

Notas sobre lógica y semántica declarativa de la programación lógica

Marco Trillo

19 de abril de 2009

Resumen

Este documento introduce de forma breve algunos conceptos de la lógica matemática, comenzando en la lógica de proposiciones, para pasar después a la lógica de predicados y, por último, presentar la semántica declarativa básica de la programación lógica “pura”.

Índice

1. Lógica de proposiciones	2
1.1. Átomos	2
1.2. Fórmulas	2
1.3. Semántica	3
1.3.1. Semántica de los átomos	3
1.3.2. Semántica de las fórmulas	3
1.3.3. Propiedades	4
1.3.4. Fortaleza y debilidad de fórmulas	5
1.4. Consecuencia y equivalencia lógica	5
1.4.1. Algunas equivalencias lógicas	5
1.4.2. Deducción: sistemas deductivos	7
2. Lógica de predicados	7
2.1. Alfabeto	8
2.2. Término	8
2.3. Fórmula	9
2.4. Semántica	9
2.4.1. Interpretación	9
2.4.2. Valuación	10
2.4.3. Semántica de los términos	10
2.4.4. Semántica de las fórmulas	10
2.4.5. Variables libres y ligadas	11
2.4.6. Ejemplo	12
2.4.7. Propiedades	12
2.4.8. Algunas equivalencias lógicas	13

3. Programación lógica	13
3.1. Cláusulas de Horn	14
3.2. Universo de un programa lógico	14
3.2.1. Universo de Herbrand	14
3.2.2. Base de Herbrand	15
3.2.3. Interpretación de Herbrand	15
3.2.4. Modelo de Herbrand	15
3.3. Modelos de Herbrand de un programa lógico	16
3.3.1. Sustitución, instancia y programa base	16
3.3.2. Formulación de un modelo de Herbrand	17
3.4. El menor modelo de Herbrand	17
3.4.1. Diferentes modelos de Herbrand de un programa lógico	17
3.4.2. Construcción “desde arriba”: intersección de modelos	18
3.4.3. Construcción “desde abajo”: semántica del punto fijo	19
3.5. Conclusiones	21
4. Referencias	22
4.1. Bibliografía	22
4.2. Otras referencias	22

1. Lógica de proposiciones

La lógica de proposiciones es un lenguaje que permite expresar razonamientos en base a dos *valores de verdad*: “verdadero” y “falso”. Sintácticamente consta de **átomos**, unidades indivisibles a los que se asocia un valor de verdad, y **fórmulas**, que permiten expresar razonamientos compuestos a partir de átomos, y cuyo significado es también un valor de verdad que vendrá dado en función del de los átomos componentes.

1.1. Átomos

Un **átomo** es, como dice su nombre, una entidad indivisible e independiente en el lenguaje de lógica de predicados – se denotará como Σ al conjunto de átomos a considerar. Estos átomos pueden representarse mediante un conjunto finito de letras ($\Sigma = \{\mathbf{a}, \mathbf{b}, \dots, \mathbf{z}\}$) o de forma más general, por un conjunto infinito de identificadores alfanuméricos.

1.2. Fórmulas

A partir de un conjunto Σ de átomos se forma un lenguaje \mathcal{L} – o \mathcal{L}_Σ , para explicitar su dependencia de Σ – de fórmulas definido inductivamente como el menor conjunto cerrado bajo las siguientes reglas:

1. $\perp \in \mathcal{L}$ y $\top \in \mathcal{L}$.
2. Para cualquier $a \in \Sigma$, $a \in \mathcal{L}$.
3. Para cualquier $\alpha, \beta \in \mathcal{L}$ se cumple:
 - $\neg\alpha \in \mathcal{L}$.
 - $\alpha \wedge \beta \in \mathcal{L}$.

- $\alpha \vee \beta \in \mathcal{L}$.
- $\alpha \rightarrow \beta \in \mathcal{L}$.
- $\alpha \leftrightarrow \beta \in \mathcal{L}$.

El alfabeto del lenguaje será por tanto $A = \{\perp, \top, \neg, \wedge, \vee, \rightarrow, \leftrightarrow\} \cup \Sigma$.

Las fórmulas definidas por la regla (3) se llaman *fórmulas compuestas* (y las definidas por la regla (2) son *fórmulas atómicas*).

Un subconjunto $\Gamma \subseteq \mathcal{L}$ de fórmulas se llama **teoría**.

1.3. Semántica

1.3.1. Semántica de los átomos

La semántica de los átomos viene dada por una **interpretación**, que es cualquier función que asocie un significado a cada átomo de un conjunto Σ :

$$\mathfrak{I} : \Sigma \rightarrow \{\mathbf{T}, \mathbf{F}\}$$

Un ejemplo para $\Sigma = \{\mathbf{p}, \mathbf{q}, \mathbf{r}\}$ podría ser la función tal que $\mathfrak{I}(\mathbf{p}) = \mathbf{T}$, $\mathfrak{I}(\mathbf{q}) = \mathbf{F}$, $\mathfrak{I}(\mathbf{r}) = \mathbf{T}$.

Se denotará $S = \{\mathbf{T}, \mathbf{F}\}^\Sigma$ como el conjunto de todas las interpretaciones posibles para los átomos de Σ . En realidad este conjunto también puede ser visto como 2^Σ , es decir, el “conjunto potencia” de Σ y por tanto identificar una interpretación como un subconjunto de Σ ; los átomos que estén en el subconjunto tendrán semántica \mathbf{T} , y los que no \mathbf{F} , o viceversa. Ambas notaciones son equivalentes.

Si $|\Sigma| = n$, hay 2^n interpretaciones posibles.

1.3.2. Semántica de las fórmulas

Siendo \mathcal{L} cualquier lenguaje lógico de proposiciones definido sobre Σ , puede asociarse un significado a una fórmula $\alpha \in \mathcal{L}$ a partir de una interpretación $\mathfrak{I} : \Sigma \rightarrow \{\mathbf{T}, \mathbf{F}\}$. Para ello se define la relación *satisface*, denotada como $\mathfrak{I} \models \alpha$, que se lee “ \mathfrak{I} *satisface* la fórmula α ”, por la cual la interpretación \mathfrak{I} asocia la semántica de “verdadero” (\mathbf{T}) a la fórmula α . $\mathfrak{I} \not\models \alpha$ se lee “ \mathfrak{I} *no satisface* la fórmula α ”, y equivale a asociar la semántica de “falso” (\mathbf{F}).

La relación se define por inducción estructural de la siguiente manera, para cualquier interpretación $\mathfrak{I} : \Sigma \rightarrow \{\mathbf{T}, \mathbf{F}\}$, para cualquier $p \in \Sigma$, $\alpha, \beta \in \mathcal{L}$:

- $\mathfrak{I} \not\models \perp$.
- $\mathfrak{I} \models \top$.
- $\mathfrak{I} \models p$ si y sólo si $\mathfrak{I}(p) = \mathbf{T}$.
- $\mathfrak{I} \models \neg\alpha$ si y sólo si $\mathfrak{I} \not\models \alpha$.
- $\mathfrak{I} \models \alpha \wedge \beta$ si y sólo si $\mathfrak{I} \models \alpha$ y $\mathfrak{I} \models \beta$.
- $\mathfrak{I} \models \alpha \vee \beta$ si y sólo si o bien $\mathfrak{I} \models \alpha$, o bien $\mathfrak{I} \models \beta$, o ambos.
- $\mathfrak{I} \models \alpha \rightarrow \beta$ si y sólo si o bien $\mathfrak{I} \not\models \alpha$, o bien $\mathfrak{I} \models \beta$, o ambos.

- $\mathfrak{I} \models \alpha \leftrightarrow \beta$ si y sólo si o bien $\mathfrak{I} \models \alpha$ y $\mathfrak{I} \models \beta$, o bien $\mathfrak{I} \not\models \alpha$ y $\mathfrak{I} \not\models \beta$.

La siguiente tabla muestra el valor de verdad asignado a las fórmulas compuestas a partir de α y β en función de los valores asignados a α y β :

α	β	$\neg\alpha$	$\alpha \wedge \beta$	$\alpha \vee \beta$	$\alpha \rightarrow \beta$	$\alpha \leftrightarrow \beta$
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

A una tabla de estas características se le denomina *tabla de verdad*.

1.3.3. Propiedades

En lo siguiente se asumirá cualquier lenguaje \mathcal{L} sobre Σ , donde $p \in \Sigma$ es cualquier átomo, $\mathfrak{I} : \Sigma \rightarrow \{T, F\}$ es cualquier interpretación, $\alpha, \beta \in \mathcal{L}$ cualesquier fórmulas, y $\Gamma \subseteq \mathcal{L}$ cualquier teoría o conjunto de fórmulas..

- Se dice que \mathfrak{I} es un **modelo** de la fórmula α sii¹ $\mathfrak{I} \models \alpha$. Se dice que \mathfrak{I} es un **contraejemplo** de α sii $\mathfrak{I} \not\models \alpha$.

