

RESOLUCIÓN EN LÓGICAS ANOTADAS SOBRE BIRRETÍCULOS

RICARDO NEFTALÍ RAMÍREZ OSORIO

UNIVERSIDAD DEL VALLE
FACULTAD DE CIENCIAS NATURALES Y EXACTAS
DEPARTAMENTO DE MATEMÁTICAS
SANTIAGO DE CALI
2009

RESOLUCIÓN EN LÓGICAS ANOTADAS SOBRE BIRRETÍCULOS

RICARDO NEFTALÍ RAMÍREZ OSORIO

Trabajo de grado presentado como requisito para optar al título de
Matemático

Director
GUILLERMO ORTÍZ RICO

UNIVERSIDAD DEL VALLE
FACULTAD DE CIENCIAS NATURALES Y EXACTAS
DEPARTAMENTO DE MATEMÁTICAS
SANTIAGO DE CALI
2009

UNIVERSIDAD DEL VALLE
FACULTAD DE CIENCIAS NATURALES Y EXACTAS
PROGRAMA ACADÉMICO DE MATEMÁTICAS

RICARDO NEFTALÍ RAMÍREZ OSORIO, 1982

RESOLUCIÓN EN LÓGICAS ANOTADAS SOBRE BIRRETÍCULOS

TEMAS RELACIONADOS

- Lógica de Primer Orden
- Lógicas anotadas
- Birretículos
- Lógicas anotadas sobre birretículos
- Resolución-SLD
- Programación lógica

Resumen

Este trabajo presenta los elementos básicos de la *lógica de Primer Orden*: su *sintaxis* y su *semántica*. Describimos un tipo especial de fórmula, llamado *formas normales*, que nos permiten automatizar los procesos; definimos las aplicaciones de *sustitución* y formalizamos el concepto de *Resolución* en el caso clásico. Luego enriquecemos el Principio de Resolución, por medio de una estrategia de refinamiento, obteniendo la *Resolución Lineal*, esta se define y se dan ejemplos.

Se define el Principio de *Resolución-SLD*, que consiste de una estrategia de Resolución Lineal junto con un criterio de selección de literales, aplicado a cláusulas definidas. Se hace una presentación detallada de la *semántica operacional, declarativa y de punto fijo*, del Principio de Resolución-SLD y se demuestra que es un método deductivo *valido* y *completo*. Todo esto sobre lógicas de Primer Orden.

Se describen las *lógicas Anotadas*, empleando como modelo a la lógica anotada Q_τ . Se elige como conjunto de anotación la estructura de *birretículo*, el cual se define y se exhiben algunas propiedades; y así, se reconstruye el algoritmo de Resolución-SLD y su semántica en lógica Anotada sobre birretículos.

Finalmente, se demuestra la *validez* y la *completez* del Principio de Resolución-SLD para lógicas Anotadas sobre birretículos; todo esto basado en el trabajo presentado por Komendantskaya[18].

Contenido

Introducción	v
1. Lógica de Primer Orden	1
1.1. Sintaxis	1
1.2. Semántica	5
1.3. Formas normales	11
1.4. Sustitución	22
1.5. Resolución	28
1.6. Resolución lineal	33
2. Programación lógica	39
2.1. Resolución-SLD	40
2.2. Semántica operacional	43
2.3. Semántica declarativa	49
2.4. Semántica de Punto Fijo	63
2.5. Validez y Completez -SLD	71
3. Lógicas anotadas	76
3.1. Lógicas anotadas $\mathcal{Q}_\tau$	77
3.2. Birretículos ($\mathcal{B}$)	78
3.3. Lógica anotada sobre $\mathcal{B}$	81
3.4. Semántica anotada en $\mathcal{B}$	82

3.5. Programas lógicos anotados en $\mathcal{B}$	86
3.6. Resolución-SLD anotada en $\mathcal{B}$	87
3.7. Semántica operacional anotada	89
3.8. Semántica declarativa anotada en $\mathcal{B}$	90
3.9. Semántica de Punto Fijo anotada en $\mathcal{B}$	92
4. Validez y Completez -SLD anotada	95
4.1. Validez -SLD en $\mathcal{B}$	95
4.2. Completez -SLD en $\mathcal{B}$	100
4.3. Conclusiones	103
A. Ejemplo Final	105
A.1. Planteamiento	105
A.2. Programa	107
A.3. Resolución	108
Referencias bibliográficas	114
ANEXOS	
Lista de símbolos	115
Índice alfabético	117

Índice de gráficas

1.1. Interpretación de fórmulas (Ejemplo 6)	9
1.2. Programa lógico	21
1.3. Programa lógico definido	22
1.4. Unificación (Ejemplo 17)	27
1.5. Refutación (Ejemplo 22)	33
1.6. Derivación lineal	34
1.7. Refutación lineal (Ejemplo 23)	35
1.8. Refutación lineal (Ejemplo 24)	36
1.9. Refutación lineal con selección de literal (Ejemplo 25)	37
2.1. Refutación-SLD (Ejemplo 28)	47
2.2. Diagrama de Hasse del retículo completo $\wp(\mathfrak{B}_{\mathfrak{A}})$ (Ejemplo 35) . .	56
3.1. Birretículo $\mathcal{B}_{25}$ (Ejemplo 46)	81
A.1. Resolución-SLD Objetivo 1 (Ejemplo Final)	109
A.2. Resolución-SLD Objetivo 2 (Ejemplo Final)	110

Introducción

Desde hace algunas décadas la comunidad informática se ha interesado por descubrir y exhibir formas de resolución de problemas e introducirlos en una computadora de tal forma que se pueda aplicar de manera iterativa un método, y se logren resultados correctos y de manera eficiente. Toda esta aspiración se encuentra reflejada en el desarrollo de los lenguajes de programación.

Durante los años 30, los trabajos realizados por Frege[7], Skolem[37] y Löwenheim[25], eran considerados los cimientos lógicos, a partir del *cálculo de predicados*, para la formación de los métodos deductivos. Pero en esta época se presentarían varios sucesos significativos, a saber:

- Herbrand[13] propuso varias versiones de métodos deductivos, y enuncia y demuestra el teorema que llevaría su nombre.
- Gödel[11] demuestra la completitud del cálculo proposicional.
- Tarski[40] desarrolló la primera teoría semántica para el cálculo de predicados.
- Turing[41] y Church[3] descubren de manera independiente la indecidibilidad del cálculo de predicados, suceso que tendría un reflejo negativo para las aspiraciones de la lógica computacional.

La lógica matemática comenzó a adquirir fuerza en el campo informático cuando se concibió la idea de emplear los avances lógicos del momento en las labores programables desempeñadas por un ordenador. En 1956, Newell inició el proceso al presentar su proyecto llamado “*Logic Theorist*”, proyecto que se convirtió en el primer programa diseñado para imitar las habilidades para resolver problemas de un ser humano y por eso se reconoce como “el primer programa de inteligencia artificial”. Luego, Davis y Putnam([27]) en 1960, definieron una nueva forma de expresión para las fórmulas del cálculo de predicados que permitiría sistematizar los procedimientos de deducción; esta expresión fue llamada “*forma clausal*”. Además, para estos tiempos Prawitz[32] empleaba lo que hoy conocemos como *unificación*.

Gilmore[9], Davis y Putnam realizaban propuestas de técnicas de deducción automáticas basadas en versiones de los métodos deductivos derivados del Teorema de Herbrand, los cuales generaban instancias básicas de un conjunto de cláusulas y se verificaba si eran insatisfacibles (o contradictorias). Sin embargo, y aunque Gilmore, Davis y Putnam desarrollaron métodos de verificación rápida de insatisfacibilidad; los conjuntos de instancias básicas de Herbrand podían llegar a ser muy extensos, y más aún si poseían símbolos de función, lo cual hacían estos métodos poco eficientes.

Robinson[34] quedó fascinado con el método de Davis y Putnam, y junto a la idea de *unificación* de Prawitz, desarrolló en 1965 un método deductivo al que llamó *Resolución*. Este método solucionaba el inconveniente de la generación (ilimitada) de instancias básicas, al introducir las reglas de resolución empleando la unificación. Así, estableció tres partes básicas del método deductivo:

- (i) Representar las fórmulas en *forma clausal*, que aunque no son equivalentes lógicamente si que lo son en cuestión de insatisfacibilidad.
- (ii) Obtener el conjunto de sustituciones, de máxima generalidad, de las variables por términos que unificaran las discordancias.
- (iii) Identificar las parejas de cláusulas resolubles, a partir de la unificación adecuada y las reglas de resolución.

Para ser precisos, el Principio de resolución identifica las parejas resolubles con la unificación pertinente y genera la cláusula resolvente; esta se añade al conjunto de cláusulas y se aplica reiteradamente el proceso hasta obtener: ó la cláusula vacía, y decimos que la deducción es correcta; ó bien no es posible aplicar más resoluciones, y decimos que la deducción no es correcta; ó el proceso no termina y esto se debe a la indecibilidad de la lógica de Primer Orden.

Green[12] en 1969, se destacó por el descubrimiento de que la unificación iba generando restricciones sobre las variables que al llegar, aplicando Resolución, a la cláusula vacía constituían una solución al problema y es por esto que se considera a Green el precursor de la programación lógica. Sin embargo, para el Principio de Resolución, cuando el número de cláusulas crece, el número de pasos para llegar a la deducción de la cláusula vacía puede crecer mucho, por esta razón, es necesario aplicar *criterios selectivos* que simplifiquen el proceso y lo hagan mucho mas eficiente y fácil de implementar.

Se estudia una *estrategia de refinamiento* llamada *Resolución Lineal*. Esta estrategia fue propuesta por Loveland[24] y Luckham[26] de manera independiente. Es exitosamente implementable y consiste en tomar una cláusula central (o de cabecera) y resolverla con otra cláusula, luego la cláusula resuelta se resuelve con otra cláusula y se aplica reiteradamente este proceso.

La estrategia de Resolución Lineal puede combinarse con un criterio de *selección de literal*; esta selección elimina alternativas de resolución y puede significar una disminución considerable en los procesos de búsqueda, ya que para llegar a la deducción de la cláusula vacía es necesario ir eliminando cada literal de la cláusula central (u objetivo), entonces podemos seleccionar primero cuál es el literal que se quiere resolver, y luego considerar solo las cláusulas (laterales) que permiten hacer esto.

El descubrimiento de Green y los avances de Loveland y Luckham, inspiró a Kowalski[22] a plantear un método muy eficiente conocido como *Resolución-SLD* (en inglés, **S**elective **L**inear for **D**efinite clauses), que consiste de la estrategia de Resolución Lineal junto con una aplicación de selección de literal sobre cláusulas definidas (o de Horn[14]), que son cláusulas con a lo más un literal positivo.

