional donde el objetivo principal es tratar de mecanizar el problema, situación que posible en ciertos casos pero que como veremos más adelante es imposible en general.
En este momento nos limitaremos a dar algunos ejemplo sencillos de busqueda de modelos, más adelante podremos ver ejemplos más complicados.

Ejemplo 2.8 Sea $M = \{P(b), Q(b), R(b), \exists x(P(x) \wedge \neg(Q(x) \vee R(x)), \ \forall x(R(x) \rightarrow P(x))\}$ Queremos ver si M tiene un modelo.
En este caso lo más fácil es ir construyendo un modelo para cada fórmula de M, si logramos que todas las fórmulas de M sean verdaderas al mismo tiempo entonces habremos construido un modelo para M. Un modelo de M tiene que ser una estructura de la forma $\mathfrak{A} = \langle A, P^{\mathfrak{A}}, Q^{\mathfrak{A}}, R^{\mathfrak{A}}, b^{\mathfrak{A}}\rangle$. Analicemos cada fórmula de M.

(a). $P(b)$. Para que $\mathfrak{A} \models P(b)$ se debe cumplir que $b^{\mathfrak{A}} \in P^{\mathfrak{A}}$. Así que al menos debemos tener $P^{\mathfrak{A}} = \{b^{\mathfrak{A}}\}$

(b). $Q(b)$. Para que $\mathfrak{A} \models Q(b)$ se debe cumplir que $b^{\mathfrak{A}} \in Q^{\mathfrak{A}}$, así que basta $Q^{\mathfrak{A}} = \{b^{\mathfrak{A}}\}$

(c). $R(b)$. Para que $\mathfrak{A} \models Q(b)$ se debe cumplir que $b^{\mathfrak{A}} \in R^{\mathfrak{A}}$ por lo que basta con $R = \{b^{\mathfrak{A}}\}$.

(d). $\exists x(P(x) \wedge \neg(Q(x) \vee R(x))$. Es decir, hay un elemento de $|\mathfrak{A}|$ tal que cumple P y no cumple Q ni R. Claramente este elemento no puede ser $b^{\mathfrak{A}}$ por lo que $P^{\mathfrak{A}}$ debe tener al menos otro elemento, digamos $P^{\mathfrak{A}} = \{b^{\mathfrak{A}}, a_1\}$.

(e). $\forall x(R(x) \rightarrow P(x))$. Debemos tener que todo elemento de A que cumple $R^{\mathfrak{A}}$, debe cumplir $P^{\mathfrak{A}}$, pero como ha sido definido $R^{\mathfrak{A}}$ esto ya se cumple pues $b^{\mathfrak{A}} \in R^{\mathfrak{A}}$ y $b^{\mathfrak{A}} \in P^{\mathfrak{A}}$.

De manera que $P^{\mathfrak{A}} = \{b^{\mathfrak{A}}, a_1\}$ y $Q^{\mathfrak{A}} = \{b^{\mathfrak{A}}\} = R^{\mathfrak{A}}$ y para que el modelo resulte más natural podemos tomar $A = \{0, 1\}$, $b^{\mathfrak{A}} = 0$ y $a_1 = 1$, con lo que queda $\mathfrak{A} = \langle\{0, 1\}, \{0, 1\}, \{0\}, \{0\}, 0\rangle$.

Veamos otro ejemplo.

Ejemplo 2.9 Sea $M = \{\exists x P(x) \wedge \exists x Q(x), \ \neg\exists x(P(x) \wedge Q(x)), \ \forall x \neg R(x, x), \ \forall x \forall y(R(x, y) \rightarrow R(y, x)), \ \forall x(P(x) \rightarrow \exists y(Q(y) \wedge R(x, y))), \ \forall x(Q(x) \rightarrow \exists y(Q(y) \wedge R(x, y)))\}$ Queremos hallar un modelo de M, veamos cada fórmula.

(a). $\exists x P(x) \wedge \exists x Q(x)$. Es decir, debe existir $a_1 \in |\mathfrak{A}|$ tal que $P^{\mathfrak{A}}(a_1)$ y además debe existir $a_2 \in |\mathfrak{A}|$ tal que $Q^{\mathfrak{A}}(a_2)$. Obsérvese que podriamos tener $a_1 = a_2$.

(b). $\neg\exists x(P(x) \wedge Q(x))$. Está fórmula indica que no hay un individuo que cumpla $P^{\mathfrak{A}}$ y $Q^{\mathfrak{A}}$, de manera que los individuos encontrados en (a) deben ser diferentes.

(c). $\forall x \neg R(x,x)$. Esto es la relación $R^{\mathfrak{A}}$ es antireflexiva, por lo que los pares (a,a) no pueden pertenecer a $R^{\mathfrak{A}}$. Obsérvese que aún no necesitamos que exista un par en la relación.

(d). $\forall x \forall y(R(x,y) \to R(y,x))$. Esta fórmula indica que la relación $R^{\mathfrak{A}}$ es simétrica pero todavia no se exige que haya un par relacionado.

(e). $\forall x(P(x) \to \exists y(Q(y) \wedge R(x,y)))$. Esto significa que para cada elemento a que cumpla $P^{\mathfrak{A}}$ debe existir un elemento b que cumpla $Q^{\mathfrak{A}}$ de manera que el par (a,b) esté en la relación $R^{\mathfrak{A}}$. Como en (a) obtuvimos que $P^{\mathfrak{A}}(a_1)$ entonces debe existir $b \in |\mathfrak{A}|$ tal que $Q^{\mathfrak{A}}(b)$ y $R^{\mathfrak{A}}(a_1,b)$. Obsérvese que podemos tomar $b = a_2$, además, por (d), también debemos tener $R^{\mathfrak{A}}(a_2,a_1)$.

(f). $\forall x(Q(x) \to \exists y(Q(y) \wedge R(x,y)))$. Está fórmula es similar a la anterior. Como tenemos que $Q^{\mathfrak{A}}(a_2)$, por (a), entonces debe existir $a_3 \in |\mathfrak{A}|$ tal que $Q^{\mathfrak{A}}(a_3)$ y $R^{\mathfrak{A}}(a_2,a_3)$, además por (c) tenemos que $a_3 \neq a_2$ y por (b), $a_3 \neq a_1$. Finalmente también debemos tener que $R^{\mathfrak{A}}(a_3,a_2)$ para que se cumpla (d).

De manera que las interpretaciones pueden quedar como sigue:

$$|\mathfrak{A}| = \{1,2,3\}$$

$$P^{\mathfrak{A}} = \{1\}$$

$$Q^{\mathfrak{A}} = \{2,3\}$$

$$R^{\mathfrak{A}} = \{(1,2),(2,1),(2,3),(3,2)\}$$

2.4.1 Validez Universal

Ya hemos visto dos relaciones entre las interpretaciones y las fórmulas, que son la relación de satisfacción y la relación de verdad. Observemos la diferencia entre ellas, la fórmula φ es satisfacible si existe una interpretación $\mathfrak{A}$ y un estado de las variables s que la satisface; la fórmula φ es verdadera o tiene un modelo si existe un interpretación $\mathfrak{A}$ de manera que cualquier estado de las variables satisface a φ. Existe otra relación aún más fuerte en la que φ debe ser verdadera en todas las interpretaciones.

Definición 2.7 Sea φ una Σ-fórmula. Si $\mathfrak{A} \models \varphi$ para toda Σ-interpretación $\mathfrak{A}$ entonces decimos que φ es *universalmente válida* y lo denotamos con $\models \varphi$.

Veamos algunos ejemplos

Ejemplo 2.10 La fórmula $\neg\exists x\varphi \leftrightarrow \forall x\neg\varphi$ es universalmente válida.
Sea $\mathfrak{A}$ una Σ-interpretación, veamos que $\mathfrak{A} \models \neg\exists x\varphi \leftrightarrow \forall x\neg\varphi$. Sea s un estado de las variables, hay 2 casos:

- $\mathfrak{A} \models \neg\exists x\varphi[s]$. Esto es equivalente con: $\mathfrak{A} \not\models \exists x\varphi[s]$ syss no existe $a \in |\mathfrak{A}|$ tal que $\mathfrak{A} \models \varphi[s(x/a)]$ syss para todo $a \in |\mathfrak{A}|$, $\mathfrak{A} \not\models \varphi[s(x/a)]$ syss para todo $a \in |\mathfrak{A}|$, $\mathfrak{A} \models \neg\varphi[s(x/a)]$ syss $\mathfrak{A} \models \forall x\neg\varphi[s]$.

- $\mathfrak{A} \not\models \neg\exists x\varphi[s]$. Es decir, $\mathfrak{A} \models \exists x\varphi[s]$ syss existe $a \in |\mathfrak{A}|$ tal que $\mathfrak{A} \models \varphi[s(x/a)]$ syss existe $a \in |\mathfrak{A}|$, $\mathfrak{A} \not\models \varphi[s(x/a)]$ syss no para todo $a \in |\mathfrak{A}|$, $\mathfrak{A} \models \neg\varphi[s(x/a)]$ syss $\mathfrak{A} \not\models \forall x\neg\varphi[s]$.

Por lo tanto se tiene que $\mathfrak{A} \models \neg\exists x\varphi[s]$ syss $\mathfrak{A} \models \forall x\neg\varphi[s]$. Es decir, $\mathfrak{A} \models \neg\exists x\varphi \leftrightarrow \forall x\neg\varphi$. Por lo tanto, como $\mathfrak{A}$ era arbitraria, se tiene que $\models \neg\exists x\varphi \leftrightarrow \forall x\neg\varphi$.

Ejemplo 2.11 Vamos a demostrar la existencia del dios Bacco, esto es, veremos que existe alguien tal que si él bebe entonces todos beben. Con $B(x)$ queremos decir que x bebe. veremos que la fórmula $\exists x(B(x) \to \forall yB(y))$ es universalmente válida.
Sea $\mathfrak{A}$ una Σ-interpretación y s un estado de las variables, veamos que $\mathfrak{A} \models \exists x(B(x) \to \forall yB(y))[s]$, es decir, $\mathfrak{A} \models (B(x) \to \forall yB(y))[s(x/b)]$, p.a. $b \in |\mathfrak{A}|$. Tenemos dos casos:

- $B^{\mathfrak{A}} = |\mathfrak{A}|$. En este caso, como $B^{\mathfrak{A}}(a)$ p.t. $a \in |\mathfrak{A}|$ se tiene que $\mathfrak{A} \models B(y)[s(y/a)]$ p.t. $a \in |\mathfrak{A}|$. Por lo tanto $\mathfrak{A} \models \forall yB(y)[s]$, en particular, por el lema 2.3, para cualquier $b \in |\mathfrak{A}|$ se tiene que $\mathfrak{A} \models \forall yB(y)[s(x/b)]$. Por lo tanto $\mathfrak{A} \models (B(x) \to \forall yB(y))[s(x/b)]$.