También se dice que \mathfrak{I} es un modelo de Γ , escrito $\mathfrak{I} \models \Gamma$, si es modelo de todas sus fórmulas (es decir, si $\mathfrak{I} \models \gamma$ para todo $\gamma \in \Gamma$).

- Sii una fórmula α se satisface para cualquier interpretación \mathfrak{I} , se dice que es una **tautología**. Son ejemplos de tautologías \top y $p \vee \neg p$. Análogamente, sii una fórmula nunca se satisface en cualquier interpretación \mathfrak{I} , se dice que es una **contradicción**. Son ejemplos de contradicciones \perp y $p \wedge \neg p$.

Es claro que si α es una contradicción, entonces $\neg\alpha$ es una tautología, y viceversa.

- Sii cualquier interpretación asigna el mismo valor de verdad a dos fórmulas α y β , se dice que α y β son **lógicamente equivalentes** y se denota $\alpha \equiv \beta$. Es claro que $\alpha \equiv \beta$ sii $\alpha \leftrightarrow \beta$ es una tautología.

- Dada una teoría o conjunto de fórmulas $\Gamma \subseteq \mathcal{L}$, se dice que Γ es **consistente** sii existe al menos una interpretación que satisfaga todas las fórmulas. Si Γ es un conjunto finito $\{\gamma_1, \gamma_2, \dots, \gamma_n\}$, se tiene que Γ es consistente sii la fórmula $\gamma_1 \wedge \gamma_2 \wedge \dots \wedge \gamma_n$ admite algún modelo. Sii Γ no es consistente se dice que es **inconsistente**. Es claro que conjunto vacío $\{\}$ es consistente.

- Sii cualquier interpretación que satisfaga todas las fórmulas de $\Gamma \subseteq \mathcal{L}$ también satisface $\alpha \in \mathcal{L}$, se dice que α es **consecuencia lógica** de Γ y se denota $\Gamma \models \alpha$. En el caso particular de que $\Gamma = \{\}$, se tiene que cualquier interpretación satisface todas las fórmulas de $\{\}$, y por tanto “ $\{\} \models \alpha$ ”, o simplemente “ $\models \alpha$ ”, es equivalente a decir “ α es una tautología”.

Siendo Γ un conjunto finito $\{\gamma_1, \gamma_2, \dots, \gamma_n\}$, se tiene que α es consecuencia lógica de Γ sii $\gamma_1 \wedge \gamma_2 \wedge \dots \wedge \gamma_n \rightarrow \alpha$ es una tautología.

En general, si $\Gamma = \{\alpha\}$, se escribe “ $\alpha \models \beta$ ” como equivalente a “ $\{\alpha\} \models \beta$ ”.

¹En adelante se utilizará “sii” como abreviatura de “si y sólo si”.

1.3.4. Fortaleza y debilidad de fórmulas

Se define la función $M : \mathcal{L} \rightarrow 2^S$ que asocia a cualquier fórmula de \mathcal{L} el conjunto de interpretaciones que la satisfacen; es decir, sus modelos. En general, para cualquier $\alpha \in \mathcal{L}$, $\mathfrak{S} \in S$, se tiene que $\mathfrak{S} \in M(\alpha)$ si y sólo si $\mathfrak{S} \models \alpha$.

Debido a las definiciones que se han dado en el apartado anterior, se tiene lo siguiente, para cualquier $\alpha, \beta \in \mathcal{L}$:

- Sii α es una tautología, es decir, sii $\models \alpha$, entonces $M(\alpha) = S$ (todas las interpretaciones son modelo de α).
- Sii α es una contradicción, entonces $M(\alpha) = \emptyset$.
- Sii $\alpha \equiv \beta$ entonces $M(\alpha) = M(\beta)$.
- Sii $\alpha \models \beta$ entonces $M(\alpha) \subseteq M(\beta)$.

Este último punto sugiere las nociones de *fortaleza* y *debilidad* de fórmulas. Dadas cualquier $\alpha, \beta \in \mathcal{L}$, se dice que “ α es *más fuerte* que β ” (o, equivalentemente, “ β es *más débil* que α), si y sólo si $M(\alpha) \subseteq M(\beta)$ – es decir, $\alpha \models \beta$.

Es claro que las tautologías son *más débiles* que cualquier otra fórmula, es decir, $\alpha \models \top$ para cualquier α . Análogamente, las contradicciones son *más fuertes* que cualquier otra fórmula, es decir, $\perp \models \alpha$ para cualquier α .

1.4. Consecuencia y equivalencia lógica

Se debe recalcar la diferencia que existe entre “ $\alpha \rightarrow \beta$ ” y “ $\alpha \models \beta$ ”, para $\alpha, \beta \in \mathcal{L}$.

La primera es una nueva fórmula lógica de \mathcal{L} que formaliza una *consecuencia* dentro de un dominio (interpretación) particular. Es decir, será cierta en una determinada interpretación sii β es cierta si α es cierta en la interpretación, según se definió en la relación “satisface”.

La segunda de las expresiones no es una fórmula lógica; ni siquiera el símbolo “ \models ” forma parte del alfabeto del lenguaje lógico que se definió en la sección correspondiente; y expresa, según se ha dicho, una *consecuencia lógica*: una consecuencia debido a las leyes del lenguaje lógico que es siempre cierta, en cualquier dominio (interpretación), y que equivale por tanto a “ $\models \alpha \rightarrow \beta$ ”.

Análogamente, la misma diferencia existe entre “ $\alpha \leftrightarrow \beta$ ” y “ $\alpha \equiv \beta$ ”.

Propiedad : Sea $\gamma(\alpha) \in \mathcal{L}$ una fórmula, donde α puede o no aparecer como subfórmula. Si $\alpha \equiv \beta$, entonces $\gamma(\alpha) \equiv \gamma(\beta)$.

1.4.1. Algunas equivalencias lógicas

Algunas equivalencias lógicas, para cualquier $\alpha, \beta, \gamma \in \mathcal{L}$:

1. $\neg\neg\alpha \equiv \alpha$ (doble negación).
2. $\alpha \wedge \alpha \equiv \alpha$ (propiedad idempotente).
3. $\alpha \wedge \beta \equiv \beta \wedge \alpha$ (propiedad conmutativa).
4. $(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ (propiedad asociativa).

5. $\alpha \vee \alpha \equiv \alpha$ (propiedad idempotente).
6. $\alpha \vee \beta \equiv \beta \vee \alpha$ (propiedad conmutativa).
7. $(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$ (propiedad asociativa).
8. $(\alpha \vee \beta) \wedge \gamma \equiv (\alpha \wedge \gamma) \vee (\beta \wedge \gamma)$ (propiedad distributiva).
9. $(\alpha \wedge \beta) \vee \gamma \equiv (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$ (propiedad distributiva).
10. $\alpha \vee \perp \equiv \alpha$ (elemento neutro).
11. $\alpha \wedge \top \equiv \alpha$ (elemento neutro).
12. $\top \rightarrow \alpha \equiv \alpha$.
13. $\alpha \rightarrow \perp \equiv \neg\alpha$.
14. $\alpha \rightarrow \beta \equiv (\neg\alpha) \vee \beta$.
15. $\neg(\alpha \vee \beta) \equiv (\neg\alpha) \wedge (\neg\beta)$ (de Morgan).
16. $\neg(\alpha \wedge \beta) \equiv (\neg\alpha) \vee (\neg\beta)$ (de Morgan).
17. $\alpha \wedge \perp \equiv \perp$ (elemento absorbente).
18. $\alpha \wedge \neg\alpha \equiv \perp$.
19. $\alpha \vee \neg\alpha \equiv \top$.
20. $\alpha \rightarrow \top \equiv \top$.
21. $\perp \rightarrow \alpha \equiv \top$.
22. $\alpha \vee \top \equiv \top$ (elemento absorbente).
23. $\alpha \wedge \beta \rightarrow \alpha \equiv \top$.

Estas equivalencias lógicas pueden utilizarse para probar ciertas tautologías. Por ejemplo, se probará que “ $(\alpha \rightarrow \beta) \wedge \neg\beta \rightarrow \neg\alpha$ ” (para cualquier $\alpha, \beta \in \mathcal{L}$) es una tautología:

$$\begin{aligned}
& (\alpha \rightarrow \beta) \wedge \neg\beta \rightarrow \neg\alpha \\
\equiv & (\neg\alpha \vee \beta) \wedge \neg\beta \rightarrow \neg\alpha & (14) \\
\equiv & (\neg\alpha \wedge \neg\beta) \vee (\beta \wedge \neg\beta) \rightarrow \neg\alpha & (8) \\
\equiv & (\neg\alpha \wedge \neg\beta) \vee \perp \rightarrow \neg\alpha & (18) \\
\equiv & \neg\alpha \wedge \neg\beta \rightarrow \neg\alpha & (10) \\
\equiv & \top & (23)
\end{aligned}$$

Por tanto se tiene $(\alpha \rightarrow \beta) \wedge \neg\beta \rightarrow \neg\alpha \equiv \top$, es decir, $\models (\alpha \rightarrow \beta) \wedge \neg\beta \rightarrow \neg\alpha$ o, equivalentemente, $\{\alpha \rightarrow \beta, \neg\beta\} \models \neg\alpha$. Esta consecuencia lógica se conoce en lógica clásica como *Modus Tollens*.