El trabajo independiente pero interactivo entre el grupo de investigación de Kowalski en Edimburgo (se dice que aportaron la parte teórica) y el grupo de investigación de Colmerauer en Montreal(1970) y Marsella(1973) (se dice que aportaron la realización), dieron lugar a la primera implementación de un lenguaje de programación lógica, conocido como *PROLOG*, acrónimo de PROgramación LOGica; este se basa en la lógica de Primer Orden y las cláusulas de Horn, e incorpora la estrategia de resolución más refinada: la *Resolución-SLD*.

Las investigaciones en programación lógica e inteligencia artificial se han hecho cada vez mas apetecidas por el sin número de aplicaciones y desarrollos que se presentan en campos de la ingeniería. En esta dirección, los estudios se han enfocado recientemente en ciertas lógicas denominadas *no-clásicas* con el fin de poder dar un manejo adecuado a la clase de inconsistencias que surgen cuando se almacena información de diferentes fuentes; una de ellas son las *Lógicas Anotadas*.

Las lógicas Anotadas fueron introducidas por Subrahmanian[39] en su artículo “*On the Semantics of Quantitative Logic Programs*” y desde ese momento ocuparon un lugar significativo en el desarrollo de la programación lógica y los métodos deductivos; ya que, enriquecieron de expresividad a las fórmulas de la lógica de Primer Orden; añadiendo a estas, información tomada de una estructura algebraica apropiada. Entre aquellas estructuras algebraicas se encuentra la de *birretículo*, un conjunto parcialmente ordenado que, desde su introducción realizada por Ginsberg[10], se han ido convirtiendo en uno de los conjuntos de anotaciones más representativos. Si desea ampliar conocimientos sobre birretículos, referirse al trabajo de Rodriguez[35].

A grandes rasgos, las *fórmulas* de una lógica anotada consiste de un par de la forma $\varphi_{\langle\mu\rangle}$, donde φ es una fórmula de un lenguaje de Primer Orden y μ es la anotación tomada de una estructura algebraica. Intuitivamente, μ puede ser interpretada como el grado de creencia o certeza, de la fórmula φ asociada, respecto a una interpretación asignada del lenguaje anotado.

Nuestro objetivo es demostrar que el Principio de Resolución-SLD para lógicas Anotadas, con sus anotaciones en una estructura de birretículo, es una estrategia deductiva válida y completa, todo esto de la manera presentada por Komendantskaya-Seda[20] en su artículo “*Sound and complete SLD-resolution for bilattice-based annotated logic programs*”.

Así, este trabajo se divide en cuatro capítulos. En el primero presentamos la teoría necesaria y suficiente de la lógica de Primer Orden, para formalizar el Principio de Resolución clásico; luego se describe una estrategia de refinamiento y se extiende el Principio de Resolución al Principio de Resolución Lineal.

El segundo capítulo comprende lo relacionado con la programación lógica. Definimos el Principio de Resolución-SLD, que consiste del Principio de Resolución Lineal junto con un criterio de selección de cláusulas, todo esto sobre programas lógicos definidos. Presentamos de manera detallada y apoyados en ejemplos, la semántica operacional, declarativa y de punto fijo para la Resolución-SLD. Teoría que nos permite finalizar el capítulo con la prueba de validez y completez del Principio de Resolución-SLD en lógicas de Primer Orden.

En el Capítulo tres, describimos las lógicas Anotadas, presentamos nuestro conjunto de anotación elegido: el birretículo, y algunas de sus propiedades; y de forma similar al capítulo dos, las definiciones y enunciados de la semántica operacional, declarativa y de punto fijo, para la Resolución-SLD en lógicas Anotadas sobre birretículos.

Finalmente, en el capítulo cuatro presentamos la prueba de la validez y completez del Principio de Resolución-SLD en lógicas Anotadas sobre birretículos, todo esto de la manera presentada por Komendantskaya-Seda[20].

Capítulo 1

Lógica de Primer Orden

En la Lógica Proposicional los elementos básicos son los átomos; a partir de los cuales se construyen las fórmulas. Un átomo representa una sentencia declarativa que puede ser verdadera o falsa, pero no ambas a la vez. Desafortunadamente, los átomos en la lógica proposicional no son suficientes para el sinnúmero de conceptos que deseamos expresar. Por esta razón, se introduce la noción de *cuantificadores*, se permite representar objetos concretos mediante *constantes* y se introducen *variables* que pueden ser afectadas por los cuantificadores. Además, se dota el lenguaje de símbolos para representar *relaciones* y *funciones* entre objetos o propiedades de los mismos. Es así como el lenguaje de la Lógica Proposicional quedará extendido a lo que se denomina lenguaje de Lógica de Primer Orden, que tiene gran expresividad y análisis deductivo como veremos a lo largo de este capítulo.

El objetivo de este capítulo es realizar una definición formal de toda la estructura empleada para la construcción del Principio de Resolución en la Lógica de Primer Orden, la cual extiende la estructura de la Lógica Proposicional. Si se desea saber más sobre la estructura en Lógica Proposicional se referencia a Fitting[5] y Ben-Ari[1].

1.1. Sintaxis

Se inicia el estudio definiendo algunos conceptos básicos que especifican sólo la forma de las expresiones permitidas, mientras que el significado del lenguaje (semántica) se estudia en la siguiente sección.

Definición 1.1. El *alfabeto* $\mathcal{A}$, conocido como *alfabeto de Primer Orden*, consiste de los siguientes conjuntos de símbolos:

- **Variables:** representadas por las últimas letras del abecedario.

- **Cuantificadores:** los dos usuales, el *universal* $\forall$ y el *existencial* $\exists$.
- **Conectivos:** representan los operadores lógicos clásicos, como $\wedge, \vee, \rightarrow, \leftrightarrow, \neg$, entre otros.
- **Auxiliares:** como paréntesis, comas y puntos.

Ahora, definimos otros de los símbolos que extienden el Lenguaje Proposicional. Estos símbolos serán: los de *relación*, para referirnos a las relaciones entre los objetos, los de *función*, propios de la matemática para referirse a cierto tipo de relaciones, y los de *constante*, para referirnos a objetos concretos. Todo símbolo de función y de relación tiene asociado un número en $\mathbb{N}$, llamado *aridad*, que indica el número de argumentos. Los argumentos pueden ser constantes, variables y, en general, lo que definiremos más adelante como *términos*.

Definición 1.2. Un *lenguaje de primer orden* está determinado por $\mathcal{C}, \mathcal{R}$ y $\mathcal{F}$, y se denota $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$, donde

- $\mathcal{C}$ representa un conjunto numerable de *constantes*,
- $\mathcal{R}$ representa un conjunto numerable de símbolos de *relación* (o símbolos de *predicado*) y,
- $\mathcal{F}$ representa un conjunto numerable de símbolos de *función*.

Además, $\mathcal{V}$ denota el conjunto numerable de variables de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$.

Definición 1.3. Un *término* de un lenguaje $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ se forma únicamente de la siguiente manera: ó es una variable, ó es una constante ó tiene la forma $f(t_1, \dots, t_n)$ donde f es una función de aridad n (o n -aria) y los $t_1, \dots, t_n$ son términos de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$.

Definición 1.4. Un término se dice que es *básico* si este no contiene variables.

EJEMPLO 1. Si f, g, h son símbolos de función 1-aria, 2-aria (binaria), 3-aria, respectivamente, v, w son variables y a, b, c constantes; algunos términos son:

- | | | |
|-------------|-------------------|----------------------------|
| • a | • w | • $f(b)$ |
| • $g(a, v)$ | • $h(f(c), a, w)$ | • $h(f(v), w, f(g(b, c)))$ |



Observación 1. Debido a que la definición de término en Lógica de Primer Orden es similar a la definición de fórmula en Lógica Proposicional, se permite emplear de manera análoga los principios de Inducción y de Recursión para términos de Primer Orden, tal como define Fitting[5, pág. 111] y Ortíz[30, pág. 32].

Al igual que en nuestro lenguaje convencional, solo cierto tipo de combinación de palabras forman una expresión con significado. Lo equivalente a estas expresiones en Lógica de Primer Orden, son combinaciones llamadas *Fórmulas Bien Formadas* o simplemente *fórmulas*.

Primero definimos la más sencilla de las expresiones del lenguaje: los *átomos*.

Definición 1.5. Un **átomo** o *fórmula atómica* de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ es cualquier expresión de la forma $\mathcal{R}(t_1, \dots, t_n)$ donde $\mathcal{R}$ es una símbolo de relación n -ario (perteneciente a $\mathcal{R}$) y $t_1, \dots, t_n$ son términos de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$.

Definición 1.6. Las **fórmulas** o *fórmulas bien formadas* de un lenguaje de primer orden $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ se forman únicamente de la siguiente manera:

- Todo *átomo* es una *fórmula*,
- Si $\mathcal{F}$ es una *fórmula*, entonces la negación $\neg \mathcal{F}$ también lo es.
- Si $\mathcal{F}$ y $\mathcal{G}$ son *fórmulas*, entonces $\mathcal{F} \circledast \mathcal{G}$ es una *fórmula*, donde $\circledast$ es un conector binario.
- Si $\mathcal{F}$ es una *fórmula* y v es una variable, entonces $(\forall v)\mathcal{F}$ y $(\exists v)\mathcal{F}$ también lo son.

Observación 2. Como enunciamos en la Observación 1, los Principios de Inducción y Recursión garantizan que la construcción de fórmulas este bien definida, y que sus significados o valores de verdad sean únicos.

Definición 1.7. Una fórmula se dice que es **básica** si esta no contiene variables.

EJEMPLO 2. Si $\mathcal{P}, \mathcal{Q}, \mathcal{R}$ son símbolos de relación 1-aria, 2-aria, 3-aria, respectivamente, f, g símbolos de función 1-aria, 2-aria, respectivamente, v, w son variables y a, b, c constantes; algunas fórmulas son:

- $\mathcal{P}(a)$
- $\neg \mathcal{Q}(w, c)$
- $\neg(\neg \mathcal{P}(w) \vee \mathcal{R}(w, c, v))$
- $(\exists w)(\mathcal{Q}(f(w), v) \rightarrow \mathcal{P}(g(w, v)))$
- $\mathcal{P}(f(b))$
- $\neg \mathcal{R}(v, g(a, w), b)$
- $(\forall v)(\mathcal{P}(g(v, f(c))))$
- $(\forall v)(\exists w)(\neg \mathcal{P}(f(w)) \wedge \mathcal{Q}(f(a), v))$



La definición que se da a continuación, es con el propósito de evitar cierto tipo de confusiones en la interpretación de la ocurrencia de variables en las fórmulas, debido al alcance que tenga un cuantificador sobre ellas. Para denotar esta característica diremos, como lo hace Fitting[5, pág. 99], que la variable está *ligada* ó *libre*.