- $B^{\mathfrak{A}} \neq |\mathfrak{A}|$. Sea $b \in |\mathfrak{A}|$ tal que $b \notin B^{\mathfrak{A}}$, es decir $\mathfrak{A} \not\models B(x)[s(x/b)]$ esto implica que $\mathfrak{A} \models (B(x) \to \forall yB(y))[s(x/b)]$.

Es por fórmulas como la anterior que no debe usarse el existencial con la implicación, puesto que nos gustaría que el existencial fuera verdadero si existe alguien que lo haga verdadero, pero en este caso el existencial se cumple aún cuando no exista un individuo que sea testigo de ello.

Un concepto de gran importancia que depende de la validez universal es el de fórmulas lógicamente equivalentes.

Definición 2.8 Sean φ, ψ fórmulas. Decimos que φ es *lógicamente equivalente* a ψ si $\models \varphi \leftrightarrow \psi$. Esto lo denotamos con $\varphi \equiv \psi$.

Las siguientes proposiciones que muestran equivalencias lógicas serán de utilidad más adelante.

Proposición 2.4 *(Leyes de la Negación)*
Se tienen las siguientes equivalencias lógicas.

(a). Doble Negación: $\neg\neg\varphi \equiv \varphi$.

(b). De Morgan: $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$.

(c). De Morgan: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$.

(d). $\neg(\varphi \to \psi) \equiv \varphi \wedge \neg\psi$

(e). $\neg(\varphi \leftrightarrow \psi) \equiv \neg\varphi \leftrightarrow \psi \equiv \varphi \leftrightarrow \neg\psi$.

(f). $\neg\forall x\varphi \equiv \exists x\neg\varphi$

(g). $\neg\exists x\varphi \equiv \forall x\neg\varphi$

dem:

El inciso (g) se demostró en el ejemplo 2.10, el resto se deja como ejercicio al lector. $\dashv$

Proposición 2.5 *(Eliminación de la Implicación y de la Equivalencia)*
Se cumplen las siguientes equivalencias lógicas.

(a). $\varphi \to \psi \equiv \neg\varphi \vee \psi$.

(b). $\varphi \to \psi \equiv \neg(\varphi \wedge \neg\psi)$.

(c). $\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$.

(d). $\varphi \leftrightarrow \psi \equiv (\varphi \wedge \psi) \vee (\neg\varphi \wedge \neg\psi)$.

dem:

Se deja como ejercicio al lector. $\dashv$

Proposición 2.6 *(Eliminación de un cuantificador).*

(a). $\forall x\varphi \equiv \neg\exists x\neg\varphi$.

(b). $\exists x\varphi \equiv \neg\forall x\neg\varphi$.

dem:

Ver el ejercicio 2.4.11 $\dashv$

Proposición 2.7 *(Conmutatividad, Asociatividad y Distributividad)*

(a). $\varphi \vee \psi \equiv \psi \vee \varphi$.

(b). $\varphi \wedge \psi \equiv \psi \wedge \varphi$.

(c). $\varphi \vee (\psi \vee \chi) \equiv (\varphi \vee \psi) \vee \chi$.

(d). $\varphi \wedge (\psi \wedge \chi) \equiv (\varphi \wedge \psi) \wedge \chi$.

(e). $\varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$.

(f). $\varphi \wedge (\psi \vee \chi) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$.

dem:

Ver el ejercicio 2.4.12. $\dashv$

2.4.2 Consecuencia Lógica

En esta sección formalizaremos uno de los conceptos más importantes de la lógica tradicional, que es el concepto de argumento correcto, concepto que aparece en matemáticas en diversas ocasiones, por ejemplo, cualquier demostración tiene que ser un argumento correcto. De hecho para muchos la lógica es el estudio de métodos y patrones de razonamiento en especial del razonamiento que puede expresarse en forma de argumento. Para nuestros propósitos un consiste de una serie de afirmaciones llamados *premisas* y una afirmación llamada *conclusión*; un argumento es *correcto* si siempre que se suponen sus premisas verdaderas, se puede demostrar que su conclusión también es verdadera. Nosotros ya tenemos una noción formal de verdad, pero aún necesitamos una definición formal de argumento correcto, cosa que lograremos en esta sección.

Definición 2.9 Sean M un conjunto de fórmulas y φ una fórmula. Decimos que φ es *consecuencia lógica* de M, lo que se denota con $M \models \varphi$, si todo modelo de M es un modelo de φ. Es decir, Si para toda Σ-interpretación $\mathfrak{A}$, si $\mathfrak{A} \models M$ entonces $\mathfrak{A} \models \varphi$.

Si $M \models \varphi$ tambien decimos que M *implica lógicamente* a φ.
Tenemos las siguientes observaciones:

(a). Como es usual en lógica utilizamos el símbolo $\models$ para denotar dos conceptos distintos:

   (i) La relación "ser modelo de" entre interpretaciones y fórmulas, $\mathfrak{A} \models \varphi$.

   (ii) La relación "ser consecuencia lógica de" entre conjuntos de fórmulas y fórmulas, $M \models \varphi$

   Esto no debe causar confusiones, cuando a la izquierda del símbolo $\models$ se encuentra una interpretación estamos hablando de la primera relación, en cambio, si encontramos un conjunto de fórmulas, nos referimos a la segunda relación.

(b). Si $M = \{\psi\}$ entonces escribimos $\psi \models \varphi$ en lugar de $\{\psi\} \models \varphi$, lo mismo sucede con $\psi_1, \ldots, \psi_n \models \varphi$.

La idea intuitiva de *argumento correcto* queda plasmada en el concepto formal de consecuencia lógica. Un argumento con premisas $\varphi_1, \ldots, \varphi_n$ y conclusión ψ es correcto si y sólo si $\varphi_1, \ldots, \varphi_n \models \psi$, es decir si la conclusión ψ es consecuencia lógica del conjunto de premisas $\{\varphi_1, \ldots, \varphi_n\}$.

Ejemplo 2.12 Si sabemos que todos los gatos maullan y que Cleofas es un gato entonces es correcto concluir que Cleofas maulla. Vamos a demostrar la correctud de este argumento. Las premisas son $\forall x(G(x) \to M(x)),\ G(c)$ y la conclusión es $M(c)$. Vamos a demostrar que:

$$\forall x(G(x) \to M(x)),\ G(c) \models M(c).$$

Sea $\mathfrak{A}$ un modelo de $\forall x(G(x) \to M(x)),\ G(c)$, es decir, $\mathfrak{A} \models \forall x(G(x) \to M(x))$ y $\mathfrak{A} \models G(c)$, veamos que $\mathfrak{A} \models M(c)$.
Sea $s : VAR \longrightarrow |\mathfrak{A}|$, entonces por hipótesis tenemos que p.t. $a \in |\mathfrak{A}|, \mathfrak{A} \models G(c)[s(x/a)]$ y $\mathfrak{A} \models (G(x) \to M(x))[s(x/a)]$. En particular para $c^{\mathfrak{A}}$ se cumple que $\mathfrak{A} \models G(c)[s(x/c^{\mathfrak{A}})]$ y $\mathfrak{A} \models (G(x) \to M(x))[s(x/c^{\mathfrak{A}})]$, es decir, se cumple $G^{\mathfrak{A}}(c^{\mathfrak{A}})$ y además se cumple que $\mathfrak{A} \not\models G(x)[s(x/c^{\mathfrak{A}})]$ o $\mathfrak{A} \models M(x)[s(x/c^{\mathfrak{A}})]$. Por lo tanto $\mathfrak{A} \models M(x)[s(x/c^{\mathfrak{A}})]$, es decir, $M^{\mathfrak{A}}(c^{\mathfrak{A}})$, o equivalentemente $\mathfrak{A} \models M(c)$. De manera que en efecto Cleofas maulla.

Veamos ahora algunas propiedades de la consecuencia lógica.

Proposición 2.8 *Se cumplen las siguientes propiedades.*

(a). Insatisfacibilidad implica Trivialidad. Si $M = \{\varphi, \neg\varphi\}$ entonces $M \models \psi$ para toda fórmula ψ.

(b). Monotonía. Si $M \models \varphi$ y $M \subseteq N$ entonces $N \models \varphi$.

(c). Preservación del Modus Ponens. Si $M \models \varphi \to \psi$ y $M \models \varphi$ entonces $M \models \psi$.

(d). Versión Semántica del Teorema de la Deducción. $M \models \varphi \to \psi$ syss $M, \varphi \models \psi$.

dem:

(a). Sea $M = \{\varphi, \neg\varphi\}$ es claro que M no tiene un modelo de manera que la condición "todo modelo de M es modelo de ψ se cumple por vacuidad al no tener M un modelo. Por lo tanto $M \models \psi$.

(b). Tenemos que mostrar que $N \models \varphi$. Sea $\mathfrak{A}$ un modelo de N, como $M \subseteq N$ entonces $\mathfrak{A}$ es también modelo de M y como $M \models \varphi$ entonces tenemos que $\mathfrak{A} \models \psi$. $\therefore N \models \varphi$.

(c). Sea $\mathfrak{A}$ un modelo de M. Por hipótesis se tiene que $\mathfrak{A} \models \varphi \to \psi$ y $\mathfrak{A} \models \varphi$, por lo que mediante la proposición 2.3 concluimos que $\mathfrak{A} \models \psi$. Por lo tanto $M \models \psi$.

(d). $\Rightarrow$) Supongamos que $M \models \varphi \to \psi$ y sea $\mathfrak{A}$ un modelo de M, φ. Como $\mathfrak{A}$ es modelo de M entonces $\mathfrak{A} \models \varphi \to \psi$ y además $\mathfrak{A} \models \varphi$. Así que por la proposición 2.3 concluimos que $\mathfrak{A} \models \psi$. $\therefore M, \varphi \models \psi$.
$\Leftarrow$) Supongamos que $M, \varphi \models \psi$ y sea $\mathfrak{A}$ un modelo de M. Si $\mathfrak{A} \not\models \varphi$ entonces, por definición tenemos que $\mathfrak{A} \models \varphi \to \psi$, en caso contrario $\mathfrak{A} \models \varphi$, por lo que $\mathfrak{A}$ es modelo de M, φ así que por hipótesis concluimos que $\mathfrak{A} \models \psi$ de donde se siguee que $\mathfrak{A} \models \varphi \to \psi$. Por lo tanto en cualquier caso $\mathfrak{A}$ resulta ser modelo de $\varphi \to \psi$, es decir, $M \models \varphi \to \psi$.