1.4.2. Dedución: sistemas deductivos

Sea \mathcal{L} cualquier lenguaje proposicional de fórmulas. Sea $\Gamma \subseteq \mathcal{L}$ una teoría. Dicha teoría estará formada por fórmulas que formalizan ciertos hechos y relaciones lógicas entre ellos de un cierto dominio. El objetivo será obtener una nueva fórmula $\alpha \in \mathcal{L}$ que sea *consecuencia lógica* de la fórmulas de Γ , es decir, tal que $\Gamma \models \alpha$. Naturalmente, para que esto tenga sentido, el conjunto Γ ha de ser consistente. Si Γ fuese inconsistente, se tendría $\Gamma \models \alpha$ para cualquier α .

Un **sistema deductivo** D es un sistema formal que permitirá, mediante una serie de reglas, *derivar* fórmulas α a partir de teorías Γ . Se escribe “ $\Gamma \vdash_D \alpha$ ” para expresar “ α se deriva de Γ en D ”.

Pueden definirse las siguientes propiedades para un sistema deductivo:

- **Corrección (*Soundness*)**: Un sistema de reglas D es *correcto* si, para cualquier teoría Γ y fórmula α :

$$\text{Si } \Gamma \vdash_D \alpha, \text{ entonces } \Gamma \models \alpha$$

- **Completitud (*Completeness*)**: Un sistema D será *completo* si, para cualquier teoría Γ y fórmula α :

$$\text{Si } \Gamma \models \alpha, \text{ entonces } \Gamma \vdash_D \alpha$$

Como se observa, la corrección es una propiedad fundamental de cualquier sistema deductivo, puesto que es la que permitirá afirmar que las deducciones o derivaciones en D son consecuencias lógicas válidas.

2. Lógica de predicados

La **lógica de predicados** es una extensión de la lógica de proposiciones. Si la lógica de proposiciones permitía expresar razonamientos en base a átomos sobre el significado de “verdadero” o “falso”, la lógica de predicados permite razonar sobre *relaciones* o *propiedades*² que presentan objetos de un **universo**. Para ello, proporciona herramientas para representar objetos concretos del universo, e introduce el concepto de *variable* que permite referenciar un objeto cualquiera del universo (sin concretar), un mecanismo de abstracción.

La lógica de predicados es de *primer orden* (también conocida como FOL, de *first order logic*) si dichas variables sólo pueden hacer referencia a un objeto cualquiera del universo y no a una “relación cualquiera” entre ciertos objetos.

Sintácticamente, el lenguaje consta de *términos*, que permiten representar los objetos del universo, y *fórmulas*, que permiten expresar el hecho de que n objetos estén relacionados mediante una relación o propiedad determinada, y razonamientos lógicos en base a estos hechos (al igual que en la lógica de proposiciones). El significado de un término será por tanto un objeto del universo, y el de una fórmula un valor de verdad.

Se definirá a continuación el *alfabeto* sobre el que se construye el lenguaje:

²En este documento se utilizará “relación” y “propiedad” sin hacer distinciones, de forma intercambiable.

2.1. Alfabeto

Un alfabeto para el lenguaje de la lógica de predicados está formado por los siguientes elementos:

- Las **constantes**: permitirán hacer referencia a un objeto concreto del universo. Se representarán por identificadores alfanuméricos con tipografía “versales”; por ejemplo ROJO, NORTE, ORQUIDEA. También es frecuente su representación por identificadores que comienzan con letra minúscula. Será \mathcal{C} el conjunto de las constantes.
- Las **variables**: permitirán hacer referencia a un objeto del universo sin especificar. Como convención léxica, las variables se representarán mediante identificadores alfanuméricos que comienzan por una letra mayúscula y con tipografía “cursiva”: por ejemplo X , Z , *Resultado* o *Valor*. Será \mathcal{V} el conjunto de las variables.
- Los **funtores**: permitirán hacer referencia a *objetos compuestos*, como por ejemplo una estructura de datos. Un objeto compuesto podrá componerse de n objetos, a este número n se llamará la *aridad* del funtor. Se representarán por un identificador alfanumérico con tipografía “versales” seguido por su aridad, de forma que f/n hará referencia al funtor con identificador f y aridad n (realmente una constante no es más que un funtor de aridad 0).

Por ejemplo, un funtor FECHA/3 podría representar una estructura “fecha” compuesta de tres objetos (un día, un mes y un año).

Será \mathcal{F} el conjunto de los funtores.

- Los **predicados**: expresan una propiedad determinada que cumplen determinadas tuplas de n objetos; a este número n se llamará la *aridad* del predicado. Se representarán por un identificador alfanumérico con tipografía “versales” seguido por su aridad, de forma que p/n hará referencia al predicado con identificador p y aridad n .

Por ejemplo, en un universo de objetos en el espacio, un predicado DENTRO_DE/2 podría expresar la relación “dentro de”. En un universo numérico (por ejemplo el de los números naturales), la relación SUMAN/3 podría expresar la relación de que la suma de dos objetos es un tercero.

Será \mathcal{P} el conjunto de los predicados.

2.2. Término

El **término** será la categoría sintáctica que permitirá referenciar objetos del universo. Utilizará por tanto átomos (para representar objetos simples del universo), variables (para hacer referencia a un objeto sin especificar) y funtores (para representar objetos compuestos de otros objetos).

El conjunto de términos \mathcal{T} se define, a partir de un alfabeto dado, de forma inductiva como el menor conjunto cerrado bajo las siguientes reglas:

1. Cualquier constante $a \in \mathcal{C}$ es un término, $a \in \mathcal{T}$.
2. Cualquier variable $v \in \mathcal{V}$ es un término, $v \in \mathcal{T}$.
3. Si $f/n \in \mathcal{F}$ es un funtor, y $t_1, \dots, t_n \in \mathcal{T}$, entonces $f(t_1, \dots, t_n) \in \mathcal{T}$.

2.3. Fórmula

El conjunto de fórmulas \mathcal{L} sobre un alfabeto dado se define inductivamente como el menor conjunto cerrado bajo las siguientes reglas:

1. $\top \in \mathcal{L}$ y $\perp \in \mathcal{L}$
2. Si $p/n \in \mathcal{P}$, y $t_1, \dots, t_n \in \mathcal{T}$, entonces $p(t_1, \dots, t_n) \in \mathcal{L}$.
3. Si $\alpha, \beta \in \mathcal{L}$ son fórmulas también lo son:
 - $\neg\alpha$
 - $\alpha \wedge \beta$
 - $\alpha \vee \beta$
 - $\alpha \rightarrow \beta$
 - $\alpha \leftrightarrow \beta$
4. Si $x \in \mathcal{V}$ es una variable, y $\alpha \in \mathcal{L}$ es una fórmula, también son fórmulas:
 - $\forall x \alpha$
 - $\exists x \alpha$

Los símbolos $\{\top, \perp, (,) \neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ también son parte del alfabeto del lenguaje, aunque se identificará “alfabeto” simplemente con constantes, variables, funtores y predicados.

Las fórmulas definidas por la regla (2) se llamarán **fórmulas atómicas** o, simplemente, **átomos**.

2.4. Semántica

En adelante, será \mathcal{D} el conjunto de los objetos del universo considerado (también llamado *dominio*).

Al igual que en la lógica de proposiciones, será necesario asignar un significado a los elementos del alfabeto, lo que se denomina *interpretar* el alfabeto.

2.4.1. Interpretación

Una **interpretación** \mathfrak{S} de un alfabeto asocia:

- Cada constante $a \in \mathcal{C}$ con un significado $a_{\mathfrak{S}} \in \mathcal{D}$ en el dominio.
- Cada funtor $f/n \in \mathcal{F}$ con una función $f_{\mathfrak{S}} : \mathcal{D}^n \rightarrow \mathcal{D}$.
- Cada predicado $p/n \in \mathcal{P}$ con una relación $p_{\mathfrak{S}} \subseteq \underbrace{\mathcal{D} \times \dots \times \mathcal{D}}_n$.

Ha de observarse que las variables no forman parte de la interpretación del alfabeto. Esto es debido a que las variables hacen referencia a un objeto sin especificar del universo, y por tanto no tienen un significado estático definido a priori, sino uno dinámico que puede variar según el contexto.

2.4.2. Valuación

La semántica de las variables vendrá dada por una **valuación**, que como se ha dicho podrá variar según el contexto. Una valuación es una función $\phi : \mathcal{V} \rightarrow \mathcal{D}$. Si $m \in \mathcal{D}$ es cualquier objeto y $x \in \mathcal{V}$ cualquier variable, entonces $\phi\{m/x\} : \mathcal{V} \rightarrow \mathcal{D}$ es una nueva valuación, obtenida a partir de ϕ de la siguiente manera:

$$\phi\{m/x\}(y) = \begin{cases} m & y = x \\ \phi(y) & y \neq x \end{cases}$$

para todo $y \in \mathcal{V}$.