Definición 1.8. La ocurrencia de una **variable libre** en una *fórmula* $\mathcal{F}$ esta definida como sigue:

- Las ocurrencias de variables *libres* en un *átomo* son todas las ocurrencias de variables en esta fórmula.
- Las ocurrencias de variables *libres* en $\neg \mathcal{F}$ son las ocurrencias de variables *libres* en $\mathcal{F}$.
- Las ocurrencias de variables *libres* en $(\mathcal{F}_1 \diamond \mathcal{F}_2)$ son las ocurrencias de variables *libres* en $\mathcal{F}_1$ junto con las ocurrencias de variables *libres* en $\mathcal{F}_2$, donde $\diamond$ es un conectivo binario.
- Las ocurrencias de variables *libres* en $(\forall v)\mathcal{F}$ y $(\exists v)\mathcal{F}$ son las ocurrencias de variables *libres* en $\mathcal{F}$, excepto para las ocurrencias de v .

La ocurrencia de una variable es llamada **ligada** si esta no es *libre*.

En otras palabras, diremos que la ocurrencia de una variable en una fórmula es *ligada* si actúa un cuantificador sobre ella. En otro caso, la ocurrencia es *libre*.

EJEMPLO 3. Obsérvense las variables en las dos fórmulas siguientes:

- (i) $(\forall v)((\mathcal{R}(v, w) \wedge \mathcal{P}(v, w)) \rightarrow \exists w \mathcal{Q}(v, w))$
- (ii) $(\exists v)(\mathcal{Q}(v, w))$

La variable v esta *ligada* en ambas fórmulas. La variable w está *libre* en (ii) y en el antecedente de (i), y *ligada* en el consecuente.



Definición 1.9. Una **sentencia** o llamada **fórmula cerrada** de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ es una fórmula de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ sin ocurrencias de variables *libres*.

1.2. Semántica

En la sección anterior se definió los objetos que forman nuestro lenguaje en estudio. Ahora, nos interesa establecer el significado que tendrá cada uno de estos objetos, es decir, diremos como se interpretan las constantes, símbolos de relación y símbolos de función. Para ello emplearemos una función de *interpretación*, que asigna a cada objeto del lenguaje un objeto en un conjunto, distinto de vacío, llamado *Dominio* o *Universo*.

Además, dado que las fórmulas pueden tener ocurrencias de variables libres, diremos lo que representan en nuestro *Dominio* por medio de una función de *asignación*. Este desarrollo lo hacemos con base en el trabajo de Celma[2], Fitting[5] y Ortíz[29].

Definición 1.10. Una **estructura** para el lenguaje de Primer Orden $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ es un par $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{I} \rangle$, donde $\mathcal{D}$ es un conjunto no vacío llamado el **Dominio** de $\mathfrak{E}$, e $\mathfrak{I}$ es una función, llamada **Interpretación**, que asocia:

- A cada constante $c \in \mathcal{C}$, un elemento $c^{\mathfrak{I}}$ en $\mathcal{D}$.
- A cada símbolo de función n -ario $f \in \mathcal{F}$, una función n -aria $f^{\mathfrak{I}} : \mathcal{D}^n \rightarrow \mathcal{D}$.
- A cada símbolo de relación n -ario $\mathcal{R} \in \mathcal{R}$, una relación n -aria $\mathcal{R}^{\mathfrak{I}} \subseteq \mathcal{D}^n$, es decir, $\mathcal{R}^{\mathfrak{I}}$ será un conjunto formado por n -uplas de elementos de $\mathcal{D}$.

EJEMPLO 4. Sea el lenguaje que incluye las constantes a, b y c , un símbolo de relación $\mathcal{Q}$ de aridad 2 “...es múltiplo de...” y otro $\mathcal{R}$ de aridad 1 “...es un número par.”.

Consideremos la siguiente estructura $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{I} \rangle$ siendo $\mathcal{D} = \{1, 2, \dots, 6\}$. Una interpretación $\mathfrak{I}$ definida para este lenguaje, puede ser:

$$\begin{aligned} a^{\mathfrak{I}} &= 1, & b^{\mathfrak{I}} &= 2, & c^{\mathfrak{I}} &= 6, \\ \mathcal{Q}^{\mathfrak{I}} &= \{(5, 1)(4, 2), (6, 2), (6, 1), (2, 1), (6, 3)\} \\ \mathcal{R}^{\mathfrak{I}} &= \{2, 6\} \end{aligned}$$

Otra interpretación $\mathfrak{I}^*$, para el mismo lenguaje, será:

$$\begin{aligned} a^{\mathfrak{I}^*} &= 2, & b^{\mathfrak{I}^*} &= 3, & c^{\mathfrak{I}^*} &= 5, \\ \mathcal{Q}^{\mathfrak{I}^*} &= \{(5, 1)(4, 2), (6, 2), (2, 1), (3, 1), (1, 1)\} \\ \mathcal{R}^{\mathfrak{I}^*} &= \{4\} \end{aligned}$$



Definición 1.11. Una *asignación* en una estructura $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ es una función $\mathfrak{A} : \mathcal{V} \rightarrow \mathcal{D}$, es decir, una función del conjunto de variables de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ al dominio de $\mathfrak{E}$. Denotamos la imagen de la variable v bajo una *asignación* $\mathfrak{A}$ por $v^{\mathfrak{A}}$.

La interpretación de los elementos que conforman el lenguaje $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ nos ofrece la manera de asignar valores de verdad a las fórmulas de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$. El valor de verdad de una fórmula de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ se define a partir de la función de *interpretación* y de *asignación* sobre sus componentes. Primero, definiremos la interpretación de los *términos*, ya que son los componentes de las fórmulas.

Definición 1.12. Sea $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ una estructura para el lenguaje $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ y sea $\mathfrak{A}$ una asignación para esta estructura. A cada término t de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ se asocia el valor $t^{\mathcal{I}, \mathfrak{A}}$ en $\mathcal{D}$ como sigue:

- Para una constante c , $c^{\mathcal{I}, \mathfrak{A}} = c^{\mathcal{I}}$.
- Para una variable v , $v^{\mathcal{I}, \mathfrak{A}} = v^{\mathfrak{A}}$.
- Para un símbolo de función n -aria f , $[f(t_1, \dots, t_n)]^{\mathcal{I}, \mathfrak{A}} = f^{\mathcal{I}}(t_1^{\mathcal{I}, \mathfrak{A}}, \dots, t_n^{\mathcal{I}, \mathfrak{A}})$, donde $t_1, \dots, t_n$ son términos de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$.

Observación 3. Note que en el último item de la definición anterior la interpretación de la función y sus componentes se obtiene evaluando el símbolo de función en $\mathcal{D}$ y aplicando este (nuevo) símbolo de función a las componentes interpretadas, donde estas últimas se obtienen por aplicación recursiva de esta definición. Gracias al Principio de Recursión, del que se habló en las Observaciones 1 y 2, se garantiza la asignación de un único valor en $\mathcal{D}$ a cada término del lenguaje $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$.

Observación 4. En el caso de términos básicos, el valor asociado $t^{\mathcal{I}, \mathfrak{A}}$ en $\mathcal{D}$ no depende de la función de asignación $\mathfrak{A}$ de la estructura $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$. Simplemente, en este caso, escribiremos $t^{\mathcal{I}}$ en lugar de $t^{\mathcal{I}, \mathfrak{A}}$, para enfatizar este punto.

EJEMPLO 5. Supongamos que el lenguaje tiene una constante a , los símbolos de función f, g y h , unarias y binaria, respectivamente, y v una variable. De aquí, algunos de los términos son:

- $h(f(v), f(g(f(a))))$
- $h(h(f(a), a), h(f(v), f(g(f(a))))))$

Consideremos dos estructuras $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ y asignaciones $\mathfrak{A}$ para este lenguaje:

(i) Sea $\mathcal{D} = \mathbb{Z}$, $a^{\mathfrak{J}} = 1$, $f^{\mathfrak{J}}$ es la función *sucesor*, $g^{\mathfrak{J}}$ es la función *predecesor*, $h^{\mathfrak{J}}$ es la función producto y la asignación $\mathfrak{A}$, descrita por $v^{\mathfrak{A}} = 2$. Entonces,

- $[h(f(v), f(g(f(a))))]^{\mathfrak{J}, \mathfrak{A}} = 6$, veamos:

$$\begin{aligned} [h(f(v), f(g(f(a))))]^{\mathfrak{J}, \mathfrak{A}} &= h^{\mathfrak{J}}(f^{\mathfrak{J}}(v^{\mathfrak{A}}), f^{\mathfrak{J}}(g^{\mathfrak{J}}(f^{\mathfrak{J}}(a^{\mathfrak{J}})))) \\ &= h^{\mathfrak{J}}(f^{\mathfrak{J}}(2), f^{\mathfrak{J}}(g^{\mathfrak{J}}(f^{\mathfrak{J}}(1)))) = h^{\mathfrak{J}}(3, f^{\mathfrak{J}}(g^{\mathfrak{J}}(2))) \\ &= h^{\mathfrak{J}}(3, f^{\mathfrak{J}}(1)) = h^{\mathfrak{J}}(3, 2) = 6 \end{aligned}$$

- $[h(h(f(a), a), h(f(v), f(g(f(a)))))]^{\mathfrak{J}, \mathfrak{A}} = 12$, veamos:

$$\begin{aligned} [h(h(f(a), a), h(f(v), f(g(f(a)))))]^{\mathfrak{J}, \mathfrak{A}} &= h^{\mathfrak{J}}(h^{\mathfrak{J}}(f^{\mathfrak{J}}(a^{\mathfrak{J}}), a^{\mathfrak{J}}), 6) \\ &= h^{\mathfrak{J}}(h^{\mathfrak{J}}(f^{\mathfrak{J}}(1), 1), 6) \\ &= h^{\mathfrak{J}}(h^{\mathfrak{J}}(2, 1), 6) \\ &= h^{\mathfrak{J}}(2, 6) = 12 \end{aligned}$$