$\dashv$

La formalización de un criterio para decidir la correctud de argumentos no es la única razón por la que la consecuencia lógica juega un papel central en la lógica. En general, obtener las consecuencias lógicas de un conjunto M significa intuitivamente extraer conocimiento que estaba implícito en M, problema de gran importancia en Ciencias de la computación, cuya aplicación más cercana resulta ser la programación lógica.
La extracción de conocimiento a partir de un conjunto de hechos M puede resultar muy simple o sumamente complicada, los casos simples llevan a cierta automatización del proceso de extracción de conocimiento mediante diversas técnicas como la *programación lógica* o el *razonamiento automático*.
Mediante los casos dificiles han llevado a investigar en general si siempre es posible, dada una base de conocimiento M y una fórmula φ contestar a la pregunta ¿ $M \models \varphi$?. Pregunta que contestaremos en la sección 2.7.

Ejercicios

2.4.1.- Demostrar que si $\mathfrak{A} \models \varphi \to \psi$ y $\mathfrak{A} \models \neg\psi$ entonces $\mathfrak{A} \models \neg\varphi$.

2.4.2.- Demostrar que $\mathfrak{A} \models \varphi \wedge \psi$ syss $\mathfrak{A} \models \varphi$ y $\mathfrak{A} \models \psi$.

2.4.3.- Demostrar que si $\mathfrak{A} \models \varphi$ entonces $\mathfrak{A} \not\models \neg\varphi$.

2.4.4.- Demostrar que si $\mathfrak{A} \models \varphi$ entonces $\mathfrak{A} \models \varphi \vee \psi$.

2.4.5.- Sea $\mathfrak{A}$ una interpretación. Sea $T_{\mathfrak{A}}$ el conjunto de todos los enunciados verdaderos en $\mathfrak{A}$, es decir, $T_{\mathfrak{A}} = \{\varphi \mid \varphi$ es enunciado y $\mathfrak{A} \models \varphi\}$. Demostrar que si $T_{\mathfrak{A}} \models \psi$ entonces $\psi \in T_{\mathfrak{A}}$.

2.4.6.- Genere un modelo para cada uno de los siguientes conjuntos.

(a) $M = \{P(a),\ Q(a),\ R(a),\ \exists x\ (P(x) \wedge \neg(Q(x) \vee R(x))),\ \forall x\ (R(x) \to P(x))\}$

(b) $M = \{\exists x\ (P(x) \wedge \neg Q(x)),\ \exists x\ (Q(x) \wedge \neg P(x)),\ \exists z\ (P(z) \wedge Q(z)),\ \forall x\ (P(x) \to \exists y\ (Q(y) \wedge R(x,y))),\ \forall x\ (Q(x) \to \exists y\ (P(y) \wedge R(x,y)))\}$

2.4.7.- Demostrar la validez universal en cada caso.

(a) $\models \forall x\varphi \to \exists\varphi$

(b) $\models \forall x(\varphi \to \exists y\varphi)$

(c) $\models \exists x\exists y\varphi \leftrightarrow \exists y\exists x\varphi$

(d) $\models \exists x\forall y\varphi \to \forall y\exists x\varphi$

2.4.8.- Muestre mediante ejemplos que las siguientes fórmulas no son universalmente válidas.

(a) $\varphi\{x/t\} \to \forall x\varphi$.

(b) $\exists x\varphi \to \varphi\{x/t\}$ donde $\varphi\{x/t\}$ es admisible.

(c) $\exists x\varphi \to \forall x\varphi$.

(d) $\forall x(\varphi \vee \psi) \to \forall x\varphi \vee \forall x\psi$.

2.4.9.- Demostrar la proposición 2.4.

2.4.10.- Demostrar la proposición 2.5.

2.4.11.- Demostrar la proposición 2.6.

2.4.12.- Demostrar la proposición 2.7.

2.4.13.- Demostrar la consecuencia lógica en cada caso:

(a) $\forall x \forall y\ (R(x,y) \to \neg R(y,x)) \models \forall x \neg R(x,x)$

(b) $\forall x \forall y\ (D(x) \ \wedge\ M(y) \to T(x,y)) \models \forall z\ (D(z) \ \wedge\ M(z) \to T(z,z))$

(c) $\forall x\ (C(x) \ \wedge\ \neg \exists y\ P(y,x) \to O(x)),\ C(a),\ \forall z \neg P(z,a) \models O(a)$

2.4.14.- Demostrar la propiedad de contrapositiva: $M, \varphi \models \neg\psi$ syss $M, \psi \models \neg\varphi$.

2.4.15.- Sea $M = \{\varphi_1, \ldots, \varphi_n\}$ tal que $M \models \psi$ y $T \models \varphi_i$ para $1 \leq i \leq n$. Demuestre que $T \models \psi$.

2.4.16.- Demostrar que $M, \varphi, \psi \models \chi$ syss $M, \varphi \wedge \psi \models \chi$.

2.4.17.- Demostrar que $M \models \varphi$ syss $M \cup \{\neg\varphi\}$ no tiene modelo.

2.4.18.- Sea $\mathfrak{A}$ una interpretación. Sea $T_{\mathfrak{A}}$ el conjunto de todos los enunciados verdaderos en $\mathfrak{A}$, es decir, $T_{\mathfrak{A}} = \{\varphi \mid \varphi$ es enunciado y $\mathfrak{A} \models \varphi\}$. Demostrar que si $T_{\mathfrak{A}} \models \psi$ entonces $\psi \in T_{\mathfrak{A}}$.

2.4.19.- Demuestre que $M \models \varphi$ syss $\forall M \models \forall\varphi$. Este hecho muestra que en el proceso de extracción de conocimiento el conjunto de hipótesis y la fórmula que se desea concluir pueden pensarse implicitamente cuantificados universalmente, esto será de gran importancia para la programación lógica.

2.5 Formas Normales

En esta sección desarrollaremos diversas formas normales para fórmulas que son transformaciones de una fórmula que conservan ciertas propiedades semánticas. Las formúlas en forma normal tienen diversas ventajas que veremos más adelante.

Antes de pasar a las formas normales se simplifican las fórmulas mediante el procedimiento conocido como *rectificación*.

Definición 2.10 Una fórmula φ esta *rectificada* si y sólo si se cumplen las siguientes condiciones:

(a). φ no tiene presencias libres y acotadas de una misma variable, es decir, $Vl(\varphi) \cap Vac(\varphi) = \varnothing$.

(b). φ no tiene cuantificadores de la misma variable con alcances ajenos.

(c). φ no tiene cuantificadores que acotan las mismas presencias de una variable (cuantificadores múltiples).

(d). φ no tiene cuantificadores vacuos, es decir, cuantificadores tales que la variable del cuantificador no figure en el alcance del mismo.

Lema 2.5 *Se cumplen las siguientes equivalencias lógicas.*

(a). Eliminación de Cuantificadores Múltiples.

(i) $\forall x \forall x \varphi \equiv \forall x \varphi$.

(ii) $\exists x \exists x \varphi \equiv \exists x \varphi$.

(b). Renombre de variables. Si y *no figura libre en* φ *y* $\varphi\{x/y\}$ *es admisible entonces:*

(i) $\forall x \varphi \equiv \forall y \varphi\{x/y\}$.

(ii) $\exists x \varphi \equiv \exists y \varphi\{x/y\}$.

(c). Eliminación de Cuantificadores Vacuos. Si x *no figura libre en* φ *entonces:*

(i) $\forall x \varphi \equiv \varphi$.

(ii) $\exists x \varphi \equiv \varphi$.

<u>**dem:**</u>

Se deja como ejercicio. $\dashv$

Una consecuencia inmediata de los lemas anteriores es la siguiente proposición.

Proposición 2.9 *Siempre es posible transformar mediante un algoritmo una fórmula dada φ en una fórmula $Rec(\varphi)$ de manera que $Rec(\varphi)$ está rectificada y $Rec(\varphi) \equiv \varphi$.*

<u>**dem:**</u>

Sea φ una fórmula. Si $Vl(\varphi) \cap Vac(\varphi) \neq \varnothing$ podemos aplicar el lema 2.5(b) para renombrar las presencias acotadas de todas las variables que figuraban libres y acotadas en φ, análogamente si existia en φ cuantificadores con alcances ajenos de la misma variable; los cuantificadores múltiples se eliminan mediante el lema 2.5(a) y los cuantificadores vácuos se eliminan con el lema 2.5(c). La fórmula obtenida al terminar estos procesos es $Rec(\varphi)$ y obsérvese que no es única. $\dashv$

El objetivo principal en esta sección es transformar una fórmula dada en una fórmula proposicional que sea en cierto sentido equivalente a la fórmula original. Con las siguientes transformaciones esto se logra parcialmente.

2.5.1 Forma Normal Negativa

El objetivo de esta forma normal es obtener una fórmula equivalente a una fórmula dada de manera que las negaciones solo afecten a fórmulas atómicas.

Definición 2.11 Una fórmula φ está en *forma normal negativa* si las negaciones que figuran en φ afectan solo a fórmulas atómicas.

Proposición 2.10 *Sea φ una fórmula. Podemos encontrar de manera algorítmica una fórmula ψ logicamente equivalente a φ y tal que ψ está en forma normal negativa. En tal caso ψ se denota con $FNN(\psi)$.*

dem:

Dada φ, aplicarle exhaustivamente las leyes de negación (lema 2.4). La fórmula ψ obtenida es $FNN(\varphi)$. $\dashv$

Ejemplo 2.13 Sea $\varphi = \forall x(Px \vee \neg\exists y(Qy \wedge Rxy)) \vee \neg(Py \wedge \neg\forall xPx)$.
La forma normal negativa de $\neg\exists y(Qy \wedge Rxy)$ se obtiene así: $\neg\exists y(Qy \wedge Rxy) \equiv \forall y\neg(Qy \wedge Rxy) \equiv \forall y(\neg Qy \vee \neg Rxy)$.
La forma normal negativa de $\neg(Py \wedge \neg\forall xPx)$ se obtiene así: $\neg(Py \wedge \neg\forall xPx) \equiv \neg Py \vee \neg\neg\forall xPx \equiv \neg Py \vee \forall xPx$.
Por lo tanto tenemos que $FNN(\varphi) = \forall x(Px \vee \ \forall y(\neg Qy \vee \neg Rxy)) \vee (\neg Py \vee \forall xPx)$.

2.5.2 Forma Normal Prenex

El primer paso para eliminar los cuantificadores de una fórmula consiste en "factorizarlos" de manera que todos queden juntos al principio de la fórmula, esto se logran con la llamada forma normal prenex.
El siguiente lema describe la manera de "factorizar" cuantificadores.

Lema 2.6 *Se cumplen las siguientes equivalencias lógicas.*

(a). Si x no figura libre en φ entonces:

(i) $\varphi \wedge \forall x\psi \equiv \forall x(\varphi \wedge \psi)$.

(ii) $\varphi \wedge \exists x\psi \equiv \exists x(\varphi \wedge \psi)$.

(iii) $\varphi \vee \forall x\psi \equiv \forall x(\varphi \vee \psi)$.

(iv) $\varphi \vee \exists x\psi \equiv \exists x(\varphi \vee \psi)$.

(v) $\varphi \to \forall x\psi \equiv \forall x(\varphi \to \psi)$.