Se utilizará también la notación $\phi\{m_1/x_1, \dots, m_n/x_n\}$ para especificar un cambio múltiple de contexto.

2.4.3. Semántica de los términos

Como se ha dicho, los términos representan los objetos del universo. Por tanto, la semántica de los términos será una correspondencia $\mathcal{T} \rightarrow \mathcal{D}$ que asocia un significado del dominio \mathcal{D} a cada término.

La *semántica de los términos* dependerá de la interpretación \mathfrak{S} del alfabeto y de una valuación ϕ de las variables en un contexto determinado. Se denotará como $\mathfrak{S}_\phi : \mathcal{T} \rightarrow \mathcal{D}$.

Si $t \in \mathcal{T}$ es un término, el significado $\mathfrak{S}_\phi(t) \in \mathcal{D}$ vendrá dado de la siguiente manera:

- Si $t \in \mathcal{C}$, $\mathfrak{S}_\phi(t) = t_{\mathfrak{S}}$.
- Si $t \in \mathcal{V}$, se asigna $\mathfrak{S}_\phi(t) = \phi(t)$.
- Si $t = f(t_1, \dots, t_n)$ donde $f/n \in \mathcal{F}$ y $t_1, \dots, t_n \in \mathcal{T}$, se asigna $\mathfrak{S}_\phi(t) = f_{\mathfrak{S}}(\mathfrak{S}_\phi(t_1), \dots, \mathfrak{S}_\phi(t_n))$.

2.4.4. Semántica de las fórmulas

A partir de una interpretación \mathfrak{S} del alfabeto y – lo que es una novedad con respecto a la lógica de proposiciones – de un contexto (valuación) inicial de variables ϕ puede asociarse un significado a una fórmula $\alpha \in \mathcal{L}$. Para ello se define la relación *satisface*, denotada como $\mathfrak{S} \models_\phi \alpha$, que se lee “ \mathfrak{S} *satisface* la fórmula α en el contexto ϕ ”, por la cual la interpretación \mathfrak{S} en el contexto ϕ asocia la semántica de “verdadero” (T) a la fórmula α . $\mathfrak{S} \not\models_\phi \alpha$ se lee “ \mathfrak{S} *no satisface* la fórmula α en el contexto ϕ ”, y equivale a asociar la semántica de “falso” (F).

La relación se define inductivamente de la siguiente manera, para cualquier interpretación \mathfrak{S} y contexto inicial ϕ , para cualquier $x \in \mathcal{V}$, $\alpha, \beta \in \mathcal{L}$:

- $\mathfrak{S} \models_\phi \top$.
- $\mathfrak{S} \not\models_\phi \perp$.
- $\mathfrak{S} \models_\phi p(t_1, \dots, t_n)$ donde $p/n \in \mathcal{P}$ si y sólo si $\langle \mathfrak{S}_\phi(t_1), \dots, \mathfrak{S}_\phi(t_n) \rangle \in p_{\mathfrak{S}}$.
- $\mathfrak{S} \models_\phi \neg\alpha$ si y sólo si $\mathfrak{S} \not\models_\phi \alpha$.

- $\mathfrak{S} \models_{\phi} \alpha \wedge \beta$ si y sólo si $\mathfrak{S} \models_{\phi} \alpha$ y $\mathfrak{S} \models_{\phi} \beta$.
- $\mathfrak{S} \models_{\phi} \alpha \vee \beta$ si y sólo si o bien $\mathfrak{S} \models_{\phi} \alpha$, o bien $\mathfrak{S} \models_{\phi} \beta$, o ambos.
- $\mathfrak{S} \models_{\phi} \alpha \rightarrow \beta$ si y sólo si o bien $\mathfrak{S} \not\models_{\phi} \alpha$, o bien $\mathfrak{S} \models_{\phi} \beta$, o ambos.
- $\mathfrak{S} \models_{\phi} \alpha \leftrightarrow \beta$ si y sólo si o bien $\mathfrak{S} \models_{\phi} \alpha$ y $\mathfrak{S} \models_{\phi} \beta$, o bien $\mathfrak{S} \not\models_{\phi} \alpha$ y $\mathfrak{S} \not\models_{\phi} \beta$.
- $\mathfrak{S} \models_{\phi} \forall x \alpha$ sii, para todo $m \in \mathcal{D}$, $\mathfrak{S} \models_{\phi\{m/x\}} \alpha$.
- $\mathfrak{S} \models_{\phi} \exists x \alpha$ sii, para algún $m \in \mathcal{D}$, $\mathfrak{S} \models_{\phi\{m/x\}} \alpha$.

2.4.5. Variables libres y ligadas

La semántica de las fórmulas $\forall x \alpha$ y $\exists x \alpha$ (para todo $x \in \mathcal{V}$, $\alpha \in \mathcal{L}$), se ha definido de forma que $\mathfrak{S} \models_{\phi} \forall x \alpha$ sii, para todo $m \in \mathcal{D}$, $\mathfrak{S} \models_{\phi\{m/x\}} \alpha$ (análogamente para “ \exists ”). Se observa que $\phi\{m/x\}(x) = m$ para todo ϕ por definición, es decir, es irrelevante la valuación que tenga en ϕ la variable x para la semántica de $\forall x \alpha$ (y de $\exists x \alpha$). Por ello, se dice que la variable x está **ligada** en dichas fórmulas.

Las variables que apareciesen en una fórmula que no están ligadas se dice que están **libres**.

Se define la función $FV : \mathcal{L} \cup \mathcal{T} \rightarrow 2^{\mathcal{V}}$, que obtiene el conjunto de variables libres de una fórmula o término, de la siguiente manera, para todo $c \in \mathcal{C}$, $x \in \mathcal{V}$, $f/n \in \mathcal{F}$, $p/n \in \mathcal{P}$, $\alpha \in \mathcal{L}$.

- $FV(c) = \emptyset$.
- $FV(x) = \{x\}$.
- $FV(f(t_1, \dots, t_n)) = FV(t_1) \cup \dots \cup FV(t_n)$, donde $t_1, \dots, t_n \in \mathcal{T}$.
- $FV(\top) = FV(\perp) = \emptyset$.
- $FV(p(t_1, \dots, t_n)) = FV(t_1) \cup \dots \cup FV(t_n)$, donde $t_1, \dots, t_n \in \mathcal{T}$.
- $FV(\neg\alpha) = FV(\alpha)$.
- $FV(\alpha \wedge \beta) = FV(\alpha) \cup FV(\beta)$.
- $FV(\alpha \vee \beta) = FV(\alpha) \cup FV(\beta)$.
- $FV(\alpha \rightarrow \beta) = FV(\alpha) \cup FV(\beta)$.
- $FV(\alpha \leftrightarrow \beta) = FV(\alpha) \cup FV(\beta)$.
- $FV(\forall x \alpha) = FV(\alpha) - \{x\}$.
- $FV(\exists x \alpha) = FV(\alpha) - \{x\}$.

Si $\alpha \in \mathcal{L}$ es cualquier fórmula, se dice que α es *cerrada* o *base* sii $FV(\alpha) = \emptyset$.

Si $t \in \mathcal{T}$ es cualquier término, se dice que t es un *término base* (*ground term*) sii $FV(t) = \emptyset$. Decir *término base* equivale a decir *término sin variables*, puesto que no es posible que haya variables ligadas en un término.

Es claro que si α es una fórmula cerrada, su semántica no depende de una valuación inicial. Por ello, se escribirá simplemente “ $\mathfrak{S} \models \alpha$ ” para denotar que una interpretación \mathfrak{S} satisface la fórmula cerrada α ; en general, $\mathfrak{S} \models \alpha$ si y sólo si $\mathfrak{S} \models_{\phi} \alpha$ para toda valuación inicial ϕ .