(ii) Sea $\mathcal{D}$ la colección de todas las palabras sobre el alfabeto $\{b, c\}$, $a^{\mathfrak{J}} = b$, $f^{\mathfrak{J}}$ es la operación que añade la c al final de la palabra, $g^{\mathfrak{J}}$ es la operación que añade la b al principio de la palabra, $h^{\mathfrak{J}}$ es la concatenación y la asignación $\mathfrak{A}$, descrita por $v^{\mathfrak{A}} = bcb$. Entonces,

- $[h(f(v), f(g(f(a))))]^{\mathfrak{J}, \mathfrak{A}} = bcbcbbcc$, veamos:

$$\begin{aligned} [h(f(v), f(g(f(a))))]^{\mathfrak{J}, \mathfrak{A}} &= h^{\mathfrak{J}}(f^{\mathfrak{J}}(v^{\mathfrak{A}}), f^{\mathfrak{J}}(g^{\mathfrak{J}}(f^{\mathfrak{J}}(a^{\mathfrak{J}})))) \\ &= h^{\mathfrak{J}}(f^{\mathfrak{J}}(bcb), f^{\mathfrak{J}}(g^{\mathfrak{J}}(f^{\mathfrak{J}}(b)))) \\ &= h^{\mathfrak{J}}(bcbcb, f^{\mathfrak{J}}(g^{\mathfrak{J}}(bc))) \\ &= h^{\mathfrak{J}}(bcbcb, f^{\mathfrak{J}}(bbc)) \\ &= h^{\mathfrak{J}}(bcbcb, bbcc) = bcbcbbcc \end{aligned}$$

- $[h(h(f(a), a), h(f(v), f(g(f(a)))))]^{\mathfrak{J}, \mathfrak{A}} = bcbcbcbbcc$, veamos:

$$\begin{aligned} [h(h(f(a), a), h(f(v), f(g(f(a)))))]^{\mathfrak{J}, \mathfrak{A}} &= h^{\mathfrak{J}}(h^{\mathfrak{J}}(f^{\mathfrak{J}}(a^{\mathfrak{J}}), a^{\mathfrak{J}}), bcbcbbcc) \\ &= h^{\mathfrak{J}}(h^{\mathfrak{J}}(f^{\mathfrak{J}}(b), b), bcbcbbcc) \\ &= h^{\mathfrak{J}}(h^{\mathfrak{J}}(bc, b), bcbcbbcc) \\ &= h^{\mathfrak{J}}(bcb, bcbcbbcc) = bcbcbcbbcc \end{aligned}$$

◆

Ahora, se asocia un valor de verdad a las *fórmulas* del lenguaje de Primer Orden $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$, a partir de la estructura $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{J} \rangle$ y la función de asignación $\mathfrak{A}$ de este lenguaje. Emplearemos la siguiente definición para interpretar el valor de verdad en las fórmulas cuantificadas.

Definición 1.13. Sean $\mathfrak{A}, \mathfrak{B}$ asignaciones en la estructura $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ y v una variable. Diremos que $\mathfrak{A}$ y $\mathfrak{B}$ son v – **variantes** si asignan los mismos valores (en $\mathcal{D}$) para cada variable excepto, posiblemente, en v .

Debido a que el desarrollo de este capítulo es sobre Lógica Clásica, el conjunto de valores de verdad para las fórmulas prevalece, siendo éste el bivalente $\{0, 1\}$ donde, 0 representa lo *falso* y 1 lo *verdadero*.

Definición 1.14. Sean $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ una estructura del lenguaje $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ y $\mathfrak{A}$ una asignación para esta estructura. Entonces, a cada fórmula de primer orden $\mathcal{F}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ se asocia el valor de verdad $\mathcal{F}^{\mathfrak{A}}$ como sigue:

- Para los átomos, $[\mathcal{R}(t_1, \dots, t_n)]^{\mathfrak{A}} = 1$ sí, y solo sí, $(t_1^{\mathfrak{A}}, \dots, t_n^{\mathfrak{A}}) \in \mathcal{R}^{\mathfrak{I}}$.
- $[\neg \mathcal{F}]^{\mathfrak{A}} = \neg [\mathcal{F}^{\mathfrak{A}}]$.
- $[\mathcal{F}_1 \diamond \mathcal{F}_2]^{\mathfrak{A}} = \mathcal{F}_1^{\mathfrak{A}} \diamond \mathcal{F}_2^{\mathfrak{A}}$, para $\diamond$ un conectivo binario.
- $[(\forall v)\mathcal{F}]^{\mathfrak{A}} = 1$ sí, y solo sí, $\mathcal{F}^{\mathfrak{B}} = 1$, para *toda* asignación $\mathfrak{B}$ en $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ que es v -variante de $\mathfrak{A}$.
- $[(\exists v)\mathcal{F}]^{\mathfrak{A}} = 1$ sí, y solo sí, $\mathcal{F}^{\mathfrak{B}} = 1$, para *alguna* asignación $\mathfrak{B}$ en $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ que es v -variante de $\mathfrak{A}$.

Observación 5. El segundo ítem nos muestra que si el valor de verdad de una fórmula $\mathcal{F}$ es 1 (Verdadero), entonces el valor de verdad de la misma fórmula negada $\neg \mathcal{F}$ es 0 (Falso) y viceversa, todo esto gracias nuevamente al Principio de Recursión (ver Fitting[5, pág. 111]).

Observación 6. Al igual que se indicó en la Observación 4 para los términos básicos, el valor asociado $\mathcal{F}^{\mathfrak{A}}$ en $\{0, 1\}$ para una fórmula básica y/o sentencia $\mathcal{F}$, no depende de la función de asignación $\mathfrak{A}$ de la estructura $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ y escribiremos $\mathcal{F}^{\mathfrak{I}}$ en lugar de $\mathcal{F}^{\mathfrak{A}}$, para enfatizar este punto. En otras palabras, el valor de verdad de fórmulas que tienen ocurrencias de variables libres, será subjetivo a la función de asignación de la estructura $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ donde se encuentra definida.

En conclusión, la *estructura*, formada por el *Dominio* y la función de *interpretación*, y la *asignación* nos permiten conocer el valor de verdad de cualquier fórmula de un Lenguaje de Primer Orden.

EJEMPLO 6. Con base en las estructuras dadas en el Ejemplo 4, el valor de verdad para las fórmulas:

- $\mathcal{R}(4)$
- $(\exists v)\mathcal{Q}(v, 3)$
- $(\forall v)\neg\mathcal{R}(v)$
- $(\forall v)(\exists w)\mathcal{Q}(v, w)$

esta dado en la Gráfica 1.1. Observe que no se depende de la función de asignación, debido a que las fórmulas planteadas no tienen ocurrencias de variables libres. ◆

FÓRMULA	INTERP/CIÓN		DESCRIPCIÓN
	$\mathfrak{I}$	$\mathfrak{I}^*$	
$\mathcal{R}(4)$	0	1	En $\mathfrak{I}$ es 0 por que el $4 \notin \mathcal{R}^{\mathfrak{I}}$, mientras que el $4 \in \mathcal{R}^{\mathfrak{I}^*}$.
$(\forall v)\neg\mathcal{R}(v)$	0	0	En ambos es 0, ya que $2 \in \mathcal{R}^{\mathfrak{I}}$ y $4 \in \mathcal{R}^{\mathfrak{I}^*}$, es decir, existe una interpretación para la cual $[\neg\mathcal{R}(v)]^{\mathfrak{I}} = 0$.
$(\exists v)\mathcal{Q}(v, 3)$	1	0	En $\mathfrak{I}$ es 1, ya que $(6, 3) \in \mathcal{Q}^{\mathfrak{I}}$, mientras que en $\mathfrak{I}^*$ es 0, ya que ningún número es múltiplo de 3.
$(\forall v)(\exists w)\mathcal{Q}(v, w)$	0	1	En $\mathfrak{I}$ es 0 por que 1 y 3 no son múltiplos de algún valor en $\mathcal{D}$, mientras que en $\mathfrak{I}^*$ para todos los valores de v existe un w que se relaciona.

Gráfica 1.1. Interpretación de fórmulas (Ejemplo 6)

Definición 1.15. Una fórmula $\mathcal{F}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ es *interpretada en 1* por una estructura $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{I} \rangle$ de este lenguaje, si $[\mathcal{F}]^{\mathfrak{I}, \mathfrak{A}} = 1$ para *toda* asignación $\mathfrak{A}$ de $\mathfrak{E}$. En este caso diremos que: $\mathfrak{E}$ es un **modelo** para $\mathcal{F}$, $\mathfrak{E}$ *satisface* a $\mathcal{F}$ ó $\mathfrak{E}$ interpreta a $\mathcal{F}$ en verdadero.

Definición 1.16. Una fórmula $\mathcal{F}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ es **satisfacible** si existe una estructura $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{I} \rangle$ y una *asignación* $\mathfrak{A}$ (llamada *asignación de satisfacción*) tal que $[\mathcal{F}]^{\mathfrak{I}, \mathfrak{A}} = 1$.

Definición 1.17. Una fórmula $\mathcal{F}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ es **válida** (o *universalmente válida*) si $\mathcal{F}$ es *interpretada en 1* para *todas* las estructuras de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$, es decir, si $[\mathcal{F}]^{\mathfrak{I}, \mathfrak{A}} = 1$ para todas las interpretaciones $\mathfrak{I}$ y asignaciones $\mathfrak{A}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$, es decir, si *todas* las estructuras de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ son *modelos* para $\mathcal{F}$. Esto lo denotamos como $\models \mathcal{F}$.

Definición 1.18. Una fórmula $\mathcal{F}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ es *insatisfacible* (o *contradictoria*), si para *todas* las estructuras $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{I} \rangle$ y/o *todas* las asignaciones $\mathfrak{A}$ en $\mathfrak{E}$, se tiene que $[\mathcal{F}]^{\mathfrak{I}, \mathfrak{A}} = 0$, es decir, si ninguna estructura es un *modelo* de $\mathcal{F}$.

Definición 1.19. Una fórmula $\mathcal{F}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ es *no-válida* (o *falsificable*) si $[\mathcal{F}]^{\mathfrak{I}, \mathfrak{A}} = 0$ para *alguna* interpretación $\mathfrak{I}$ y/o asignación $\mathfrak{A}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$.

Definición 1.20. Denotaremos por $\mathcal{S}$ a un conjunto de fórmulas de Primer Orden. Diremos que $\mathcal{S}$ es *válido* o *satisfacible* si cada una de las fórmulas de $\mathcal{S}$ es *válida* o *satisfacible*, respectivamente. Diremos que $\mathcal{S}$ es *insatisfacible* si por lo menos una de las fórmulas de $\mathcal{S}$ es *insatisfacible*, es decir, si en cada una de las posibles estructuras o asignaciones existe por lo menos una cláusula interpretada en 0. Además, una estructura $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{I} \rangle$ es un *modelo* para $\mathcal{S}$ si $\mathfrak{E}$ es un modelo para cada una de las fórmulas de $\mathcal{S}$.