(vi) $\varphi \to \exists x\psi \equiv \exists x(\varphi \to \psi)$.

(b). Si x no figura libre en ψ entonces:

(i) $\forall x\varphi \to \psi \equiv \exists x(\varphi \to \psi)$.

(ii) $\exists x\varphi \to \psi \equiv \forall x(\varphi \to \psi)$.

dem:

Se deja como ejercicio. $\dashv$

Definición 2.12 Si ψ es una fórmula de la forma $\psi = Q_1x_1 \ldots Q_nx_n\chi$ donde χ es libre de cuantificadores y para toda i, $Q_i \in \{\forall, \exists\}$ se llama fórmula en *forma normal prenex.*. La cadena de cuantificadores $Q_1x_1 \ldots Q_nx_n$ es el *prefijo* de ψ mientras que χ es la *matriz* de ψ.

Proposición 2.11 *Sea φ una fórmula. Entonces se puede construir mediante un algoritmo una fórmula ψ en forma normal prenex tal que $\varphi \equiv \psi$. A ψ se le denota con $FNP(\varphi)$. Obsérvese que $FNP(\varphi)$ no es única.*

dem:

Dada φ obtener $FNN(Rec(\varphi))$ y aplicarle el lema 2.6 hasta "factorizar" todos sus cuantificadores, la fórmula obtenida es $FNP(\varphi)$. $\dashv$

Ejemplo 2.14 Sea φ la misma fórmula del ejemplo 2.13.
Tenemos $Rec(\varphi) = \forall x(Px \vee \neg\exists v(Qv \wedge Rxv)) \vee \neg(Py \wedge \neg\forall zPz)$.
La forma normal prenex de $\neg\exists v(Qv \wedge Rxv)$ es $\forall v\neg(Qv \wedge Rxv)$.
La forma normal prenex de $\forall x(Px \vee \forall v\neg(Qv \wedge Rxv))$ es $\forall x\forall v(Px \vee \neg(Qv \wedge Rxv))$.
La forma normal prenex de $\neg\forall zPz$ es $\exists z\neg Pz$.
La forma normal prenex de $Py \wedge \exists z\neg Pz$ es $\exists z(Py \wedge \neg Pz)$.
La forma normal prenex de $\neg\exists z(Py \wedge \neg Pz)$ es $\forall z\neg(Py \wedge \neg Pz)$.
Por lo tanto la forma normal prenex de φ es

$$FNP(\varphi) = \forall x\forall v\forall z(Px \vee \neg(Qv \wedge Rxv) \vee \neg(Py \wedge \neg Pz)).$$

Ejemplo 2.15 Sea $\varphi = \forall xPx \to \exists y\forall zRyz$. Entonces

$$\begin{array}{ll} \forall xPx \to \exists y\forall zRyz & \equiv \\ \exists y(\forall xPx \to \forall zRyz) & \equiv \\ \exists y\forall z(\forall xPx \to Ryz) & \equiv \\ \exists y\forall z\exists x(Px \to Ryz). & \end{array}$$

$\therefore \quad FNP(\varphi) = \exists y\forall z\exists x(Px \to Ryz)$.

2.5.3 Forma Normal Conjuntiva

Ya que hemos transformado una fórmula en una forma normal prenex ahora presentamos una manera de simplificar una fórmula proposiciones, en particular nos va a interesar simplificar la matriz de una forma normal prenex, esto se logra mediante la llamada *forma normal conjuntiva* que consiste en expresar cualquier fórmula libre de cuantificadores como una conjunción de disyunciones de literales.

Definición 2.13 Una fórmula libre de cuantificadores φ está en *forma normal conjuntiva* (FNC) si $\varphi = \bigwedge_{i \leq n} \chi_i$ y para cada i, $\chi_i = \bigvee_{j \leq m_i} L_{ij}$ donde para cada i, j, L_{ij} es una literal.

Proposición 2.12 *Cualquier fórmula φ libre de cuantificadores se puede transformar mediante un algoritmo en una fórmula ψ quu está en forma normal conjuntiva tal que $\varphi \equiv \psi$. A ψ se le denota con $FNC(\varphi)$.*

dem:

Utilizando la proposición 2.5 podemos suponer que φ solo contiene negaciones y conjunciones. Si φ es una literal L entonces ya está en FNC pues $L \equiv (L \vee L) \wedge (L \vee L)$. Análogamente si φ es una conjunción de literales entonces φ ya está en FNC, pues $\varphi \equiv \varphi \wedge \varphi$.
En cualquier otro caso mediante distributividades o la siguiente equivalencia, obtenida con distributividad (prop 2.7), se puede obtener la FNC deseada: $(\alpha_1 \vee \ldots \vee \alpha_n) \wedge (\beta_1 \vee \ldots \vee \beta_m) \equiv (\alpha_1 \wedge \beta_1) \vee (\alpha_1 \wedge \beta_2) \vee \ldots (\alpha_1 \wedge \beta_m) \vee \ldots \vee (\alpha_n \wedge \beta_1) \vee (\alpha_n \wedge \beta_2) \vee \ldots (\alpha_n \wedge \beta_m)$. $\dashv$

Ejemplo 2.16 Obtener la forma normal conjuntiva de $\varphi = (\neg P \rightarrow Q) \rightarrow (\neg R \rightarrow S)$.[2]. Primero eliminamos las implicaciones obteniendo $\neg(\neg P \wedge \neg Q) \rightarrow \neg(\neg R \wedge \neg S) \equiv \neg(\neg(\neg P \wedge \neg Q) \wedge \neg\neg(\neg R \wedge \neg S)) \equiv (\neg P \wedge \neg Q) \vee \neg(\neg R \wedge \neg S) \equiv (\neg P \wedge \neg Q) \vee R \vee S \equiv (\neg P \vee R \vee S) \wedge (\neg Q \vee R \vee S) = FNC(\varphi)$.

2.5.4 Forma Normal de Skolem

Ahora nuestro interés se centra en los cuantificadores existenciales y buscamos un posible método para eliminarlos de una fórmula, de manera que se preserve la satisfacibilidad. Para esto utilizaremos el proceso de *skolemización*[3], el cual consiste en substituir las fórmulas existenciales por testigos

[2]Como solo nos interesa la forma, omitimos los argumentos de todos los símbolos de relación

[3]El nombre es debido a su creador, Thoralf Skolem (1887-1963).

de las cuantificaciones existenciales de la fórmula, de la siguiente manera:

$\exists x\varphi$	se substituye por	$\varphi\{x/c\}$
$\forall x_1 \dots \forall x_n \exists y \varphi$	se substituye por	$\forall x_1 \dots \forall x_n \varphi\{y/g(x_1, \dots, x_n)\}$

aquí, c es un símbolo de constante nuevo, llamado *constante de Skolem* y g es un símbolo de función nuevo llamado *función de Skolem*. Con $Sko(\varphi)$ denotamos a la skolemización de φ que se obtiene al eliminar mediante este proceso, de izquierda a derecha, todos los existenciales de la fórmula φ.

Ejemplo 2.17

φ	$sko(\varphi)$
$\exists y R(y)$	$R(c)$
$\forall x \exists y P(x, y)$	$\forall x P(x, g(x))$
$\forall x \forall z \exists y R(x, g(y), f(z))$	$\forall x \forall z R(x, g(h(x, z)), f(z))$

Hasta ahora hemos visto que una fórmula φ es lógicamente equivalente a cualquiera de sus formas normales, situación muy conveniente pues si hallamos un modelo de una de éstas formas normales, éste mismo nos servira como modelo de φ. Desafortunadamente no sucede lo mismo con la skolemización, sin embargo tenemos todavia una relación entre φ y $sko(\varphi)$.

Definición 2.14 Sean φ, ψ dos fórmulas, decimos que φ, ψ son *satisfaciblemente equivalentes* cuando φ es satisfacible si y sólo si ψ es satisfacible. Esta propiedad sera denotada con $\varphi \sim_{sat} \psi$.

Es claro que $\varphi \equiv \psi$ implica que $\varphi \sim_{sat} \psi$, mas no al reves.

Ejemplo 2.18 Claramente $\forall x\ P(x, a) \sim_{sat} \forall x\ P(a, x)$ pero si interpretamos a como 0 y P como $\leq$ sobre $\mathbb{N}$ observamos que $\forall x\ P(x, a) \not\equiv \forall x\ P(a, x)$.

Ejemplo 2.19 $\exists x P(x) \sim_{sat} P(c)$, este ejemplo muestra que una fórmula satisfaciblemente equivalente a la fórmula existencial dada, resulta mucho más fácil de manejar.

Con ayuda del siguiente teorema demostraremos que $\varphi \sim_{sat} sko(\varphi)$.

Teorema 2.1 *(AE). (Forma Normal de Skolem). Sean Σ una signatura, g un símbolo de función que no está en Σ y $\forall x_1 \dots \forall x_n \exists y \varphi$ una Σ-fórmula cerrada tal que $x_1 \dots x_n$ no están acotadas en φ (de manera que la sustitución $\{y/g(x_1 \dots x_n)\}$ es admisible). Sea $\Sigma_g = \Sigma \cup \{g\}$. Entonces todo Σ_g-modelo de $\forall x_1 \dots \forall x_n \varphi\{y/g(x_1 \dots x_n)\}$ es un Σ_g-modelo de $\forall x_1 \dots \forall x_n \exists y \varphi$. Inversamente, todo Σ-modelo de $\forall x_1 \dots \forall x_n \exists y \varphi$ puede expanderse a un Σ_g-modelo de $\forall x_1 \dots \forall x_n \varphi\{y/g(x_1 \dots x_n)\}$.*

dem:

Como se tiene $\models \forall x_1 \dots \forall x_n \varphi\{y/g(x_1 \dots x_n)\} \to \forall x_1 \dots \forall x_n \exists y \varphi$, en particular, para cualquier Σ_g-estructura $\mathfrak{A}$ se tiene que $\mathfrak{A} \models \forall x_1 \dots \forall x_n \varphi\{y/g(x_1 \dots x_n)\} \to \forall x_1 \dots \forall x_n \exists y \varphi$. Si además tenemos que $\mathfrak{A} \models \forall x_1 \dots \forall x_n \varphi\{y/g(x_1 \dots x_n)\}$ entonces necesariamente $\mathfrak{A} \models \forall x_1 \dots \forall x_n \exists y \varphi$. De manera que $\mathfrak{A}$ es también un Σ_g-modelo de $\forall x_1 \dots \forall x_n \exists y \varphi$.
Inversamente. Sea $\mathfrak{A}$ un Σ-modelo de $\forall x_1 \dots \forall x_n \exists y \varphi$. Por definición de verdad tenemos que dados cualesquiera $a_1, \dots, a_n \in \mid \mathfrak{A} \mid$ existe $a \in \mid \mathfrak{A} \mid$ tal que $\mathfrak{A} \models \varphi[s(y/a)]$ para todo estado s tal que $s(x_i) = a_i$. Como no sabemos si el elemento a que existe es único o no tenemos que hacer uso del axioma de elección para escoger uno. Formalmente, mediante AE, definimos una función $G : \mid \mathfrak{A} \mid^n \longrightarrow \mid \mathfrak{A} \mid$ tal que $G(a_1, \dots, a_n) = a$ syss $\mathfrak{A} \models \varphi[s(y/a)]$.
Sea $\mathfrak{B}$ el Σ_g-modelo tal que $\mid \mathfrak{A} \mid = \mid \mathfrak{B} \mid$ que interpreta los símbolos de Σ exactamente igual que $\mathfrak{A}$ y tal que $g^{\mathfrak{B}} = G$. Vamos a demostar que $\mathfrak{B} \models \forall x_1 \dots \forall x_n \varphi\{y/g(x_1 \dots x_n)\}$, para lo cual basta demostrar que, para cualesquiera $a_1, \dots, a_n \in \mid \mathfrak{B} \mid$ y todo estado $s : VAR \longrightarrow \mid \mathfrak{B} \mid$ tal que $s(x_i) = a_i$ se tiene que $\mathfrak{B} \models \varphi\{y/g(x_1 \dots x_n)\}[s]$. Pero mediante el lema de sustitución para fórmulas, puesto que $\{y/g(x_1 \dots x_n)\}$ es admisible, demostrar la última afirmación es equivalente a demostrar que $\mathfrak{B} \models \varphi[s(y/a)]$ donde $a = g(x_1, \dots, x_n)^{\mathfrak{B}} = G(a_1, \dots, a_n)$, pero esto es inmediato por la manera en que se definió a. Por lo tanto $\mathfrak{B}$ es un Σ_g-modelo de $\forall x_1 \dots \forall x_n \varphi\{y/g(x_1 \dots x_n)\}$. $\dashv$

Corolario 2.2 *$\forall x_1 \dots \forall x_n \exists y \varphi$ tiene un modelo syss $\forall x_1 \dots \forall x_n \varphi\{y/g(x_1, \dots, x_n)\}$ tiene un modelo.*

Corolario 2.3 *Si φ es una fórmula entonces $\varphi \sim_{sat} Sko(\varphi)$*

dem:

Es consecuencia inmediata del teorema. $\dashv$
A partir de todos los resultados anteriores podemos construir para cada fórmula su forma normal de Skolem.

Definición 2.15 Una fórmula cerrada de la forma $\forall x_1 \dots \forall x_n \varphi$, donde φ no tiene cuantificadores y está en forma normal conjuntiva es una fórmula en *forma normal de Skolem.*

Teorema 2.2 *Para cada fórmula φ podemos construir efectivamente una fórmula ψ en forma normal de Skolem de manera que $\varphi \sim_{sat} \psi$. ψ se denota con $FNS(\varphi)$.*

dem:

Es consecuencia inmediata de todos los resultados anteriores de esta sección. $\dashv$

Ejemplo 2.20 Sea $\varphi = \forall x \exists y(\exists z \forall w R(x,y,z,w) \to \exists w P(w))$. Primero obtenemos la forma normal prenex de φ, $\varphi \equiv \forall x \exists y \exists v(\exists z \forall w R(x,y,z,w) \to P(v)) \equiv \forall x \exists y \exists v \forall z(\forall w R(x,y,z,w) \to P(v)) \equiv \forall x \exists y \exists v \forall z \exists w(R(x,y,z,w) \to P(v)) = FNP(\varphi)$. Ahora aplicamos el proceso de skolemización a $FNP(\varphi)$:

(a). Eliminamos el primer existencial $\exists y$ mediante la función de skolem $f(x)$, obteniendo $\forall x \exists v \forall z \exists w(R(x,f(x),z,w) \to P(v))$.

(b). Eliminamos el segundo existencial $\exists v$ mediante la función de skolem $g(x)$, obteniendo $\forall x \forall z \exists w(R(x,f(x),z,w) \to P(g(x)))$.

(c). Eliminamos el tercer existencial $\exists w$ mediante la función de skolem $h(x,z)$, obteniendo $\forall x \forall z(R(x,f(x),z,h(x,z)) \to P(g(x)))$.

$\therefore \quad sko(\varphi) = \forall x \forall z(R(x,f(x),z,h(x,z)) \to P(g(x)))$.

Finalmente para obtener $FNS(\varphi)$ basta poner la matriz de $sko(\varphi)$ en forma normal conjuntiva.
$FNC(R(x,f(x),z,h(x,z)) \to P(g(x))) \equiv \neg R(x,f(x),z,h(x,z)) \vee P(g(x))$.

$\therefore \quad FNS(\varphi) = \forall x \forall z(\neg R(x,f(x),z,h(x,z)) \vee P(g(x)))$

2.5.5 Forma Clausular

Por último obtenemos la forma clausular de una fórmula, que consiste simplemente en eliminar los cuantificadores universales en la forma normal de Skolem respectiva.

Definición 2.16 Una *cláusula* es una disyunción de literales.

Con esta definición podemos redefinir una *forma normal conjuntiva* como una conjunción de cláusulas.

Definición 2.17 Si $\varphi = \forall x_1 \ldots \forall x_n \chi$ es una fórmula en forma normal de Skolem entonces χ se le llama la *forma clausular* de φ y se denota con $Cl(\varphi)$.

Observemos que $Cl(\varphi)$ es una conjunción de cláusulas, libre de cuantificadores; además es inmediato que existe un proceso efectivo para construir $Cl(\varphi)$ y que $Cl(\varphi) \sim_{sat} \varphi$. Con esto hemos logrado el objetivo planteado, cada fórmula φ puede transformarse en una fórmula proposicional satisfaciblemente equivalente $Cl(\varphi)$. Más adelante trabajaremos con las formas clausulares que nos ofrecen diversas ventajas y veremos su importancia en la programación lógica.

Ejemplo 2.21 Sea φ la fórmula del ejemplo 2.20. La forma cláusular de φ es

$$Cl(\varphi) = \neg R(x, f(x), z, h(x, z)) \vee P(g(x))$$

Ejercicios

2.5.1.- Obtener las formas normales, FNN, FNP, FNS y la forma clausular de cada una de las siguientes fórmulas:

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

(b) $\forall x (M(x) \to \neg \exists y R(x, y))$.

(c) $\neg \forall x \neg \forall y \neg \forall z P(x, y) \vee \neg \exists x \neg \exists y (\neg \exists z T(x, y, z) \to R(x, y))$.

(d) $\forall x\ (\exists y R(x, y) \wedge \forall y \neg S(x, y) \to \neg \exists y R(x, y) \wedge P(w))$.

(e) $\forall x \forall y\ (P(x, y) \to \exists z Q(z)) \vee \exists x \forall y R(x, y, z)$.

(f) $\neg \forall x \exists y \forall z (R(x, y, z) \leftrightarrow Q(x, z, y))$

(g) $\exists x \forall y S(x, y) \wedge \forall y \exists x \wedge S(y, x)$.

(h) $\neg(\forall x M(x) \vee \exists y \neg O(y)) \vee \forall z M(z) \vee \exists w \neg O(w)$.

(i) $\neg \forall x(\exists y P(x, y) \vee \neg \exists z P(z, x))$

(j) $\neg \forall x(M(x) \vee \exists y \neg O(y)) \vee \forall z M(z) \vee \exists w \neg O(w)$.

(k) $(\forall x(P(x) \rightarrow \exists y R(y)) \vee \neg \exists x(Q(x) \vee \forall z P(x, z))) \wedge \exists v S(v)$.

2.5.2.- Una fórmula libre de cuantificadores φ está en *forma normal disyuntiva* (FND) si $\varphi = \bigvee_{i \leq n} \chi_i$ y para cada i, $\chi_i = \bigwedge_{j \leq m_i} L_{ij}$ donde para cada i, j, L_{ij} es una literal. Desarrollar resultados análogos a los de esta subsección para formas normales disyuntivas.

2.5.3.- Demostrar el lema 2.5.

2.5.4.- Demostrar el lema 2.6.

2.5.5.- Demostrar que la relación $\sim_{sat}$ es una relación de equivalencia.

2.6 Los Teoremas Fundamentales de la Lógica

En esta sección estudiaremos algunos teoremas fundamentales para la lógica de primer orden, tanto desde el punto de vista teórico como desde el punto de vista computacional.

2.6.1 El Teorema de Herbrand

El teorema de Herbrand proporciona una primera relación entre la sintáxis y la semántica al mostrar la equivalencia entre modelos como los definidos anteriormente y cierta clase de modelos especiales que son modelos sintácticos.

Definición 2.18 Sea Σ una signatura con al menos una constante. Una Σ-interpretación $\mathfrak{A}$ es una *interpretación o estructura de Herbrand* syss el universo de $\mathfrak{A}$ es el conjunto de términos cerrados de Σ, es decir,

$$|\mathfrak{A}| = TERM_0(\Sigma) = \{t \mid t \text{ es un } \Sigma\text{-término cerrado}\}$$

la interpretación de los símbolos de constante es $c^{\mathfrak{A}} = c$ y la interpretación de los símbolos funcionales es:

$$f(t_1, \ldots, t_n)^{\mathfrak{A}} = f(t_1, \ldots, t_n)$$

Un *modelo de Herbrand* para un conjunto de fórmulas M es una estructura de Herbrand que es modelo de M.

Obsérvese que lo único que no esta determinado en una estructura de Herbrand es la interpretación de los símbolos de predicado y que los símbolos de constante y de función se interpretan como ellos mismos, por esta razón a los modelos de Herbrand tambien se les llama modelos sintacticos.

El siguiente lema nos dice como se evaluan los términos en una estructura de Herbrand:

Lema 2.7 *Sean Σ una signatura con al menos una constante y $\mathfrak{A}$ una Σ-interpretación de Herbrand. Sean t un término con variables $x_1, \ldots, x_n$ y s un estado de las variables tal que $s(x_i) = r_i$ donde r_i es un término cerrado, para $1 \leq i \leq n$. Entonces:*

$$t^{\mathfrak{A}}[s] = t\{x_1/r_1, \ldots, x_n/r_n\}$$

En particular si t es un término cerrado, $t^{\mathfrak{A}}[s] = t$

dem:

(Por inducción sobre los términos). Ejercicio.

El lema anterior nos dice que la interpretación de términos en una estructura de Herbrand coincide con aplicar una substitución al término y que los términos cerrados se interpretan como ellos mismos.

Definición 2.19 Sean Σ una signatura y M un conjunto de Σ-fórmulas cerradas y universales. El conjunto de instancias cerradas $\varphi\sigma$ donde $\forall x_1 \ldots \forall x_n \varphi$ pertenece a M se llama *conjunto de instancias cerradas* de M y se denota $IC(M)$

A continuación presentamos el Teorema Clásico de Herbrand en una forma moderna:

Teorema 2.3 *(de Herbrand).*
Sean Σ una signatura con al menos una constante y M un conjunto de Σ-enunciados universales. Entonces las siguientes condiciones son equivalentes:

(a). M tiene un modelo.