2.4.6. Ejemplo

A modo de ejemplo, tómesese el universo de los números naturales ($\mathcal{D} = \mathbb{N}$). Sea el siguiente alfabeto:

$$\begin{aligned}\mathcal{C} &= \{\text{ZERO}\} \\ \mathcal{F} &= \{\text{SUCC}/1\} \\ \mathcal{P} &= \{\text{EVEN}/1\}\end{aligned}$$

Y sea la interpretación \mathfrak{S} tal que

$$\begin{aligned}\text{ZERO}_{\mathfrak{S}} &= 0 \\ \text{SUCC}_{\mathfrak{S}}(n) &= 1 + n \text{ para todo } n \in \mathbb{N} \\ \text{EVEN}_{\mathfrak{S}} &= \{\langle 0 \rangle, \langle 2 \rangle, \langle 4 \rangle, \langle 6 \rangle, \langle 8 \rangle, \dots\}\end{aligned}$$

Siendo $\alpha = \text{EVEN}(\text{SUCC}(\text{SUCC}(\text{SUCC}(\text{ZERO}))))$, se calculará la semántica de α según la interpretación \mathfrak{S} y en cualquier valuación inicial ϕ (ya que $\text{FV}(\alpha) = \emptyset$):

$$\mathfrak{S} \models \alpha \text{ si y sólo si } \langle \mathfrak{S}_{\phi}(\text{SUCC}(\text{SUCC}(\text{SUCC}(\text{ZERO})))) \rangle \in \text{EVEN}_{\mathfrak{S}}.$$

$$\begin{aligned}\mathfrak{S}_{\phi}(\text{SUCC}(\text{SUCC}(\text{SUCC}(\text{ZERO})))) &= \\ &= 1 + \mathfrak{S}_{\phi}(\text{SUCC}(\text{SUCC}(\text{ZERO}))) = \\ &= 1 + 1 + \mathfrak{S}_{\phi}(\text{SUCC}(\text{ZERO})) = \\ &= 1 + 1 + 1 + \mathfrak{S}_{\phi}(\text{ZERO}) = \\ &= 1 + 1 + 1 + 0 = \\ &= 3\end{aligned}$$

Como $\langle 3 \rangle \notin \text{EVEN}_{\mathfrak{S}}$, se tiene

$$\mathfrak{S} \not\models \alpha$$

Es decir, la semántica de la fórmula α en la interpretación \mathfrak{S} (en cualquier valuación) es F (“falso”).

2.4.7. Propiedades

En lo siguiente se asumirá cualquier lenguaje de predicados \mathcal{L} , cualquier interpretación \mathfrak{S} , siendo $\alpha, \beta \in \mathcal{L}$ cualesquier fórmulas cerradas – es decir, $\text{FV}(\alpha) = \text{FV}(\beta) = \emptyset$ –, y $\Gamma \subseteq \mathcal{L}$ cualquier teoría o conjunto de fórmulas.

Las siguientes propiedades y definiciones son análogas a las que se hizo para la lógica proposicional, pero teniendo en cuenta que, como se ha dicho, α y β son fórmulas cerradas. También hay alguna diferencia de nomenclatura.

- Se dice que \mathfrak{S} es un **modelo** de la fórmula cerrada α sii $\mathfrak{S} \models \alpha$. Se dice que \mathfrak{S} es un **contraejemplo** de α sii $\mathfrak{S} \not\models \alpha$.

También se dice que \mathfrak{S} es un modelo de Γ , escrito $\mathfrak{S} \models \Gamma$, si es modelo de todas sus fórmulas (es decir, si $\mathfrak{S} \models \gamma$ para todo $\gamma \in \Gamma$).

- Sii una fórmula cerrada α se satisface para cualquier interpretación \mathfrak{S} , se dice que es una **validez lógica**. Análogamente, sii una fórmula nunca se satisface en cualquier interpretación \mathfrak{S} , se dice que es una **contradicción**.
Es claro que si α es una contradicción, entonces $\neg\alpha$ es una validez lógica, y viceversa.
- Sii cualquier interpretación asigna el mismo valor de verdad a dos fórmulas α y β , se dice que α y β son **lógicamente equivalentes** y se denota $\alpha \equiv \beta$.
Es claro que $\alpha \equiv \beta$ sii $\alpha \leftrightarrow \beta$ es una validez lógica.
- Dada una teoría o conjunto de fórmulas $\Gamma \subseteq \mathcal{L}$, se dice que Γ es **consistente** sii existe al menos una interpretación que satisfaga todas las fórmulas. Si Γ es un conjunto finito $\{\gamma_1, \gamma_2, \dots, \gamma_n\}$, se tiene que Γ es consistente sii la fórmula $\gamma_1 \wedge \gamma_2 \wedge \dots \wedge \gamma_n$ admite algún modelo. Sii Γ no es consistente se dice que es **inconsistente**. Es claro que conjunto vacío $\{\}$ es consistente.
- Sii cualquier interpretación que satisfaga todas las fórmulas de $\Gamma \subseteq \mathcal{L}$ también satisface $\alpha \in \mathcal{L}$, se dice que α es **consecuencia lógica** de Γ y se denota $\Gamma \models \alpha$. En el caso particular de que $\Gamma = \{\}$, se tiene que cualquier interpretación satisface todas las fórmulas de $\{\}$, y por tanto “ $\{\} \models \alpha$ ”, o simplemente “ $\models \alpha$ ”, es equivalente a decir “ α es una validez lógica”.
Siendo Γ un conjunto finito $\{\gamma_1, \gamma_2, \dots, \gamma_n\}$, se tiene que α es consecuencia lógica de Γ sii $\gamma_1 \wedge \gamma_2 \wedge \dots \wedge \gamma_n \rightarrow \alpha$ es una validez lógica.
En general, si $\Gamma = \{\alpha\}$, se escribe “ $\alpha \models \beta$ ” como equivalente a “ $\{\alpha\} \models \beta$ ”.

2.4.8. Algunas equivalencias lógicas

Todas las equivalencias lógicas en lógica de proposiciones que aparecen en la sección 1.4.1 son también equivalencias lógicas en lógica de predicados.

Como novedad se introducen las siguientes, conocidas como *leyes “de Morgan” generalizadas*, para todo $\alpha \in \mathcal{L}$, $x \in \mathcal{V}$:

- $\neg(\forall x \alpha) \equiv \exists x \neg\alpha$
- $\neg(\exists x \alpha) \equiv \forall x \neg\alpha$

3. Programación lógica

El objetivo de la programación lógica será describir mediante fórmulas lógicas, en concreto lógica de predicados de primer orden, una serie de hechos y reglas sobre relaciones o propiedades de algún problema o computación que se quiera resolver. Un programa lógico definirá un *modelo* que satisfaga dichos hechos y reglas. Dicho modelo constituirá la *semántica declarativa* del programa lógico.

De acuerdo con esta semántica, el objetivo de un *intérprete lógico* será, dado una fórmula lógica en cierto formato, decir si el modelo que define el programa lógico satisface o no dicha fórmula (y también, si contiene variables, para qué valuación de las mismas se satisface).

3.1. Cláusulas de Horn

Las *cláusulas de Horn* son fórmulas lógicas que expresan conocimiento declarativo positivo. Tienen la siguiente forma, donde A_0, A_1, \dots, A_n son átomos – es decir, fórmulas de la forma $p(t_1, \dots, t_n)$ para cualquier $p/n \in \mathcal{P}$ y $t_1, \dots, t_n \in \mathcal{T}$ –:

$$A_0 \leftarrow A_1 \wedge \dots \wedge A_n$$

(Se ha invertido el sentido del símbolo “ \rightarrow ” ya que ésta es la forma más frecuente en programación lógica y, en concreto, en el lenguaje de programación PROLOG.)

Se ha dicho que “expresan conocimiento declarativo positivo” puesto que dichas cláusulas pueden expresar las condiciones para que exista una relación entre ciertos elementos, pero nunca cuándo esta relación *no* se da.

En adelante, tomando $\Delta = \{A_1, \dots, A_n\}$ y $A = A_0$, se representarán las cláusulas de la siguiente forma:

$$A \leftarrow \Delta$$

Si $\Delta = \emptyset$, la cláusula es lógicamente equivalente a “ A ”³

Un conjunto $\Gamma \subseteq \{A \leftarrow \Delta\}$ formado por cláusulas de Horn será un **programa lógico**.

Es claro que un programa lógico será siempre consistente, puesto que las cláusulas de Horn están formadas por fórmulas atómicas (positivas), siendo imposible expresar una contradicción.

3.2. Universo de un programa lógico

Se ha dicho que la semántica del lenguaje lógico de primer orden estaba asociada a un *universo* de significado. Así, el significado de un término era un elemento de dicho dominio, y la semántica de un predicado era una relación de elementos de dicho dominio.

¿Cuál es el universo asociado a un programa lógico? Desde el punto de vista del lenguaje de programación, estas relaciones se harán en base a los propios términos formados a partir del alfabeto (símbolos) del lenguaje, con independencia de cualquier dominio externo que el programador quiera representar con ellos.

3.2.1. Universo de Herbrand

El *universo de Herbrand*, \mathcal{H} , es el conjunto formado por todos los *términos sin variables* que pueden formarse en el alfabeto considerado. Se define como

$$\mathcal{H} = \bigcup_{i \in \mathbb{N}} H_i$$

a partir de la siguiente serie:

$$\begin{aligned} H_0 &= \mathcal{C} \\ H_{i+1} &= H_i \cup \{f(t_1, \dots, t_n) \mid f/n \in \mathcal{F}, t_1, \dots, t_n \in H_i\} \text{ para todo } i \in \mathbb{N} \end{aligned}$$

El universo de Herbrand será el universo asociado a un programa lógico (sobre un alfabeto considerado).

³Ello se sigue de que \top es elemento neutro, con respecto a la equivalencia de fórmulas, de la conjunción, y de que $\top \rightarrow \alpha \equiv \alpha$ para toda fórmula α .