La siguiente proposición será de gran relevancia para nuestro estudio, ya que enuncia la equivalencia existente entre una función que se satisface por todas las estructuras y su negación que no la satisface ninguna, debido a que es más “fácil” operar en busca de una estructura que no satisface la fórmula, que en probar que todas las estructuras lo hagan; este procedimiento será el pilar de nuestro método deductivo que construiremos más adelante y se conocerá como *refutación*.

Proposición 1.1.

Sea $\mathcal{F}$ una fórmula de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$. $\mathcal{F}$ es *válida* sí, y solo sí, $\neg \mathcal{F}$ es *insatisfacible*.

Demostración.

Sea $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{I} \rangle$ una estructura cualquiera de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$. Si $\mathcal{F}$ es válida, $\mathfrak{E}$ es un *modelo* para $\mathcal{F}$, es decir, $[\mathcal{F}]^{\mathfrak{I}, \mathfrak{A}} = 1$ para toda asignación $\mathfrak{A}$ en $\mathfrak{E}$ sí, y solo sí, $[\neg \mathcal{F}]^{\mathfrak{I}, \mathfrak{A}} = 0$ para toda asignación $\mathfrak{A}$ en $\mathfrak{E}$ ($\mathfrak{E}$ arbitraria), es decir, $\neg \mathcal{F}$ es *insatisfacible*. ■

Desde la Definición 1.15 hasta 1.20, hemos hablado de la forma de clasificar una fórmula dependiendo el valor de verdad que se le asigna en una estructura. Ahora, después de reunir un conjunto de fórmulas que sean *válidas*, deseamos conocer que podemos deducir de un conjunto de este tipo. A tal noción la llamaremos *consecuencia lógica* y se define a continuación.

Definición 1.21. Sea $\mathcal{S}$ un conjunto de fórmulas de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$. Diremos que $\mathcal{F}$ es una **consecuencia lógica** de $\mathcal{S}$ sí, y solo sí, $\mathcal{F}$ es verdadera para todos los modelos de $\mathcal{S}$; y lo denotamos por $\mathcal{S} \models \mathcal{F}$.

Proposición 1.2.

Sea $\mathcal{S}$ un conjunto de sentencias de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ y $\mathcal{F}$ una sentencia. Entonces, $\mathcal{S} \models \mathcal{F}$ sí, y solo sí, $\mathcal{S} \cup \{\neg \mathcal{F}\}$ es *insatisfacible*.

El concepto de *insatisfacibilidad* es muy importante a lo largo de este trabajo, ya que, empleando la proposición anterior, nos brinda la manera de determinar si una fórmula es consecuencia lógica de un conjunto de fórmulas dado.

Definición 1.22. Dos fórmulas $\mathcal{F}$ y $\mathcal{G}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$ se dice que son **equivalentes lógicamente**, denotado por $\mathcal{F} \equiv \mathcal{G}$, sí y solo sí, $\mathcal{F}$ y $\mathcal{G}$ tienen el mismo valor de verdad para todas las estructuras $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ y asignaciones $\mathfrak{A}$ de $\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$.

1.3. Formas normales

Continuando con el objetivo de este capítulo, deseamos definir un tipo de fórmulas de Primer Orden de tal manera que nos permita automatizar los procesos de deducción que se realicen sobre ellas; para ello, definiremos primero una *forma normal*, luego un método conocido como *Skolemización* y finalmente, la *cláusula*.

La siguiente definición debía estar en la primer sección donde se definían los elementos que usaríamos en este capítulo; se define aquí solamente para darle continuidad a la teoría planteada en esta sección.

Definición 1.23. Un **literal** es un átomo o la negación de un átomo. En el primer caso se llama **literal positivo** y en el segundo **literal negativo**.

Un par de literales $\mathfrak{l}_1$ y $\mathfrak{l}_2$ se dicen **complementarios** si $\mathfrak{l}_1 = \neg \mathfrak{l}_2$.

Definición 1.24. Una fórmula está en **forma normal conjuntiva (FNC)** (o *forma clausal*) si es una conjunción de disyunción de *literales*.

La prueba del siguiente teorema se logra por medio de un algoritmo descrito en Ben-Ari[1, pág. 65] y Ortíz[30, pág. 15]; empleando el hecho de que en toda fórmula se pueden sustituir sus conectivos, de forma equivalente por los conectivos $\{\neg, \wedge, \vee\}$, esto se debe a que $\{\neg, \wedge, \vee\}$ es un conjunto *completo* de conectivos, es decir, todos los otros conectivos se pueden definir en términos de ellos.

Teorema 1.

Toda fórmula puede ser transformada en una fórmula equivalente en *forma normal conjuntiva*.

Demostración.

La demostración es constructiva, ya que da el algoritmo para hallar la forma normal conjuntiva de una fórmula dada. Sean $\mathcal{F}, \mathcal{G}$ y $\mathcal{H}$ fórmulas.

Algoritmo FNC

Basta aplicar los siguientes pasos, que consisten en emplear equivalencias lógicas, tantas veces como sea posible:

Paso 1 Eliminar las ocurrencias de los conectivos lógicos $\rightarrow$ y $\leftrightarrow$ aplicando:

$$\begin{aligned}\mathcal{F} \leftrightarrow \mathcal{G} &\equiv (\mathcal{F} \rightarrow \mathcal{G}) \wedge (\mathcal{G} \rightarrow \mathcal{F}) \\ \mathcal{F} \rightarrow \mathcal{G} &\equiv \neg \mathcal{F} \vee \mathcal{G}\end{aligned}$$

Paso 2 Para “introducir” el conectivo $\neg$, aplicar las *leyes de Morgan* y para eliminar negaciones continuas la siguiente equivalencia:

$$\begin{aligned}\neg \neg \mathcal{F} &\equiv \mathcal{F} \\ \neg(\mathcal{F} \vee \mathcal{G}) &\equiv \neg \mathcal{F} \wedge \neg \mathcal{G} \\ \neg(\mathcal{F} \wedge \mathcal{G}) &\equiv \neg \mathcal{F} \vee \neg \mathcal{G}\end{aligned}$$

Paso 3 Aplicar las leyes distributivas de izquierda a derecha:

$$\begin{aligned}\mathcal{F} \vee (\mathcal{G} \wedge \mathcal{H}) &\equiv (\mathcal{F} \vee \mathcal{G}) \wedge (\mathcal{F} \vee \mathcal{H}) \\ \mathcal{F} \wedge (\mathcal{G} \vee \mathcal{H}) &\equiv (\mathcal{F} \wedge \mathcal{G}) \vee (\mathcal{F} \wedge \mathcal{H})\end{aligned}$$

Debido a que en todo el proceso se empleo equivalencias lógicas, la fórmula resultante es equivalente lógicamente a la fórmula inicial. ■

EJEMPLO 7. Encontraremos la forma normal conjuntiva de la fórmula

$$(\neg \mathcal{P}(v) \rightarrow \neg \mathcal{Q}(w)) \rightarrow (\mathcal{P}(w) \rightarrow \mathcal{Q}(v))$$

aplicando los tres pasos del *Algoritmo FNC*, donde $\mathcal{P}$ y $\mathcal{Q}$ son símbolos de relación, veamos:

$$\begin{aligned}
 & (\neg \mathcal{P}(v) \rightarrow \neg \mathcal{Q}(w)) \rightarrow (\mathcal{P}(w) \rightarrow \mathcal{Q}(v)) \\
 & \Leftrightarrow (\neg \neg \mathcal{P}(v) \vee \neg \mathcal{Q}(w)) \rightarrow (\neg \mathcal{P}(w) \vee \mathcal{Q}(v)) && \text{Paso 1} \\
 & \Leftrightarrow \neg(\neg \neg \mathcal{P}(v) \vee \neg \mathcal{Q}(w)) \vee (\neg \mathcal{P}(w) \vee \mathcal{Q}(v)) && \text{Paso 1} \\
 & \Leftrightarrow (\neg \neg \neg \mathcal{P}(v) \wedge \neg \neg \mathcal{Q}(w)) \vee (\neg \mathcal{P}(w) \vee \mathcal{Q}(v)) && \text{Paso 2} \\
 & \Leftrightarrow (\neg \mathcal{P}(v) \wedge \mathcal{Q}(w)) \vee (\neg \mathcal{P}(w) \vee \mathcal{Q}(v)) && \text{Paso 2} \\
 & \Leftrightarrow (\neg \mathcal{P}(v) \vee \neg \mathcal{P}(w) \vee \mathcal{Q}(v)) \wedge (\mathcal{Q}(w) \vee \neg \mathcal{P}(w) \vee \mathcal{Q}(v)) && \text{Paso 3}
 \end{aligned}$$

◆

Definición 1.25. Una fórmula $\mathcal{F}$ está en **forma normal prenexa** si tiene la forma:

$$(\Pi_1 v_1) \cdots (\Pi_n v_n) \mathcal{M}$$

donde las v_i son variables, los Π_i son cuantificadores ($\forall$ o $\exists$) y $\mathcal{M}$ es una fórmula que no contiene cuantificadores, es decir, $\mathcal{M}$ está compuesta solamente por átomos y conectivos lógicos.

La fórmula $\mathcal{M}$ es llamada la **matriz** de $\mathcal{F}$ y la expresión formada por los cuantificadores $(\Pi_1 v_1) \cdots (\Pi_n v_n)$ es llamada el **prefijo** de $\mathcal{F}$.

EJEMPLO 8. Las siguientes fórmulas están en forma normal prenexa:

- $(\forall v)(\exists w)(\mathcal{R}(v, w) \wedge \mathcal{Q}(v, w))$
- $(\forall v)(\exists w)(\exists x)(\neg \mathcal{P}(v, x) \rightarrow \mathcal{Q}(w))$
- $(\exists v)(\exists w)(\mathcal{Q}(v, w) \vee (\mathcal{P}(w) \wedge \mathcal{R}(v, w)))$
- $(\forall v)(\forall w)(\forall x)(\mathcal{R}(v) \leftarrow \neg \mathcal{Q}(w, x))$

◆

Definición 1.26. Una fórmula está en **forma normal conjuntiva prenexa (FNCP)** si es una **forma normal prenexa** cuya **matriz** está en **forma normal conjuntiva**.