(b). M tiene un modelo de Herbrand.

(c). $IC(M)$ tiene un modelo.

(d). $IC(M)$ tiene un modelo de Herbrand.

dem:

Debido a que todo modelo de Herbrand es un modelo, las implicaciones $b \Rightarrow a$ y $d \Rightarrow c$ son triviales.
La válidez universal de la fórmula $\forall x_1 \ldots \forall x_n \varphi \to \varphi\{x_1/t_1, \ldots, x_n/t_n\}$ hace inmediatas a las implicaciones $a \Rightarrow c$ y $b \Rightarrow d$. Así que basta probar la implicación $c \Rightarrow b$. Sea $\mathfrak{A}$ un modelo de $IC(M)$ dado. Definimos una estructura de Herbrand $\mathfrak{B}$, para lo cual basta decir como se interpretan los símbolos de predicado, puesto que por definición los símbolos de constante y de función se interpretan como ellos mismos. Si P es un símbolo de predicado n-ario, entonces definimos:

$$P^{\mathfrak{B}} = \{(t_1, \ldots, t_n) \mid \mathfrak{A} \models P(t_1, \ldots, t_n)\}$$

Obsérvese que, por construcción de $\mathfrak{B}$, se cumple que:

$$\mathfrak{A} \models P(t_1, \ldots, t_n) \Leftrightarrow \mathfrak{B} \models P(t_1, \ldots, t_n)$$

es decir, $\mathfrak{A}$ y $\mathfrak{B}$ validan a las mismas fórmulas atómicas cerradas. Dicho resultado se puede extender a cualquier fórmula cerrada libre de cuantificadores mediante inducción sobre fórmulas (ejercicio).
Por último veamos que $\mathfrak{B}$ es modelo de M. Sea $\forall x_1 \ldots \forall x_n \varphi$ una fórmula de M. Queremos demostrar que $\mathfrak{B} \models \forall x_1 \ldots \forall x_n \varphi$, lo cual es equivalente a mostrar que [4] para cualesquiera $t_1, \ldots, t_n \in \mid \mathfrak{B} \mid$, $\mathfrak{B} \models \varphi\{x_1/t_1, \ldots, x_n/t_n\}$. Pero esta última afirmación es equivalente, utilizando el último ejercicio, a la siguiente:

$$\text{para cualesquiera } t_1, \ldots, t_n \in \mid \mathfrak{B} \mid, \;\; \mathfrak{A} \models \varphi\{x_1/t_1, \ldots, x_n/t_n\},$$

la cual es cierta puesto que $\varphi\{x_1/t_1, \ldots, x_n/t_n\}$ pertenece a $IC(M)$ y por hipótesis, $\mathfrak{A}$ es modelo de $IC(M)$. De manera que $\mathfrak{B} \models \forall x_1 \ldots \forall x_n \varphi$ por lo que $\mathfrak{B}$ es un modelo de Herbrand de M. $\dashv$

Sirviendonos de la definición de equisatisfacibilidad 2.14 podemos reescribir parte del teorema de Herbrand como sigue:

Corolario 2.4 *Sean Σ una signatura con al menos una constante y M un conjunto de Σ-enunciados universales. Entonces*

$$M \sim_{sat} IC(M)$$

dem:

Es trivial. $\dashv$

Veamos ahora un resultado que muestra las ventajas de las fórmulas en forma normal conjuntiva o disyuntiva, el cual muestra que las propiedades semánticas de fórmulas proposicionales cerradas se pueden reducir a propiedades sintácticas muy simples.

[4] Recuerdese que $\mid \mathfrak{B} \mid = \{t \mid t \text{ es un } \Sigma - \text{término cerrado}\}$

Definición 2.20 Dos literales de la forma $L, \neg L$ forman un *par complementario.*

Teorema 2.4 *Sean Σ una signatura que contiene al menos una constante y $L_1, L_2, \ldots, L_k$ Σ-literales cerradas. La siguiente tabla caracteriza propiedades semánticas de ciertas fórmulas mediante propiedades sintácticas.*

Forma Sintáctica	*Propiedad Semántica*	*Caracterización Sintáctica*
$L_1 \wedge L_2 \wedge \ldots \wedge L_k$	*Tiene modelo*	$\{L_1, L_2, \ldots, L_k\}$ *no contiene un par complementario.*
$L_1 \wedge L_2 \wedge \ldots \wedge L_k$	*Es universalmente válida*	*Nunca se da el caso*
$L_1 \vee L_2 \vee \ldots \vee L_k$	*Tiene modelo*	*Siempre se da el caso*
$L_1 \vee L_2 \vee \ldots \vee L_k$	*Es universalmente válida*	$\{L_1, L_2, \ldots, L_k\}$ *contiene un par complementario.*

dem:

Tomemos una fórmula del tipo $L_1 \wedge L_2 \wedge \ldots \wedge L_k$. Si el conjunto $\{L_1, L_2, \ldots, L_k\}$ contiene un par complementario digamos L_i y $\neg L_i$ con $i \leq k$ entonces como $L_i \wedge \neg L_i$ no tiene modelo, la fórmula original $L_1 \wedge L_2 \wedge \ldots \wedge L_k$ no tiene modelo.
Inversamente, supongamos que $\{L_1, L_2, \ldots, L_k\}$ no tiene un par complementario, vamos a definir un modelo de Herbrand $\mathfrak{A}$ para $L_1 \wedge L_2 \wedge \ldots \wedge L_k$. Para cada símbolo de predicado $P \in \Sigma$, definimos la interpretación de P como sigue:

$$P^{\mathfrak{A}} = \{(t_1, \ldots, t_n) \in TERM_0(\Sigma)^n \mid \ P(t_1, \ldots, t_n) \in \{L_1, L_2, \ldots, L_k\}\}$$

Ahora veamos que si $1 \leq i \leq k$ entonces $\mathfrak{A} \models L_i$. Hay dos casos:

- $L_i = P(t_1, \ldots, t_n)$. Como L_i es cerrada entonces $t_1, \ldots, t_n \in TERM_0(\Sigma)$. Así que $P(t_1, \ldots, t_n) = L_i \in \{L_1, L_2, \ldots, L_k\}$. Por lo tanto, de la definición de $P^{\mathfrak{A}}$ concluimos que $\mathfrak{A} \models P(t_1, \ldots, t_n)$, es decir,$\mathfrak{A} \models L_i$.

- $L_i = \neg P(t_1, \ldots, t_n)$. Tenemos que $t_1, \ldots, t_n \in TERM_0(\Sigma)$ y $P(t_1, \ldots, t_n) \notin \{L_1, L_2, \ldots, L_k\}$ puesto que por hipótesis $\{L_1, L_2, \ldots, L_k\}$ no contiene un par complementario. Por último de la definición de $P^{\mathfrak{A}}$ se sigue que $\mathfrak{A} \not\models P(t_1, \ldots, t_n)$, es decir, $\mathfrak{A} \models \neg P(t_1, \ldots, t_n)$. Por lo tanto $\mathfrak{A} \models L_i$.

En conclusión se tiene que $\mathfrak{A} \models L_1 \vee \ldots \vee L_k$.
Los restantes casos de la tabla son consecuencia del anterior.

- Una fórmula del tipo $L_1 \wedge L_2 \wedge \ldots \wedge L_k$ nunca es válida pues siempre podemos constuir un modelo para $\neg L_1$, procediendo como en el caso anterior.

- Una fórmula del tipo $L_1 \vee L_2 \vee \ldots \vee L_k$ es válida syss $\neg L_1 \wedge \neg L_2 \wedge \ldots \wedge \neg L_k$ no tiene un modelo syss $\{\neg L_1, \neg L_2, \ldots, \neg L_k\}$ no tiene un par complementario syss $\{L_1, L_2, \ldots, L_k\}$ no tiene un par complementario.

- Una fórmula del tipo $L_1 \vee L_2 \vee \ldots \vee L_k$ siempre tiene un modelo pues, como se hizo arriba, se puede construir un modelo para L_1.

$\dashv$

2.6.2 El Teorema de Compacidad

En esta sección enunciamos el teorema de Compacidad, el cual es de importancia crucial para la automatización del razonamiento, ya que permite reducir el problema de la satisfacibilidad de un conjunto arbitrario de fórmulas a sus subconjuntos finitos, situación de gran importancia desde el punto de vista computacional en el que resulta imposible manejar objetos infinitos. Por otra parte el teorema de compacidad es una de las herramientas principales para la teoría de modelos.
Enunciamos el teorema para el caso de fórmulas cerradas y libres de cuantificadores, es decir, fórmulas sin variables.

Teorema 2.5 *(AE) (Compacidad para conjuntos de fórmulas cerradas y libres de cuantificadores).*
Sea Σ una signatura y M un conjunto de Σ-fórmulas cerradas y libres de cuantificadores. Entonces, M tiene un modelo si y sólo si todo subconjunto finito de M tiene un modelo.

<u>**dem:**</u>

La demostración, que utiliza el lema de König dependiente del axioma de elección, puede verse en [SpAn91]. Una demostración clásica, también dependiente del axioma de elección, siguiendo el método de Henkin, puede verse en [Am99]. $\dashv$

Ahora, a partir de los teoremas de Herbrand y de Compacidad restringido, demostramos el teorema general de Compacidad.

Teorema 2.6 *(AE) (Teorema de Compacidad).*
Sean Σ una signatura y M un conjunto de Σ-fórmulas cerradas. Entonces M tiene modelo si y sólo si todo subconjunto finito de M tiene modelo.

<u>**dem:**</u>

$\Rightarrow$) Es trivial.
$\Leftarrow$) Supongamos que M no tiene modelo.
Sea $M' = FNS(M)$, el conjunto de formas normales de Skolem de los elementos de M, donde para cada $\varphi \in M$ se usó un juego diferente de funciones de Skolem. Puesto que $\varphi \sim_{sat} FNS(\varphi)$, se tiene que M' no tiene modelo. Por el teorema de Herbrand, concluimos que $IC(M')$ no tiene modelo y como $IC(M')$ consta de fórmulas cerradas y libres de cuantificadores, utilizando el teorema 2.5, concluimos que existe un subconjunto finito $E' \subseteq M'$ tal que E' no tiene modelo. Pero tal E', tiene que ser de la forma $E' = FNS(E)$, donde E es algún subconjunto finito de M. De manera que E no puede tener modelo. $\dashv$

Corolario 2.5 *Sean Σ una signatura y M un conjunto de Σ-fórmulas. Entonces, M es satisfacible si y sólo si todo subconjunto finito de M es satisfacible.*

<u>**dem:**</u>

Utilizar que $\varphi \sim_{sat} \exists\varphi$ y después Skolemizar M. $\dashv$

El siguiente corolario, proporciona otra forma del teorema de Herbrand.