3.2.2. Base de Herbrand

La *base de Herbrand*, \mathcal{B} , es el conjunto formado por todos los *átomos cerrados* que pueden formarse en el alfabeto considerado. Se define de la siguiente manera:

$$\mathcal{B} = \{p(t_1, \dots, t_n) \mid p/n \in \mathcal{P}, t_1, \dots, t_n \in \mathcal{H}\}$$

3.2.3. Interpretación de Herbrand

Una *interpretación de Herbrand* del alfabeto de un programa lógico es una interpretación \mathfrak{S} que satisface los siguientes requisitos – teniendo en cuenta que el universo asociado es el universo de Herbrand \mathcal{H} , según se ha dicho:

- Asigna a cada constante $c \in \mathcal{C}$ a si misma: $c_{\mathfrak{S}} = c$.
- Asigna a cada funtor $f/n \in \mathcal{F}$ la siguiente función $f_{\mathfrak{S}} : \mathcal{H}^n \rightarrow \mathcal{H}$:

$$f_{\mathfrak{S}}(t_1, \dots, t_n) = f(t_1, \dots, t_n) \text{ para todo } t_1, \dots, t_n \in \mathcal{H}$$

- Asigna a cada predicado $p/n \in \mathcal{P}$ una relación $p_{\mathfrak{S}} \subseteq \underbrace{\mathcal{H} \times \dots \times \mathcal{H}}_n$.

Para definir una interpretación de Herbrand particular bastará, por tanto, con definir las relaciones de términos base asociadas a cada predicado.

Ha de observarse que esto último, “definir las relaciones de términos base asociados a cada predicado”, puede hacerse mediante un subconjunto de la base de Herbrand; puesto que la base de Herbrand contiene *todas* las relaciones posibles entre términos base para todos los predicados.

Así, a partir de un subconjunto $\mathfrak{S} \subseteq \mathcal{B}$ de la base de Herbrand se define \mathfrak{S} como la interpretación de Herbrand tal que, para todo predicado $p/n \in \mathcal{P}$:

$$p_{\mathfrak{S}} = \{(t_1, \dots, t_n) \in \mathcal{H}^n \mid p(t_1, \dots, t_n) \in \mathfrak{S}\} \quad (3.1)$$

En adelante se identificará “interpretación de Herbrand” como “subconjunto de la base de Herbrand”.

3.2.4. Modelo de Herbrand

Se ha dicho que un programa lógico, es decir, un conjunto Γ formado por cláusulas de Horn

$$A \leftarrow \{A_1, \dots, A_n\}$$

expresa conocimiento declarativo positivo. Un *modelo de Herbrand* de un programa lógico será una interpretación de Herbrand que satisface todas las cláusulas del programa.

Debido a que las cláusulas del programa pueden contener variables libres, se dice que una interpretación de Herbrand \mathfrak{S} satisface una cláusula dada $\gamma \in \Gamma$, es decir $\mathfrak{S} \models \gamma$, si y sólo si $\mathfrak{S} \models_{\phi} \gamma$ para toda valuación $\phi : \mathcal{V} \rightarrow \mathcal{H}$ – de hecho esta definición ya se hizo con anterioridad –.

Hay, por tanto, una *cuantificación universal* implícita en las variables de las cláusulas de un programa lógico, sin necesidad de utilizar símbolos como “ \forall ”.

3.3. Modelos de Herbrand de un programa lógico

3.3.1. Sustitución, instancia y programa base

Se define una **sustitución** θ como una función parcial de variables a términos; es decir, un conjunto de pares de la siguiente forma, donde $t_i \in \mathcal{T}$, $x_i \in \mathcal{V}$ para $i = 1, \dots, n$ y $x_1 \neq \dots \neq x_n$:

$$\theta = \{t_1/x_1, \dots, t_n/x_n\}$$

Se define el *dominio* de una sustitución θ :

$$\text{Dom}(\{t_1/x_1, \dots, t_n/x_n\}) = \{x_1, \dots, x_n\}$$

La *aplicación* de una sustitución θ a un término $t \in \mathcal{T}$ es un nuevo término $t\theta \in \mathcal{T}$, y se define de la siguiente forma, para todo $c \in \mathcal{C}$, $x \in \mathcal{V}$, $f/n \in \mathcal{F}$:

- $c\theta = c$.
- $x\theta = \begin{cases} t & t/x \in \theta \\ x & t/x \notin \theta \text{ para ningún } t \end{cases}$.
- $f(t_1, \dots, t_n)\theta = f(t_1\theta, \dots, t_n\theta)$.

Propiedad : La siguiente importante propiedad permite relacionar sustituciones con valuaciones y se demuestra por inducción estructural. Para cualquier interpretación \mathfrak{S} , valuación ϕ y término t , se tiene que la semántica de aplicar la sustitución $\theta = \{t_1/x_1, \dots, t_n/x_n\}$ a t equivale a evaluar t en el contexto $\phi' = \phi\{\mathfrak{S}_\phi(t_1)/x_1, \dots, \mathfrak{S}_\phi(t_n)/x_n\}$:

$$\mathfrak{S}_\phi(t\theta) = \mathfrak{S}_{\phi'}(t) \tag{3.2}$$

La aplicación de una sustitución θ a un átomo $A = p(t_1, \dots, t_n)$ donde $p/n \in \mathcal{P}$ y $t_1, \dots, t_n \in \mathcal{T}$, denotada $A\theta$, se define como

$$(p(t_1, \dots, t_n))\theta = p(t_1\theta, \dots, t_n\theta)$$

Siendo e un término o átomo, se dice que e' es una instancia de e si $e' = e\theta$ para alguna sustitución θ . Se dice que e' es una *instancia base* si e' es una instancia tal que $\text{FV}(e') = \emptyset$.

Programa lógico base : El programa lógico base Γ' asociado a Γ resultará de sustituir todos los átomos que aparecen en las cláusulas de Γ por todas sus instancias base posibles:

$$\Gamma' = \{A_0\theta \leftarrow \{A_1\theta, \dots, A_n\theta\} \mid A_0 \leftarrow \{A_1, \dots, A_n\} \in \Gamma \text{ y } A_0\theta, \dots, A_n\theta \in \mathcal{B}\}$$

Manejar el programa lógico base permite abordar de forma cómoda la cuantificación universal implícita de las variables.

3.3.2. Formulación de un modelo de Herbrand

Modelo de Herbrand de átomos : Sea $\mathfrak{S} \subseteq \mathcal{B}$ cualquier interpretación de Herbrand y $p(t_1, \dots, t_n)$ cualquier átomo, donde $p/n \in \mathcal{P}$ y $t_1, \dots, t_n \in \mathcal{T}$. Aplicando la definición se tiene:

$\mathfrak{S} \models p(t_1, \dots, t_n)$ si y sólo si $\langle \mathfrak{S}_\phi(t_1), \dots, \mathfrak{S}_\phi(t_n) \rangle \in p_{\mathfrak{S}}$ para toda valuación ϕ .

Pero, teniendo en cuenta (3.1) y (3.2), esto puede reescribirse de la siguiente manera:

$\mathfrak{S} \models p(t_1, \dots, t_n)$ si y sólo si $(p(t_1, \dots, t_n))\theta \in \mathfrak{S}$ para todas las instancias base $(p(t_1, \dots, t_n))\theta$.

En el caso de considerar átomos cerrados (es decir, $p(t_1, \dots, t_n) \in \mathcal{B}$), se simplifica:

$\mathfrak{S} \models p(t_1, \dots, t_n)$ si y sólo si $p(t_1, \dots, t_n) \in \mathfrak{S}$.

Modelo de Herbrand de un programa base : Sea $\mathfrak{S} \subseteq \mathcal{B}$ cualquier interpretación de Herbrand. Se tiene por definición que, dada cualquier cláusula instanciada $(A \leftarrow \Delta) \in \Gamma'$ del programa base asociado a Γ :

$\mathfrak{S} \models A \leftarrow \Delta$ si y sólo si $\mathfrak{S} \not\models \Delta$ o $\mathfrak{S} \models A$ o ambos.

Como A y las fórmulas de Δ son átomos base, según lo anterior, se tiene:

$\mathfrak{S} \models A \leftarrow \Delta$ si y sólo si $\Delta \not\subseteq \mathfrak{S}$ o $A \in \mathfrak{S}$ o ambos.

que puede reescribirse de la siguiente manera:

$\mathfrak{S} \models A \leftarrow \Delta$ si y sólo si se tiene $A \in \mathfrak{S}$ si es el caso que $\Delta \subseteq \mathfrak{S}$. (3.3)

Esta última forma es la que se utilizará en adelante.

3.4. El menor modelo de Herbrand

3.4.1. Diferentes modelos de Herbrand de un programa lógico

Se ha dicho que un programa lógico define un modelo que satisface sus cláusulas, y que el objetivo de un intérprete lógico será decir si dicho modelo satisface una fórmula (atómica) arbitraria, y, si ésta contiene variables, para qué valuación de las mismas la satisface.