EJEMPLO 9. Las siguientes fórmulas están en forma normal conjuntiva prenexa:

- $(\forall v)(\exists w)(\mathcal{R}(v, w) \wedge \mathcal{Q}(v, w))$
- $(\forall v)(\exists w)(\exists x)(\neg \mathcal{P}(v, x) \wedge \mathcal{Q}(w))$
- $(\exists v)(\exists w)(\mathcal{Q}(v, w) \wedge (\mathcal{P}(w) \vee \mathcal{R}(v, w)))$
- $(\forall v)(\forall w)(\forall x)(\mathcal{R}(v) \wedge \neg \mathcal{Q}(w, x))$



De manera similar a la prueba del Teorema 1 se realiza la prueba del siguiente teorema; lo que haremos será añadir las equivalencias lógicas correspondientes a la acción de los cuantificadores.

Teorema 2.

Toda fórmula puede ser transformada en una fórmula equivalente en *forma normal conjuntiva prenexa*.

Demostración.

Al igual que la prueba del Teorema 1, esta demostración se realiza de forma constructiva.

Sean $\mathcal{F}, \mathcal{G}, \mathcal{H}$ fórmulas y Π_1, Π_2 cuantificadores que representan a los usuales $\forall$ y $\exists$.

Algoritmo FNCP

Basta aplicar los siguientes pasos, que consisten en emplear equivalencias lógicas, tantas veces como sea posible:

Paso 1 Eliminar las ocurrencias de los conectivos lógicos $\rightarrow$ y $\leftrightarrow$ aplicando:

$$\begin{aligned}\mathcal{F} \leftrightarrow \mathcal{G} &\equiv (\mathcal{F} \rightarrow \mathcal{G}) \wedge (\mathcal{G} \rightarrow \mathcal{F}) \\ \mathcal{F} \rightarrow \mathcal{G} &\equiv \neg \mathcal{F} \vee \mathcal{G}\end{aligned}$$

Paso 2 Transformar la fórmula obtenida en el **Paso 1** aplicando las siguientes equivalencias:

$$\begin{aligned}\neg \neg \mathcal{F} &\equiv \mathcal{F} \\ \neg(\mathcal{F} \vee \mathcal{G}) &\equiv \neg \mathcal{F} \wedge \neg \mathcal{G} \\ \neg(\mathcal{F} \wedge \mathcal{G}) &\equiv \neg \mathcal{F} \vee \neg \mathcal{G} \\ \neg(\forall v)\mathcal{F}(v) &\equiv (\exists v)\neg \mathcal{F}(v) \\ \neg(\exists v)\mathcal{F}(v) &\equiv (\forall v)\neg \mathcal{F}(v)\end{aligned}$$

Paso 3 En la fórmula obtenida en el **Paso 2**, cambiar el nombre de variables iguales ligadas por diferentes cuantificadores.

$$\begin{aligned}(\Pi_1 v)\mathcal{F}(v) \vee (\Pi_2 v)\mathcal{G}(v) &\equiv (\Pi_1 v)\mathcal{F}(v) \vee (\Pi_2 w)\mathcal{G}(w) \\ (\Pi_1 v)\mathcal{F}(v) \wedge (\Pi_2 v)\mathcal{G}(v) &\equiv (\Pi_1 v)\mathcal{F}(v) \wedge (\Pi_2 w)\mathcal{G}(w)\end{aligned}$$

donde $\Pi_1 \neq \Pi_2$.

Paso 4 Mover los cuantificadores al lado izquierdo de la fórmula, aplicando las siguientes equivalencias lógicas a la fórmula obtenida en el **Paso 3**.

$$\begin{aligned}(\Pi_1 v)\mathcal{F}(v) \vee \mathcal{G} &\equiv (\Pi_1 v)(\mathcal{F}(v) \vee \mathcal{G}) \\ (\Pi_1 v)\mathcal{F}(v) \wedge \mathcal{G} &\equiv (\Pi_1 v)(\mathcal{F}(v) \wedge \mathcal{G}) \\ (\forall v)\mathcal{F}(v) \wedge (\forall v)\mathcal{G}(v) &\equiv (\forall v)(\mathcal{F}(v) \wedge \mathcal{G}(v)) \\ (\exists v)\mathcal{F}(v) \vee (\exists v)\mathcal{G}(v) &\equiv (\exists v)(\mathcal{F}(v) \vee \mathcal{G}(v)) \\ (\Pi_1 v)\mathcal{F}(v) \vee (\Pi_2 w)\mathcal{G}(w) &\equiv (\Pi_1 v)(\Pi_2 w)(\mathcal{F}(v) \vee \mathcal{G}(w)) \\ (\Pi_1 v)\mathcal{F}(v) \wedge (\Pi_2 w)\mathcal{G}(w) &\equiv (\Pi_1 v)(\Pi_2 w)(\mathcal{F}(v) \wedge \mathcal{G}(w))\end{aligned}$$

Paso 5 Hasta aquí, la fórmula obtenida en el **Paso 4** es una forma normal prenexa. Finalmente, sólo nos queda transformar la matriz, que para este paso no tiene cuantificadores, en forma normal conjuntiva aplicando las leyes distributivas, tantas veces como sea necesario.

$$\begin{aligned}\mathcal{F} \vee (\mathcal{G} \wedge \mathcal{H}) &\equiv (\mathcal{F} \vee \mathcal{G}) \wedge (\mathcal{F} \vee \mathcal{H}) \\ \mathcal{F} \wedge (\mathcal{G} \vee \mathcal{H}) &\equiv (\mathcal{F} \wedge \mathcal{G}) \vee (\mathcal{F} \wedge \mathcal{H})\end{aligned}$$

Debido a que en todo el proceso se empleo equivalencias lógicas, la fórmula resultante es equivalente lógicamente a la fórmula inicial. ■

EJEMPLO 10. Encontraremos la *forma normal conjuntiva prenexa* de la fórmula

$$(\forall v) (((\forall v)\mathcal{P}(v) \rightarrow \mathcal{Q}(w, x)) \wedge \neg(\forall x)(\exists v)\neg\mathcal{R}(f(v, x), x))$$

aplicando los pasos del *Algoritmo FNCP*, donde $\mathcal{P}, \mathcal{Q}, \mathcal{R}$ son símbolos de relación y f es un símbolo de función, veamos:

$$(\forall v) (((\forall v)\mathcal{P}(v) \rightarrow \mathcal{Q}(w, x)) \wedge \neg(\forall x)(\exists v)\neg\mathcal{R}(f(v, x), x))$$

$$\leftrightarrow (\forall v) ((\neg(\forall v)\mathcal{P}(v) \vee \mathcal{Q}(w, x)) \wedge \neg(\forall x)(\exists v)\neg\mathcal{R}(f(v, x), x)) \quad \text{Paso 1}$$

$$\leftrightarrow (\forall v) (((\exists v)\neg\mathcal{P}(v) \vee \mathcal{Q}(w, x)) \wedge (\exists x)\neg(\exists v)\neg\mathcal{R}(f(v, x), x)) \quad \text{Paso 2}$$

$$\leftrightarrow (\forall v) (((\exists v)\neg\mathcal{P}(v) \vee \mathcal{Q}(w, x)) \wedge (\exists x)(\forall v)\neg\mathcal{R}(f(v, x), x)) \quad \text{Paso 2}$$

$$\leftrightarrow (\forall v) (((\exists v)\neg\mathcal{P}(v) \vee \mathcal{Q}(w, x)) \wedge (\exists x)(\forall v)\mathcal{R}(f(v, x), x)) \quad \text{Paso 2}$$

$$\begin{aligned}
&\leftrightarrow (\forall v)((\exists y)\neg\mathcal{P}(y) \vee \mathcal{Q}(w, x)) \wedge (\exists z)(\forall v)\mathcal{R}(f(v, z), z) && \text{Paso 3} \\
&\leftrightarrow (\forall v)((\exists y)(\neg\mathcal{P}(y) \vee \mathcal{Q}(w, x)) \wedge (\exists z)(\forall v)\mathcal{R}(f(v, z), z)) && \text{Paso 4} \\
&\leftrightarrow (\forall v)(\exists y)((\neg\mathcal{P}(y) \vee \mathcal{Q}(w, x)) \wedge (\exists z)(\forall v)\mathcal{R}(f(v, z), z)) && \text{Paso 4} \\
&\leftrightarrow (\forall v)(\exists y)(\exists z)((\neg\mathcal{P}(y) \vee \mathcal{Q}(w, x)) \wedge (\forall v)\mathcal{R}(f(v, z), z)) && \text{Paso 4} \\
&\leftrightarrow (\forall v)(\exists y)(\exists z)(\forall v)((\neg\mathcal{P}(y) \vee \mathcal{Q}(w, x)) \wedge \mathcal{R}(f(v, z), z)) && \text{Paso 4} \\
&\leftrightarrow (\forall v)(\exists y)(\exists z)((\neg\mathcal{P}(y) \vee \mathcal{Q}(w, x)) \wedge \mathcal{R}(f(v, z), z)) && \text{Paso 4}
\end{aligned}$$

◆

Definición 1.27. Una fórmula en **forma normal de Skolem** es una *forma normal conjuntiva prenexa* cuyo *prefijo* no contiene cuantificadores existenciales, es decir, si presenta la forma:

$$(\forall_1 v_1) \cdots (\forall_n v_n) \mathcal{M}$$

EJEMPLO 11. De las fórmulas descritas en forma normal conjuntiva prenexa en el Ejemplo 9, solamente la última fórmula está en *forma normal de Skolem*. La fórmula escrita a continuación está en *forma normal de Skolem*:

$$(\forall v)(\forall y)(\forall z)((\neg\mathcal{P}(y) \vee \mathcal{Q}(w, x)) \wedge \mathcal{R}(f(v, z), z))$$

◆

El siguiente algoritmo permite transformar una fórmula en su forma normal conjuntiva prenexa en la *forma normal de Skolem*. Este procedimiento se conoce como **Skolemización** y lo presentamos de la forma expuesta por Ortíz[30].

Sea $(\Pi_1 v_1) \cdots (\Pi_n v_n) \mathcal{M}$ una fórmula $\mathcal{F}$ en su forma normal conjuntiva prenexa, donde $\Pi_1, \dots, \Pi_n$ son cuantificadores, $v_1, \dots, v_n$ son variables (distintas) y $\mathcal{M}$ una matriz en forma normal conjuntiva. El propósito es eliminar todos los cuantificadores existenciales presentes en la fórmula.

Skolemización

Sea Π_m , $1 \leq m \leq n$ un cuantificador existencial del prefijo de $\mathcal{F}$.