Corolario 2.6 *Sean Σ una signatura con al menos una constante y M un conjunto de Σ-fórmulas cerradas y universales. Sea $\exists x\psi$ una fórmula cerrada, con ψ libre de cuantificadores. Las siguientes condiciones son equivalentes:*

(a). $M \models \exists x\psi$

(b). Existe un número finito de términos cerrados, $t_1, \ldots, t_n$ tales que $M \models \psi\{x/t_1\} \vee \ldots \vee \psi\{x/t_n\}$

dem:

(b)$\rightarrow$ (a). Es inmediato utilizando que $\varphi\{x/t\} \models \exists x\varphi$.

(a)$\rightarrow$ (b). Supóngase que $M \models \exists x\psi$. Entonces $M \cup \{\neg\exists x\psi\}$ no tiene modelo, es decir, $M \cup \{\forall x\neg\psi\}$ no tiene modelo. Por el teorema de compacidad (2.6) existe un subconjunto finito de $M \cup \{\forall x\neg\psi\}$ que no tiene modelo, además como $M \cup \{\forall x\neg\psi\}$ consta de fórmulas universales y cerradas entonces por el teorema de Herbrand (2.3) existe un subconjunto finito de $IC(M \cup \{\forall x\neg\psi\})$ que no tiene modelo. Tal conjunto es de la forma $M' \cup \{\neg\psi\{x/t_1\}, \ldots, \neg\psi\{x/t_n\}\}$, donde $M' \subseteq IC(M)$. Ahora bien, como para cualquier $\varphi \in IC(M)$ se tiene que $M \models \varphi$ entonces $M \cup \{\neg\psi\{x/t_1\}, \ldots, \neg\psi\{x/t_n\}\}$ tampoco tiene un modelo, pero esto sucede syss $M \cup \{\neg\psi\{x/t_1\} \wedge \ldots \wedge \neg\psi\{x/t_n\}\}$ no tiene un modelo, syss $M \models \neg(\neg\psi\{x/t_1\} \wedge \ldots \wedge \neg\psi\{x/t_n\})$ y esto es equivalente mediante las leyes de negación a $M \models \psi\{x/t_1\} \vee \ldots \vee \psi\{x/t_n\}$.

$\dashv$

2.6.3 El Teorema de Löwenheim-Skolem

La importancia del teorema de Löwenheim-Skolem es más bien teórica, aquí solo presentamos un versión simple.

Teorema 2.7 *(Löwenheim-Skolem)*
Si un conjunto de fórmulas M tiene un modelo entonces M tiene un modelo cuyo universo es a lo más numerable.

dem:

Supongamos que M tiene un modelo. Consideremos $FNS(M)$. Como $M \sim_{sat} FNS(M)$ entonces $FNS(M)$ tiene un modelo, por lo tanto por el teorema de Herbrand (2.3) $FNS(M)$ tiene un modelo de Herbrand, pero obsérvese que el universo de Herbrand es a lo más numerable. Pero por el teorema 2.1 cualquier modelo de $FNS(M)$ es modelo de M. Por lo tanto M tiene un modelo a lo más numerable. $\dashv$

Ejercicios

2.6.1.- Resuelva los 2 ejercicios que se dejaron en la prueba del teorema de Herbrand.

2.6.2.- Sea φ una Σ-fórmula y $\mathfrak{A}$ una Σ-estructura de Herbrand. Pruebe que:

(a) $\mathfrak{A} \models \forall x\varphi$ syss para cada $a \in |\mathfrak{A}|$, $\mathfrak{A} \models \varphi\{x/a\}$

(b) $\mathfrak{A} \models \exists x\varphi$ syss para alguna $a \in |\mathfrak{A}|$, $\mathfrak{A} \models \varphi\{x/a\}$

2.6.3.- Sea $\Sigma = \{c, R^{(1)}\}$. Muestre que el conjunto de Σ-fórmulas $M = \{R(c), \exists x\neg R(x)\}$ tiene un modelo pero no tiene un Modelo de Herbrand.

2.6.4.- Sean $L_1, \ldots, L_k$ literales que contienen variables. Muestre que $L_1 \wedge \ldots \wedge L_k$ es satisfacible syss $\{L_1, \ldots, L_k\}$ no contiene un par complementario. Sugerencia: Skolemizar y usar el teorema 2.4.

2.7 El Problema de la Decisión

El área de la lógica matemática caracterizada por el término *problema de la decisión* se origino a principios del siglo XX. En 1900 Hilbert formuló el problema de encontrar un algoritmo que decida la validez universal de las fórmulas de la lógica de predicados de pirmer orden, él le llamo a este problema el "Problema fundamental de la lógica matemática".
A principios del siglo el problema parecia una mera custión de invención matemática, de hecho se logro algún progreso al obtener procesos de decisión para ciertas clases de fórmulas de la lógica de predicados, como la clase monádica que consiste de fórmulas que solo tienen predicados unarios y términos sin símbolos de función, cuyo proceso de decisión se resolvio afirmativamente en 1915.

En esta sección presentamos una muy breve introducción a la *teoría de computabilidad*, rama de la lógica que se dedica averiguar que problemas son calculables y cuales no; éste es un problema básico tanto en Ciencias de la Computación como en lógica matemática o matemáticas en general. De manera más específica, si está dado un problema calculable la pregunta es si habra un proceso efectivo para resolverlo. Con esto queremos decir que hay una computadora que puede resolver el problema en un tiempo finito.
En esta sección vamos a mostrar que el problema original de Hilbert no tiene solución, es decir, es indecidible. De hecho probaremos algo más fuerte, que es la indecidibilidad de la consecuencia lógica. Antes de esto damos las definiciones previas necesarias.

Definición 2.21 Un *alfabeto* $\mathcal{A}$ es un conjunto no vacío de símbolos.

Definición 2.22 Una *cadena* es una sucesión finita de símbolos de un alfabeto dado. A las cadenas también se les llama *palabras*

Un conjunto de gran importancia al estudiar cadenas es la *estrella de Kleene* de un alfabeto dado, definida como sigue:

Definición 2.23 Sea $\mathcal{A}$ un alfabeto. El conjunto de todas las cadenas obtenidas a partir de $\mathcal{A}$ se llama la *estrella de Kleene* de $\mathcal{A}$ y se denota $\mathcal{A}^\star$. Este conjunto contiene además una cadena especial denotada ϵ tal que $\epsilon \notin \mathcal{A}$, llamada la *cadena vacía*.

Definición 2.24 Un conjunto $\mathbb{S} \subseteq \mathcal{A}^\star$ de cadenas es *decidible* si existe un procedimiento efectivo tal que dada una cadena $\omega \in \mathbb{S}$, el procedimiento decide si $\omega \in \mathbb{S}$ o $\omega \notin \mathbb{S}$.
$\mathbb{S}$ es *indecidible* si no es decidible.

Para dejar firme a la definición de indecidibilidad debemos indicar que operaciones se permiten en un procedimiento. Existen diversas maneras de formular manera matemática que es lo que puede ser calculado. En 1936 Alan Turing definio una máquina abstracta que ahora se conoce como *máquina de Turing* y mostro que todo lo que puede ser calculado de manera intuitiva también puede calcularse con una maquina de Turing.

Definición 2.25 Un conjunto $\mathbb{S}$ de cadenas sobre un alfabeto $\mathcal{A}$ es *recursivamente numerable* si existe un procedimiento efectivo que produce la respuesta "si" para cada $\omega \in \mathcal{A}^\star$ en caso de que $\omega \in \mathbb{S}$.

Obsérvese que si un conjunto $\mathbb{S}$ es recursivamente numerable y se tiene una cadena $\omega \notin \mathbb{S}$ no se sabe como se comporta el procedimiento de numerabilidad recursiva, en la mayoría de estos casos este proceso no termina.

2.7.1 Semidecidibilidad de la Consecuencia Lógica

En esta sección presentamos un proceso de semidecisión para la consecuencia lógica.
Sean $\varphi_1, \ldots, \varphi_n, \varphi$ fórmulas, queremos contestar a la pregunta

$$¿\, \varphi_1, \ldots, \varphi_n \models \varphi \,?$$

Para esto podemos reducir el problema de la siguiente manera:

$$\varphi_1, \ldots, \varphi_n \models \varphi \text{ syss } \{\varphi_1, \ldots, \varphi_n, \neg\varphi\} \text{ no tiene modelo}$$

Aplicando el teorema 2.2 tenemos que

$$\{\varphi_1, \ldots, \varphi_n, \neg\varphi\} \sim_{sat} FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\})$$

Ahora mediante el teorema de Herbrand tenemos que

$$FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}) \sim_{sat} IC(FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}))$$

Por lo tanto con ayuda de la transitividad de $\sim_{sat}$ obtenemos que

$$\varphi_1, \ldots, \varphi_n \models \varphi \text{ syss } IC(FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\})) \text{ no tiene modelo}$$

Hasta este punto todo el proceso es algorítmico. Ahora por el teorema de compacidad

$IC(FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}))$ no tiene modelo syss existe un subconjunto finito M de $IC(FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}))$ tal que M no tiene modelo. Es en este momento donde se pierde el proceso algorítmico pues el teorema de compacidad solo asegura la existencia del subconjunto finito pero la prueba no es constructiva, es decir, no sabemos a ciencia cierta cual es el subconjunto finito. El proceso que sigue también es algorítmico.
Obsérvese que las fórmulas de $IC(FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}))$ son fórmulas cerradas libres de cuantificadores, de hecho son formas normales conjuntivas cuyos conyuntos son cláusulas. Digamos que $M = \{\psi_1, \ldots, \psi_m\}$, entonces

$$\{\psi_1, \ldots, \psi_m\} \sim_{sat} \psi_1 \wedge \ldots \wedge \psi_m$$

por lo que basta ver si $\psi_1 \wedge \ldots \wedge \psi_m$ no tiene modelo. Para esto nos servimos del ejercicio 2.5.2 y calculamos $FND(\psi_1 \wedge \ldots \wedge \psi_m)$.

Sea $FND(\psi_1 \wedge \ldots \wedge \psi_m) = \chi_1 \vee \ldots \vee \chi_k$ entonces

$$\chi_1 \vee \ldots \wedge \chi_k \text{ no tiene modelo syss ninguna } \chi_i \text{ tiene modelo}$$

Sea $\chi_i = L_{i1} \wedge \ldots \wedge L_{im_i}$, mediante el teorema 2.4 tenemos que

$$\begin{array}{l} L_{i1} \wedge \ldots \wedge L_{im_i} \text{ no tiene modelo syss} \\ \{L_{i1}, \ldots, L_{im_i}\} \text{ contiene un par complementario} \end{array}$$

En conclusión se tienen dos procesos algorítmicos:

- El proceso de transformación de $\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}$ a $IC(FNS(\{\varphi_1, \ldots, \varphi_n, neg\varphi\}))$.
- El proceso para decidir si el subconjunto finito $M \subseteq IC(FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}))$ tiene modelo.