A modo de ejemplo, tómesese el siguiente alfabeto:

$$\begin{aligned} \mathcal{C} &= \{\text{ZERO}\} \\ \mathcal{F} &= \{\text{SUCC}/1\} \\ \mathcal{P} &= \{\text{EVEN}/1\} \end{aligned}$$

El universo de Herbrand asociado será el siguiente:

$$\mathcal{H} = \{\text{ZERO}, \text{SUCC}(\text{ZERO}), \text{SUCC}(\text{SUCC}(\text{ZERO})), \dots\}$$

Y la base de Herbrand será la siguiente:

$$\mathcal{B} = \{\text{EVEN}(\text{ZERO}), \text{EVEN}(\text{SUCC}(\text{ZERO})), \text{EVEN}(\text{SUCC}(\text{SUCC}(\text{ZERO}))), \dots\}$$

Sea Γ el programa lógico formado por las siguientes cláusulas:

$$\begin{aligned} & \text{EVEN}(\text{ZERO}) \\ & \text{EVEN}(\text{SUCC}(\text{SUCC}(N))) \leftarrow \text{EVEN}(N) \end{aligned}$$

¿Cuál es el modelo de Herbrand que define dicho programa? Es claro que hay muchas interpretaciones de Herbrand que satisfacen sus cláusulas. De hecho, de forma trivial, la propia base de Herbrand \mathcal{B} satisface las cláusulas del programa. En general:

Mayor modelo Herbrand : La base de Herbrand \mathcal{B} es un modelo de Herbrand de cualquier programa – y por tanto es el *mayor modelo Herbrand* de cualquier programa:

$$\mathcal{B} \models \Gamma \text{ para cualquier } \Gamma$$

Esta propiedad puede demostrarse de acuerdo con las propiedades mostradas en la sección anterior.

No obstante, retomando el ejemplo, es claro que el modelo que debe estar asociado al programa será aquél que esté definido de forma única por sus cláusulas; es decir, el siguiente:

$$\begin{aligned} I &= \{ \text{EVEN}(\text{ZERO}), \text{EVEN}(\text{SUCC}(\text{SUCC}(\text{ZERO}))), \\ & \quad \text{EVEN}(\text{SUCC}(\text{SUCC}(\text{SUCC}(\text{SUCC}(\text{ZERO}))))), \dots \} \\ &= \{ \text{EVEN}(\text{SUCC}^n(\text{ZERO})) \mid n = 0, 2, 4, \dots \} \end{aligned}$$

que será el *menor* de los modelos de Herbrand.

Menor modelo Herbrand : Una interpretación de Herbrand I es un *menor modelo Herbrand* del programa lógico Γ si

1. $I \models \Gamma$.
2. Para cualquier interpretación de Herbrand \mathfrak{S} , si $\mathfrak{S} \models \Gamma$, entonces $I \subseteq \mathfrak{S}$.

Es claro que si hay un menor modelo Herbrand de Γ , este es único, ya que si A y B fuesen dos menores modelos Herbrand de Γ , entonces $A \subseteq B$ por aplicación de la regla 2 para A y $B \subseteq A$ por la misma regla para B , con lo que se tiene $A = B$. Se denotará el menor modelo Herbrand de Γ como $I(\Gamma)$.

3.4.2. Construcción “desde arriba”: intersección de modelos

Dado el programa lógico Γ , la siguiente interpretación de Herbrand I es el menor modelo de Herbrand de Γ ($I = I(\Gamma)$):

$$I = \bigcap \{ \mathfrak{S} \mid \mathfrak{S} \models \Gamma \}$$

(En la definición está implícito que \mathfrak{S} ha de ser una interpretación de Herbrand).

Demostración: En primer lugar hay que demostrar la condición 1. Para ello, sea $A \leftarrow \Delta \in \Gamma'$ cualquier cláusula base. Supóngase $I \models \Delta$. En ese caso, puesto que $\Delta = \{A_1, \dots, A_n\}$ donde A_1, \dots, A_n son fórmulas atómicas cerradas, se tiene que $A_1, \dots, A_n \in I$. Y por tanto que para todo \mathfrak{S} tal que $\mathfrak{S} \models \Gamma$,

$A_1, \dots, A_n \in \mathfrak{S}$; y en consecuencia, necesariamente $A \in \mathfrak{S}$. Por tanto $A \in I$, quedando demostrada la condición 1.

La demostración de la condición 2 es inmediata teniendo en cuenta la definición de I .

Se ha demostrado que ésta es una construcción de $I(\Gamma)$ válida. No obstante, no es muy satisfactoria desde un punto de vista constructivo. A continuación se presentará una definición constructiva “desde abajo” del menor modelo Herbrand.

3.4.3. Construcción “desde abajo”: semántica del punto fijo

La formulación (3.3) de modelos de Herbrand (p. 17) sugiere la siguiente definición:

Operador de consecuencia inmediata : Un programa lógico Γ determina un operador $\widehat{\Gamma}$ sobre interpretaciones de Herbrand tal que $\widehat{\Gamma}(\mathfrak{S})$ es el conjunto de las fórmulas atómicas que son consecuencia directa en Γ , debido a alguna cláusula, de las de \mathfrak{S} (se considerarán átomos base).

$$\widehat{\Gamma}(\mathfrak{S}) = \{A \mid \exists \Delta : \Delta \subseteq \mathfrak{S}, A \leftarrow \Delta \in \Gamma'\}$$

Propiedad : Por definición, teniendo en cuenta (3.3), es claro que:

$$\mathfrak{S} \models \Gamma \text{ si y sólo si } \widehat{\Gamma}(\mathfrak{S}) \subseteq \mathfrak{S}. \quad (3.4)$$

Así, puede reformularse la definición de “menor modelo de Herbrand” de la siguiente forma: I es el menor modelo de Herbrand de Γ ($I = I(\Gamma)$) si y sólo si

1. $\widehat{\Gamma}(I) \subseteq I$.
2. Para cualquier \mathfrak{S} , si $\widehat{\Gamma}(\mathfrak{S}) \subseteq \mathfrak{S}$, entonces $I \subseteq \mathfrak{S}$.

Una propiedad interesante del operador $\widehat{\Gamma}$ es que es **monótono**: es decir, preserva el orden parcial \subseteq (se sigue directamente de la definición de $\widehat{\Gamma}$):

$$\text{Para cualquier } \mathfrak{S}_1, \mathfrak{S}_2, \text{ si } \mathfrak{S}_1 \subseteq \mathfrak{S}_2 \text{ entonces } \widehat{\Gamma}(\mathfrak{S}_1) \subseteq \widehat{\Gamma}(\mathfrak{S}_2)$$

Otra propiedad interesante es que si $\mathfrak{S} \models \Gamma$ entonces $\widehat{\Gamma}(\mathfrak{S})$ es un modelo de Γ menor que \mathfrak{S} : si $\widehat{\Gamma}(\mathfrak{S}) \subseteq \mathfrak{S}$, entonces $\widehat{\Gamma}(\widehat{\Gamma}(\mathfrak{S})) \subseteq \widehat{\Gamma}(\mathfrak{S})$ por la monotonía, y por tanto $\widehat{\Gamma}(\widehat{\Gamma}(\mathfrak{S})) \subseteq \mathfrak{S}$ obteniendo $\widehat{\Gamma}(\mathfrak{S}) \models \Gamma$.

Aplicando esta última propiedad al menor modelo de Herbrand de Γ , $I = I(\Gamma)$, se tiene que $\widehat{\Gamma}(I) \models \Gamma$ y que $\widehat{\Gamma}(I) \subseteq I$. Por tanto se tiene que $\widehat{\Gamma}(I) = I$ ya que el menor modelo de Herbrand es único, o por aplicación directa del punto 2 de la definición de I . Es decir, I es un *punto fijo* de $\widehat{\Gamma}$. Además, I no solo es un punto fijo de $\widehat{\Gamma}$ sino que es el *menor punto fijo*. Ya que si $\mathfrak{S} = \widehat{\Gamma}(\mathfrak{S})$ es cualquier punto fijo de $\widehat{\Gamma}$, entonces $I \subseteq \mathfrak{S}$ por aplicación directa de la regla 2.

Por tanto I sería una solución de la ecuación recursiva:

$$I = \{A \mid \exists \Delta : \Delta \subseteq I, A \leftarrow \Delta \in \Gamma'\}$$

(el caso base de la recursión viene dado para $\Delta = \emptyset$). Es importante recalcar que puede haber varios puntos fijos del operador $\widehat{\Gamma}$ (varias soluciones de la ecuación recursiva).⁴ Como se ha visto, I habrá de ser la menor de estas soluciones.

Lo que se necesita, pues, es una técnica para calcular este menor punto fijo. El método que se presentará a continuación es la aplicación particular de un resultado más general, el teorema de Knaster-Tarski. Véase [Apt88] o [Winskel93] para más detalles.