Caso 1 Si antes de Π_m no hay cuantificadores universales, se escoge un símbolo de constante, que no aparecía antes en la fórmula, se reemplaza por todas las ocurrencias de la variable v_m y se borra $(\Pi_m v_m)$ del prefijo. Por ejemplo, la fórmula

$$(\exists v)(\forall w)(\exists x)(\mathcal{P}(v, w) \vee \mathcal{Q}(v, x))$$

aplicando este caso quedaría

$$(\forall w)(\exists x)(\mathcal{P}(a, w) \vee \mathcal{Q}(a, x))$$

donde a representa una constante (que no aparecía en la matriz).

Caso 2 Sean $\Pi_1, \dots, \Pi_{m-1}$ (todos) los cuantificadores universales que aparecen a la izquierda de Π_m . Se sustituyen todas las ocurrencias de la variable v_m por un símbolo de función (que no aparezca en $\mathcal{M}$) con los argumentos $v_1, \dots, v_{m-1}$ y se elimina $(\Pi_m v_m)$ del prefijo. Por ejemplo, la fórmula

$$(\forall v)(\forall w)(\exists x)(\mathcal{P}(v, x) \vee \mathcal{Q}(w, x))$$

aplicando este caso quedaría

$$(\forall v)(\forall w)(\mathcal{P}(v, f(v, w)) \vee \mathcal{Q}(w, f(v, w)))$$

donde f representa una función binaria (que no aparecía en la matriz).

El proceso termina cuando se ha aplicado los dos casos, el que corresponda en cada iteración, hasta no quedar cuantificadores existenciales. Así, la fórmula obtenida finalmente está en *forma normal de Skolem*.

Los símbolos de función añadidos son llamados **funciones de Skolem**.

EJEMPLO 12. Para la forma normal conjuntiva prenexa obtenida en el desarrollo del ejemplo 10

$$(\forall v)(\exists y)(\exists z)((\neg \mathcal{P}(y) \vee \mathcal{Q}(w, x)) \wedge \mathcal{R}(f(v, z), z))$$

su forma normal de Skolem, aplicando el proceso de *Skolemización*, será:

$$(\forall v)((\neg \mathcal{P}(f(v))) \vee \mathcal{Q}(w, x)) \wedge \mathcal{R}(f(v, g(v)), g(v))$$

donde $f(v)$ y $g(v)$ reemplazan las variables y y z , ligadas a los cuantificadores existenciales, respectivamente. ◆

Debido a que los procedimientos descritos en los dos casos del proceso de Skolemización no son equivalencias lógicas, la forma calculada no es (en todos los casos) equivalente lógicamente a la fórmula inicial, es decir, pueden existir estructuras en las cuales la fórmula y su forma normal de Skolem tengan diferente valor de verdad. Observemos el siguiente ejemplo típico de esta situación.

EJEMPLO 13. Dada la fórmula $(\exists v)\mathcal{Q}(v)$, su forma normal de Skolem es $\mathcal{Q}(a)$, donde a es una constante. Consideremos la siguiente estructura $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{I} \rangle$ siendo $\mathcal{D} = \{1, 2\}$ y una interpretación $\mathfrak{I}$ definida por $a^{\mathfrak{I}} = 1$, $\mathcal{Q}^{\mathfrak{I}} = \{2\}$. Con base en las reglas de interpretación de la Definición 1.14, se tiene que $[(\exists v)\mathcal{Q}(v)]^{\mathfrak{I}} = 1$, ya que la asignación $\mathfrak{B}$ definida por $v^{\mathfrak{B}} = 2$ lo satisface, mientras que $[\mathcal{Q}(a)]^{\mathfrak{I}} = 0$ pues $a^{\mathfrak{I}} = 1$ y $1 \notin \mathcal{Q}^{\mathfrak{I}}$. ◆

Observación 7. *En el Ejemplo 13, tal como se enuncia en la Observación 6, el cálculo del valor de verdad de las fórmulas, no depende de la función de asignación, ya que ambas fórmulas son sentencias.*

En el Ejemplo 13 observamos que en general no se cumple que entre una fórmula y su forma normal de Skolem haya equivalencia lógica; sin embargo, se cumple una propiedad un poco más débil pero suficiente para cumplir con nuestro propósito en la construcción de la forma de deducir el Principio de Resolución.

Teorema 3 (Skolem).

Sea $\mathcal{F}_S$ la forma normal de Skolem de la fórmula $\mathcal{F}$. Entonces $\mathcal{F}_S$ es insatisfacible sí, y solo sí, $\mathcal{F}$ es insatisfacible.

La prueba de este teorema se encuentra en forma detallada en Ortíz[30, pág. 86] y dejamos al lector su consulta.

Definición 1.28. Un *cláusula* es una disyunción finita en la cual cada miembro es un literal.

Observación 8. *Con base en la definición anterior, una forma normal conjuntiva será entonces una conjunción de cláusulas.*

Recordemos que en caso de haber alguna variable libre en una fórmula dada, su interpretación está sujeta a la función de asignación, es decir, el valor de verdad de la fórmula dada es subjetivo a la asignación que se de a la variable en cuestión.

Por esta razón, de ahora en adelante y sin pérdida de generalidad, consideraremos análisis y aplicaciones sobre *sentencias* de la lógica de Primer Orden, es decir, sólo se consideran fórmulas que no contengan variables libres.

Ahora veremos de que manera una forma normal de Skolem hace su aporte más importante al desarrollo de nuestra teoría.

Consideremos una forma normal de Skolem

$$(\forall v_1) \dots (\forall v_n) \mathcal{M}.$$

Si $v_1, \dots, v_n$ son todas las variables presentes en la matriz $\mathcal{M}$, podemos prescindir de los cuantificadores, pues sabemos que cualquier variable que aparezca en $\mathcal{M}$

está cuantificada universalmente. Además, dado que $\mathcal{M}$ se encuentra en forma normal conjuntiva, $\mathcal{M}$ presenta la sintaxis

$$\mathcal{C}_1 \wedge \dots \wedge \mathcal{C}_n$$

donde cada $\mathcal{C}_i$, $i = 1, \dots, n$, es una cláusula. Podemos representar la forma de Skolem mediante el conjunto de cláusulas que la conforman, así:

$$\mathcal{S} = \{\mathcal{C}_1, \dots, \mathcal{C}_n\}.$$

A $\mathcal{S}$ lo llamaremos un **conjunto estándar de cláusulas**, o simplemente, **conjunto estándar**.

Toda esta representación tiene un propósito muy importante, y es el poder extender la aplicación del Teorema de Skolem, de modo que, un conjunto estándar $\mathcal{S} = \{\mathcal{C}_1, \dots, \mathcal{C}_n\}$ es *insatisfacible* sí, y solo sí, la fórmula $(\forall v_1) \dots (\forall v_n)(\mathcal{C}_1 \wedge \dots \wedge \mathcal{C}_n)$ es *insatisfacible*, donde $v_1, \dots, v_n$ son todas las variables que aparecen en las cláusulas $\mathcal{C}_i$, $i = 1, \dots, n$. Es así como nuestro interés se direcciona ahora por garantizar la insatisfacibilidad de un conjunto de cláusulas. Sin embargo, esto es algo que analizaremos más adelante.

Sea $\mathcal{I}_1 \vee \dots \vee \mathcal{I}_n$ una cláusula. Dado que $\vee$ es una operación binaria conmutativa, podemos ordenar la cláusula de tal forma que aparezcan en primer lugar los literales positivos, que denotamos con la letra $\mathcal{A}$, y después los literales negativos, que denotamos con la negación de la letra $\mathcal{B}$, de forma que:

$$\mathcal{I}_1 \vee \dots \vee \mathcal{I}_n = (\mathcal{A}_1 \vee \dots \vee \mathcal{A}_p) \vee (\neg \mathcal{B}_1 \vee \dots \vee \neg \mathcal{B}_q)$$

con $n = p + q$ y $\mathcal{A}_1, \dots, \mathcal{A}_p, \mathcal{B}_1, \dots, \mathcal{B}_q$ literales positivos. Ahora teniendo en cuenta la equivalencia lógica

$$(\neg \mathcal{B}_1 \vee \dots \vee \neg \mathcal{B}_q) = \neg (\mathcal{B}_1 \wedge \dots \wedge \mathcal{B}_q)$$

una cláusula se puede escribir como

$$(\mathcal{A}_1 \vee \dots \vee \mathcal{A}_p) \vee \neg (\mathcal{B}_1 \wedge \dots \wedge \mathcal{B}_q)$$

lo que equivale lógicamente a

$$(\mathcal{B}_1 \wedge \dots \wedge \mathcal{B}_q) \rightarrow (\mathcal{A}_1 \vee \dots \vee \mathcal{A}_p).$$

Luego, con el propósito de destacar lo que podemos decir de lo que ya se conoce, invertimos la notación de una cláusula en programación lógica, escribiendo ahora en primer lugar el consecuente y en segundo lugar el antecedente de la implicación, así:

$$\mathcal{A}_1, \dots, \mathcal{A}_p \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_q \quad \text{con } p, q \geq 0.$$

Observación 9. En la notación inmediatamente anterior, todos los literales que aparecen son positivos. Sin embargo, debemos recordar que los literales que se encuentran a la izquierda de $\leftarrow$ son los literales no-negados de la cláusula, mientras que los literales que se encuentran a la derecha de $\leftarrow$ son los literales negados de la cláusula.

Gráficamente podemos verlo como:

$$\begin{array}{ccc}
 \text{-----} & \leftarrow & \text{-----} \\
 \text{Literales} & & \text{Literales} \\
 \text{No - negados} & \leftarrow & \text{Negados} \\
 \text{(cláusula)} & \vdots & \text{(cláusula)} \\
 \text{-----} & \leftarrow & \text{-----}
 \end{array}$$

A la parte izquierda de $\leftarrow$ se le llama **cabeza**, y a la parte derecha de $\leftarrow$ se le llama **cuerpo**, de la cláusula. La coma que separa los literales en la cabeza representa el $\vee$ y la coma que separa los literales en el cuerpo representa el $\wedge$.

Definición 1.29. Las cláusulas se clasifican dependiendo del número de literales que tengan en la *cabeza* y/o en el *cuerpo*, y dependiendo del signo (positivo o negativo) de estos literales, veamos:

- (i) Una cláusula con un único literal se llama **cláusula unitaria**.
- (ii) Una cláusula que tiene un único literal *positivo*, adopta la expresión:

$$A \leftarrow$$

y se llama **cláusula unitaria positiva** o también **hecho** (ya que según el orden definido, sabemos que siempre se cumple).

- (iii) Una cláusula en la cual todos sus literales son átomos no negados, adopta la expresión:

$$A_1, \dots, A_p \leftarrow \quad \text{con } p \geq 1.$$

y se llama **cláusula positiva**.