Lo único que impide que el proceso completo sea un algoritmo es el uso del teorema de compacidad. Es por esto que no podemos hablar de decidibilidad sin embargo si tenemos la recursividad numerable.

Teorema 2.8 *(Numerabilidad Recursiva de la Consecuencia Lógica)*
Sea Σ una signatura finita. Existe un procedimiento efectivo que genera todas las tuplas $(\varphi_1, \ldots, \varphi_n, \varphi)$ tales que $\varphi_1, \ldots, \varphi_n \models \varphi$. Es decir, el conjunto de tales tuplas forma un conjunto recursivamente numerable.

<u>**dem:**</u>

Sea $(\varphi_1, \ldots, \varphi_n, \varphi)$ una tupla de fórmulas. Enumerense los subconjuntos finitos de $IC(FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}))$ hasta hallar uno que no tenga modelo. Si este proceso termina entonces $\varphi_1, \ldots, \varphi_n \models \varphi$.
Si $\varphi_1, \ldots, \varphi_n \not\models \varphi$ entonces el proceso no termina pues todos los subconjuntos finitos de $IC(FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}))$ tienen modelo. $\dashv$

Para terminar el capítulo mostraremos que este proceso de enumeración es lo más cercano a un algoritmo que podemos obtener para la consecuencia lógica. Es decir, es imposible obtener un proceso efectivo que también termine en los casos en que $IC(FNS(\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}))$.

2.7.2 Indecidibilidad de la Consecuencia Lógica

El teorema de indecidibilidad de Church muestra que la consecuencia lógica y por lo tanto la validez universal de una fórmula, es indecidible, para demostrarlo nos basaremos en el teorema de correspondencia de Post el cual nos limitamos a enunciar.

Definición 2.26 Considérese el siguiente problema: Dada una lista finita $[(a_1, b_1), \ldots, (a_k, b_k)]$ de parejas de cadenas no vacías sobre el alfabeto $\{0, 1\}$, encontrar una sucesión de índices $i_1, \ldots, i_n$ tal que $a_{i_1} \ldots a_{i_n} = b_{i_1} \ldots b_{i_n}$.
La lista $[(a_1, b_1), \ldots, (a_k, b_k)]$ se llama *sistema de correspondencia de Post*, la sucesión de índices es una *solución del sistema* y el problema se conoce como el *problema de Correspondencia de Post*.

Ejemplo 2.22 Considerese el sistema $[(1, 101), (10, 00), (011, 11)]$. Entonces una solución para el sistema es la sucesión $1, 3, 2, 3$ puesto que $a_1 a_3 a_2 a_3 = 101110011 = b_1 b_3 b_2 b_3$.

Teorema 2.9 *(Post)*
El problema de la correspondencia de Post es indecidible, es decir, no existe un algoritmo que reciba como entrada un sistema de correspondencia $[(a_1, b_1), \ldots, (a_k, b_k)]$ y termine con la respuesta "si" si el sistema dado tiene solución y con "no" en otro caso.

Nuestro interes está en investigar si una fórmula dada es universalmente válida, esto lo haremos con un método muy usado en teoría de la computabilidad que consiste en reducir un problema a uno ya conocido.

Definición 2.27 Sean $\mathbb{C}, \mathbb{D}$ dos conjuntos de cadenas sobre un alfabeto finito $\mathcal{A}$. decimos que $\mathbb{C}$ es *reducible* a $\mathbb{D}$ si existe una función $F : \mathcal{A}^\star \longrightarrow \mathcal{A}^\star$ tal que:

- Existe un algoritmo que calcula a F.
- Para cada $\omega \in \mathcal{A}^\star$, $\omega \in \mathbb{C}$ syss $F(\omega) \in \mathbb{D}$.

La función F se llama reducción de $\mathbb{C}$ a $\mathbb{D}$.

Así que si $\mathbb{C}$ es reducible a $\mathbb{D}$ mediante F entonces la pregunta ¿ $\omega \in \mathbb{C}$? puede contestarse al responder la pregunta equivalente ¿ $F(\omega) \in \mathbb{D}$?. También debe quedar claro el siguiente hecho:

> Si $\mathbb{C}$ es reducible a $\mathbb{D}$ entonces si $\mathbb{C}$ es decidible (recursivamente numerable) entonces $\mathbb{D}$ es decidible (recursivamente numerable).

En este hecho se basa la demostración del siguiente teorema.

Teorema 2.10 *(Teorema de Indecidibilidad de Church)*
Sea Σ una signatura finita que consta de una constante c, dos símbolos de función f_0, f_1 y un símbolo de predicado P. El conjunto de Σ-fórmulas universalmente válidas es indecidible.

dem:

La idea es definir un procedimiento efectivo que reduzca el problema de correspondencia de Post al problema de validez universal de fórmulas.
El primer paso consiste en formalizar el problema de Post en lógica de predicados, para esto usamos la signatura dada.
Dado un término t y una sucesión de índices $i_1, \ldots, i_m \in \{0,1\}^\star$. Utilizamos la abreviatura $f_{i_1,\ldots,i_m}(t)$ para el término $f_{i_1}(f_{i_2}(\ldots f_{i_m}(t)\ldots))$. Sea $\mathcal{C} = [(a_1, b_1), \ldots, (a_k, b_k)]$ un sistema de correspondencia. Considerense las siguiente fórmulas:

$$\varphi_i = P(f_{a_1}(c), f_{b_1}(c)) \wedge \ldots \wedge P(f_{a_k}(c), f_{b_k}(c)).$$

$\varphi_2 = \forall x \forall y\ (P(x,y) \to P(f_{a_1}(x), f_{b_1}(y)) \wedge \ldots \wedge P(f_{a_k}(x), f_{b_k}(y)))$.

$\varphi_3 = \exists x P(x,x)$.

Vamos a demostrar que

$$\mathcal{C} \text{ tiene solución syss } \models \varphi_1 \wedge \varphi_2 \to \varphi_3 \quad (\diamond)$$

Para esto definimos una Σ-interpretación $\mathfrak{A}_{\mathcal{C}} = \langle \{0,1\}^{\star},\ \mathcal{P}, F_0, F_1, \epsilon \rangle$, donde

$\mathcal{P}(u,v)$ syss existen $n > 0$ e $i_1, \ldots, i_n \leq k$ tales que $u = a_{i_1} \ldots a_{i_n}$ y $v = b_{i_1} \ldots b_{i_n}$.

$F_0(u) = 0u,\ F_1(u) = 1u$.

ϵ es la palabra vacía.

Es fácil ver que $\mathfrak{A}_{\mathcal{C}} \models \varphi_1 \wedge \varphi_2$. Además $\mathfrak{A}_{\mathcal{C}} \models \varphi_3$ syss $\mathcal{C}$ tiene solución.

Demostramos ahora $(\diamond)$.
$\Leftarrow$) Esta dirección es clara por la definición de $P^{\mathfrak{A}_{\mathcal{C}}} = \mathcal{P}$.
$\Rightarrow$) Sean $\mathcal{C}$ un sistema con solución $j_1, \ldots, j_n$ y $\mathfrak{A} = \langle A, \mathcal{P}', F'_o, F'_1, \epsilon' \rangle$ una Σ-interpretación tal que $\mathfrak{A} \models \varphi_1 \wedge \varphi_2$. Vamos a demostrar que $\mathfrak{A} \models \varphi_3$. Utilizamos la misma abreviatura citada arriba con F'_0 y F'_1, obsérvese que $F'_u(F'_v(a)) = F'_{uv}(a)$.
Como $\mathfrak{A} \models \varphi_1 \wedge \varphi_2$ entonces tenemos

$$\mathcal{P}'(F'_{a_{j_1}}(\ldots(F'_{a_{j_n}}(\epsilon')\ldots)), F'_{b_{j_1}}(\ldots(F'_{b_{j_n}}(\epsilon')\ldots)))$$

es decir,

$$\mathcal{P}'(F'_{a_{j_1}\ldots a_{j_n}}(\epsilon'), F'_{b_{j_1}\ldots b_{j_n}}(\epsilon'))$$

y como $a_{j_1} \ldots a_{j_n} = b_{j_1} \ldots b_{j_n}$ entonces se tiene $\mathfrak{A} \models \exists x P(x,x)$, es decir, $\mathfrak{A} \models \varphi_3$.
Por lo tanto $(\diamond)$ se cumple y por el teorema 2.9 concluimos que el conjunto de Σ-fórmulas universalmente válidas es indecidible. $\dashv$

Corolario 2.7 *El conjunto de parejas (Σ, φ) donde Σ es una signatura finita y φ es una Σ-fórmula universalmente válida es indecidible. Por lo tanto el conjunto de todas las tuplas $(\varphi_1, \ldots, \varphi_n, \varphi)$ de Σ-fórmulas donde Σ es una signatura finita y $\varphi_1, \ldots, \varphi_n \models \varphi$ también es indecidible.*

dem:

Es consecuencia inmediata del teorema 2.10 $\dashv$

Corolario 2.8 *Sea Σ una signatura finita. Entonces es indecidible si una Σ-fórmula es satisfacible.*

dem:

Es inmediato al reducir el problema de la satisfacibilidad de φ al problema de la no validez universal de $\neg\exists\varphi$. $\dashv$

Bibliografía

[Am99] J.A. Amor. Compacidad en Lógica de Primer Orden y su relación con el Teorema de Completud. Facultad de Ciencias UNAM. 1999.

[AmMi98] J. A. Amor, F. E. Miranda. Demostración Automática de Teoremas y la Nutria. Por aparecer en Aportaciones Matemáticas, Memorias del XXX Congreso Nacional de la SMM. SMM México 1998.

[GrTr96] W.K. Grassmann, J.P. Tremblay. Logic and Discrete Mathematics, A Computer Science Perspective. Prentice Hall 1996.

[Ku92] S. Kumar. Deductive Databases and Logic Programming. Addison-Wesley 1992.

[Lei97] A. Leitsch. The Resolution Calculus. Springer Verlag. 1997.

[LLo87] J.W. Lloyd. Foundations of Logic Programming, second edition. Springer-Verlag. 1987.

[MeNe96] G. Metakides, A. Nerode. Principles of Logic and Logic Programming. Studies in Computer Science and Artificial Intelligence 13. North-Holland. 1996.

[Mi99] F. Miranda P. Fundamentos Lógicos del Programa de Razonamiento Automático OTTER. Tesis de Maestria, Facultad de Ciencias UNAM 1999.

[SpAn91] V. Sperschneider, G. Antoniou. Logic, A Foundation for Computer Science. Addison-Wesley. 1991.

[WOLB92] L. Wos, R. Overbeek, E. Lusk, J. Boyle. Automated Reasoning, Introduction and Applications. McGraw-Hill. 1992.

Índice