Obtención del menor punto fijo : Comenzando en la interpretación vacía \emptyset , la idea es aplicar el operador $\widehat{\Gamma}$ de forma sucesiva, de manera que en cada iteración se obtengan las fórmulas que, debido a una consecuencia (cláusula) de Γ , es necesario añadir para que la interpretación pueda modelar Γ . Se construye para ello la siguiente serie:

$$\emptyset, \widehat{\Gamma}(\emptyset), \widehat{\Gamma}(\widehat{\Gamma}(\emptyset)), \widehat{\Gamma}(\widehat{\Gamma}(\widehat{\Gamma}(\emptyset))), \dots$$

O escrito de otra forma:

$$\begin{aligned} T_0 &= \emptyset \\ \forall i \in \mathbb{N}, i > 0 \quad T_i &= \widehat{\Gamma}(T_{i-1}) \end{aligned}$$

(Otra notación para T_i es $\widehat{\Gamma}^i(\emptyset)$).

Esta serie es ascendente, es decir:

$$\emptyset \subseteq \widehat{\Gamma}(\emptyset) \subseteq \widehat{\Gamma}(\widehat{\Gamma}(\emptyset)) \subseteq \widehat{\Gamma}(\widehat{\Gamma}(\widehat{\Gamma}(\emptyset))) \subseteq \dots$$

Esto puede demostrarse por inducción. *Base*: Es claro que $T_0 \subseteq T_1$ puesto que $\emptyset \subseteq \widehat{\Gamma}(\emptyset)$. *Paso inductivo*: Si $T_i \subseteq T_{i+1}$, se tiene $\widehat{\Gamma}(T_i) \subseteq \widehat{\Gamma}(T_{i+1})$ por la monotonicidad de $\widehat{\Gamma}$, y por tanto $T_{i+1} \subseteq T_{i+2}$.

El menor punto fijo de $\widehat{\Gamma}$ se obtendrá mediante el *supremo* de la serie ascendente:

$$I = \bigcup_{n \in \mathbb{N}} \widehat{\Gamma}^n(\emptyset)$$

Teorema : El conjunto I anteriormente definido es el *menor punto fijo* de $\widehat{\Gamma}$. Es decir $I = I(\Gamma)$.

Demostración [Winskel93, pp. 52,53]: Se dividirá en tres partes: en primer lugar $\widehat{\Gamma}(I) \subseteq I$, en segundo lugar $I \subseteq \widehat{\Gamma}(I)$, demostrando por tanto que I es un punto fijo; y en tercer lugar que es el *menor* punto fijo.

- En primer lugar hay que demostrar $\widehat{\Gamma}(I) \subseteq I$, o, equivalentemente debido a (3.4), $I \models \Gamma$. Para ello, tómesese cualquier cláusula base $(A \leftarrow \Delta) \in \Gamma'$. Si $\Delta \subseteq I$, como $\Delta = \{A_1, \dots, A_k\}$, es claro que $A_1 \in T_{n_1}, \dots, A_k \in T_{n_k}$ (ya que $I = \bigcup_{n \in \mathbb{N}} T_n$) para unos ciertos números n_1, \dots, n_k . Pero tomando $m = \max\{n_1, \dots, n_k\}$, es claro que $T_{n_1} \subseteq T_m, \dots, T_{n_k} \subseteq T_m$ ya que la serie es ascendente. Por tanto $\Delta \subseteq T_m$. Y por definición, se tiene $A \in \widehat{\Gamma}(T_m)$ y por tanto $A \in \Delta$, habiendo demostrado $I \models \Gamma$ y en consecuencia $\widehat{\Gamma}(I) \subseteq I$.

⁴Piénsese por ejemplo en cláusulas de la forma $A \leftarrow A$.

- Se demostrará $I \subseteq \widehat{\Gamma}(I)$. Para ello basta tomar cualquier $A \in I$. Es claro que $A \in T_i$ para algún $i > 0$ (ya que $T_0 = \emptyset$). Y por tanto $A \in \widehat{\Gamma}(T_{i-1})$, es decir, existe un $(A \leftarrow \Delta) \in \Gamma'$ tal que $\Delta \subseteq T_{i-1}$. Pero $T_{i-1} \subseteq I$ con lo que se tiene $A \in \widehat{\Gamma}(I)$ por definición.
- Por último, se debe demostrar que para cualquier interpretación Herbrand \mathfrak{S} tal que $\widehat{\Gamma}(\mathfrak{S}) = \mathfrak{S}$, se tiene $I \subseteq \mathfrak{S}$. Se demostrará por inducción que $\forall i \in \mathbb{N} : T_i \subseteq \mathfrak{S}$, y por tanto $I \subseteq \mathfrak{S}$. *Base:* $T_0 \subseteq \mathfrak{S}$ es trivial ya que $T_0 = \emptyset$. *Paso inductivo:* Si $T_i \subseteq \mathfrak{S}$, se tiene $\widehat{\Gamma}(T_i) \subseteq \widehat{\Gamma}(\mathfrak{S})$ por la monotonicidad. Pero $\widehat{\Gamma}(\mathfrak{S}) = \mathfrak{S}$ y $\widehat{\Gamma}(T_i) = T_{i+1}$, por tanto se tiene $T_{i+1} \subseteq \mathfrak{S}$. Se demuestra por tanto que $I = I(\Gamma)$.

En conclusión, se ha demostrado que el menor modelo Herbrand de un programa lógico Γ se puede calcular con la siguiente expresión:

$$I(\Gamma) = \bigcup_{n \in \mathbb{N}} \widehat{\Gamma}^n(\emptyset)$$

El mayor punto fijo : Aunque no sea relevante de cara al menor modelo Herbrand de un programa lógico, resulta interesante señalar que el método presentado para calcular el menor punto fijo tiene una forma *dual*.

Se observó anteriormente que si $\widehat{\Gamma}(\mathfrak{S}) \subseteq \mathfrak{S}$, entonces $\widehat{\Gamma}(\widehat{\Gamma}(\mathfrak{S})) \subseteq \widehat{\Gamma}(\mathfrak{S})$.

Comenzando en la máxima interpretación de Herbrand \mathcal{B} , la idea es aplicar el operador $\widehat{\Gamma}$ de forma sucesiva obteniendo la siguiente serie descendente (su carácter descendente se sigue por inducción a partir de la observación anterior):

$$\dots \subseteq \widehat{\Gamma}(\widehat{\Gamma}(\widehat{\Gamma}(\mathcal{B}))) \subseteq \widehat{\Gamma}(\widehat{\Gamma}(\mathcal{B})) \subseteq \widehat{\Gamma}(\mathcal{B}) \subseteq \mathcal{B}$$

El mayor punto fijo de $\widehat{\Gamma}$, que se denotará aquí como J , se obtendrá mediante el *ínfimo* de la serie descendente:

$$J = \bigcap_{n \in \mathbb{N}} \widehat{\Gamma}^n(\mathcal{B})$$

Teorema : El conjunto J anteriormente definido es el *mayor punto fijo* de $\widehat{\Gamma}$.

Obsérvese que esta construcción es completamente dual a la anterior. De hecho, si el menor punto fijo I era también el *menor punto prefijo* del operador (menor I tal que $\widehat{\Gamma}(I) \subseteq I$, que en este contexto es el menor modelo de Herbrand), el mayor punto fijo J será también el *mayor punto postfijo* (mayor J tal que $J \subseteq \widehat{\Gamma}(J)$).

3.5. Conclusiones

En conclusión, el menor modelo de Herbrand de un programa lógico constituye la semántica declarativa formal del programa en el lenguaje “puro” basado en cláusulas de Horn que se ha presentado en esta sección.

Los intérpretes lógicos reales utilizan un proceso de inferencia, la *resolución SLD*, que proporciona una semántica operacional del lenguaje que es correcta de acuerdo con su semántica declarativa. En la práctica a veces se sacrifica la corrección en casos inusuales por razones de eficiencia, y tampoco resulta completa debido a la exploración en profundidad de árboles de resolución infinitos,

pudiendo por tanto la semántica operacional no coincidir con la declarativa. Por ejemplo, alterar el orden de las cláusulas de un programa, irrelevante para la semántica declarativa, podría causar que el programa no terminase en su semántica operacional.

Por otro lado, los lenguajes lógicos reales poseen además otras características como por ejemplo la negación, que no se ha abordado en este documento.

4. Referencias

4.1. Bibliografía

- [Apt88] K. R. Apt *Introduction to logic programming.*
Department of Computer Sciences, University of Texas at Austin, July 1988.
- U. Nilsson, J. Małuszyński *Logic, Programming and Prolog (2ed).*
- [Winskel93] G. Winskel *The Formal Semantics of Programming Languages.*
MIT Press, 1993.

4.2. Otras referencias

- L.C. Paulson *Logic and Proof.*
Lecture notes 2008/09, University of Cambridge.
<http://www.cl.cam.ac.uk/teaching/0809/LogicProof/>
- V. Goranko *Deductive systems. Soundness and completeness.*
Mathematical Logic Honours 2008 lecture notes,
University of the Witwatersrand.
<http://www.maths.wits.ac.za/~goranko/ML/ML08.html>