- (iv) Una cláusula en la cual todos sus literales son átomos negados, adopta la expresión:

$$\leftarrow B_1, \dots, B_q \quad \text{con } q \geq 1.$$

y se llama **cláusula negativa** o también **objetivo**.

- (v) Una cláusula en la cual sus literales son átomos negados y no negados, adopta la expresión:

$$\mathcal{A}_1, \dots, \mathcal{A}_p \longleftarrow \mathcal{B}_1, \dots, \mathcal{B}_q \quad \text{con } p \geq 2, q \geq 1.$$

y se llama **cláusula mixta** o también **cláusula indefinida**.

- (vi) Una cláusula *mixta* con a lo más un literal positivo, adopta la expresión:

$$\mathcal{A} \longleftarrow \mathcal{B}_1, \dots, \mathcal{B}_q \quad \text{con } q \geq 0.$$

y se llama **cláusula definida** o también **cláusula de Horn**.

- (vii) Una cláusula sin literales, que aparece como resultado de un proceso de deducción, se denomina **cláusula vacía** y se representa por el símbolo $\square$.

Observación 10. *La cláusula vacía se interpreta en 0 (falso) para cualquier estructura, ya que una cláusula se interpreta en verdadero si existe al menos un literal que se interpreta en verdadero.*

Observación 11. *El tipo de cláusula exhibida en la Definición 1.29 ítem (vi), jugará un papel muy importante en el resto de este trabajo, ya que sobre estas cláusulas se definirá toda la teoría de programación lógica relacionada.*

Como se mencionó anteriormente, una forma normal de Skolem se puede representar mediante un *conjunto estándar* $\mathcal{S} = \{\mathcal{C}_1, \dots, \mathcal{C}_n\}$, donde cada $\mathcal{C}_i$, $i = 1, \dots, n$, es una cláusula. Si denotamos con $\mathcal{A}_{i,j}$ los literales positivos de $\mathcal{C}_i$ y con $\neg\mathcal{B}_{i,k}$ los literales negativos de $\mathcal{C}_i$, $j = 1, \dots, p$ y $k = 1, \dots, q$ con $p, q \geq 0$, usando la notación empleada en programación lógica, tendremos una cláusula *indefinida* como esta:

$$\mathcal{C}_i : \quad \mathcal{A}_{i,1}, \dots, \mathcal{A}_{i,p} \longleftarrow \mathcal{B}_{i,1}, \dots, \mathcal{B}_{i,q}$$

Lo que haremos será expresar cada una de las cláusulas de $\mathcal{S}$ de manera independiente. A la expresión obtenida por la reunión de estas cláusulas la llamaremos **programa lógico** (ver Gráfica 1.2), y lo denotamos por $\mathfrak{P}$.

$$\begin{array}{lll} \mathcal{C}_1 : & \mathcal{A}_{1,1}, \dots, \mathcal{A}_{1,p} \longleftarrow & \mathcal{B}_{1,1}, \dots, \mathcal{B}_{1,q} \\ \mathcal{C}_2 : & \mathcal{A}_{2,1}, \dots, \mathcal{A}_{2,p} \longleftarrow & \mathcal{B}_{2,1}, \dots, \mathcal{B}_{2,q} \\ & \vdots & \vdots \\ \mathcal{C}_n : & \mathcal{A}_{n,1}, \dots, \mathcal{A}_{n,p} \longleftarrow & \mathcal{B}_{n,1}, \dots, \mathcal{B}_{n,q} \end{array}$$

Gráfica 1.2. Programa lógico

Definición 1.30. Los *programas lógicos* adoptan la siguiente clasificación, de acuerdo con el tipo de cláusulas que lo componen:

- (i) Un programa lógico **definido** (o programa **Horn**) es un conjunto finito de cláusulas *definidas* (o cláusulas *Horn*) (Ver. Gráfica 1.3).
- (ii) Un programa lógico **indefinido** es un conjunto finito de cláusulas *indefinidas* (o cláusulas *mixtas*).

$$\begin{array}{lll}
 \mathcal{C}_1 : & \mathcal{A}_1 \longleftarrow & \mathcal{B}_{1,1}, \dots, \mathcal{B}_{1,q} \\
 \mathcal{C}_2 : & \mathcal{A}_2 \longleftarrow & \mathcal{B}_{2,1}, \dots, \mathcal{B}_{2,q} \\
 & \vdots & \vdots \\
 \mathcal{C}_n : & \mathcal{A}_n \longleftarrow & \mathcal{B}_{n,1}, \dots, \mathcal{B}_{n,q}
 \end{array}$$

Gráfica 1.3. Programa lógico definido

Observación 12. *Note que los objetivos (o cláusulas negativas) no hacen parte de los dos tipos de programas lógicos definidos anteriormente.*

En la siguiente sección, desarrollaremos la teoría de una aplicación sobre el conjunto de las variables, que representa un rol importante en este trabajo, debido a que su forma de operar estará relacionado con la forma de obtener “respuestas” en el método deductivo de nuestro interés.

1.4. Sustitución

Como lo enuncia Celma[2], ”... es importante destacar, que desde el punto de vista de la demostración de teoremas, el mayor interés reside precisamente en demostrar consecuencias lógicas, sin embargo, desde el punto de vista de la programación estamos interesados en los reemplazos que se realizan para las variables de la fórmula en estudio, ya que estos reemplazos representan precisamente la salida del programa lógico...”.

Para realizar los reemplazos que nos interesan, se empleará una aplicación que permita cambiar las variables libres de una fórmula de Primer Orden por elementos definidos de un lenguaje (de Primer Orden) como son los términos. En esta sección se formaliza su definición y sus propiedades, las cuales son un eje central para el desarrollo de la resolución en lógica de primer orden.

Sean $\mathcal{L}(\mathcal{C}, \mathcal{F}, \mathcal{R})$ un lenguaje de Primer Orden, $\mathcal{V}$ el conjunto de variables de $\mathcal{L}(\mathcal{C}, \mathcal{F}, \mathcal{R})$ y $\mathcal{T}$ el conjunto de términos de $\mathcal{L}(\mathcal{C}, \mathcal{F}, \mathcal{R})$.

Definición 1.31. Una *sustitución* es una función $\sigma : \mathcal{V} \rightarrow \mathcal{T}$ y se representa por el conjunto finito de parejas

$$\{v_1 |_{t_1}, \dots, v_n |_{t_n}\}$$

donde cada $v_i (\in \mathcal{V})$ es una variable diferente y cada $t_i (\in \mathcal{T}) \neq v_i, i = 1, \dots, n$.

La sustitución que deja las variables idénticas, es decir, que no modifica ninguna variable de cualquier expresión, la llamaremos *sustitución identidad* y se denota por $\epsilon = \{ \}$.

Observación 13. Debido a que las sustituciones son funciones, emplearemos el orden de la notación convencional $\sigma[v]$ para denotar el resultado de aplicar la función σ a v , en lugar del algebraico $v\sigma$ que emplean muchos autores, Fitting[5], Ben-Ari[1], Celma[2], Ortíz[30], entre otros.

Aunque $\mathcal{V}$ es un conjunto numerable de variables, las sustituciones son aplicadas a un número finito de ellas, por lo tanto, cada sustitución σ se aplica como una *sustitución identidad* para las variables restantes que no se encuentren definidas en σ misma.

Cabe notar que nos interesan aquellas sustituciones que para $v \in \mathcal{V}$ cualquiera, apliquen a un término distinto, es decir, que $\sigma[v] \neq v$. Al conjunto que reúne las variables que cumplen esta característica para una sustitución dada, se le conoce como *soporte* y se encuentra definido en Fitting[5, pág. 101].

Tal como se dijo en la Definición 1.31, las sustituciones son funciones definidas sobre variables pero su acción, gracias al Principio de Recursión, es fácilmente extendida para todos los términos.

Definición 1.32. Sea σ una sustitución. Entonces se define:

- $\sigma[c] = c$, para una constante c .
- $\sigma[f(t_1, \dots, t_n)] = f(\sigma[t_1], \dots, \sigma[t_n])$, para un símbolo de función n -ario.

Definición 1.33. Una *expresión* $\mathcal{E}$ es un término, un literal, una cláusula o un conjunto de cláusulas.

Definición 1.34. Sea $\mathcal{E}$ una expresión y sea $\sigma = \{v_1|_{t_1}, \dots, v_n|_{t_n}\}$ una sustitución. Una **instancia** $\sigma[\mathcal{E}]$ de $\mathcal{E}$ es obtenida por el reemplazo simultáneo de cada ocurrencia de v_i en $\mathcal{E}$ por t_i , $i = 1, \dots, n$.

Observación 14. Si $\sigma[\mathcal{E}]$ es básica, es decir, que no tiene variables, entonces $\sigma[\mathcal{E}]$ es llamada una **instancia básica** de $\mathcal{E}$.

EJEMPLO 14. Sean a, b constantes, v, w, x variables y g, f funciones 1-aria y 2-aria, respectivamente. Sea la expresión $\mathcal{Q}(g(f(v, w)), x, g(a))$ y la sustitución $\sigma = \{v|_a, w|_{f(b, w)}, x|_{g(a)}\}$, entonces su instancia será:

$$\begin{aligned}\sigma[\mathcal{Q}(g(f(v, w)), x, g(a))] &= \mathcal{Q}(g(f(\sigma[v], \sigma[w])), \sigma[x], g(\sigma[a])) \\ &= \mathcal{Q}(g(f(a, f(b, w))), g(b), g(a)).\end{aligned}$$

◆

Observación 15. Si $\mathcal{G}$ es un conjunto (numerable) de expresiones y σ es una sustitución, denotamos la instancia de $\mathcal{G}$ por σ como $\sigma[\mathcal{G}]$; esto describe un nuevo conjunto formado por las instancias de cada una de las expresiones que lo conforman.

Definición 1.35. Sean σ y ρ sustituciones, representadas por los conjuntos (finitos)

$$\sigma = \{v_1|_{t_1}, \dots, v_n|_{t_n}\} \quad \text{y} \quad \rho = \{w_1|_{s_1}, \dots, w_m|_{s_m}\}.$$

Se define la **composición** $\sigma\rho$ de σ y ρ como una nueva sustitución, tal que para cada expresión $\mathcal{E}$ se cumple que $\sigma\rho[\mathcal{E}] = \sigma[\rho[\mathcal{E}]]$, y se representa por el conjunto

$$\sigma\rho = \{v_1|_{t_1}, \dots, v_n|_{t_n}, w_1|_{\sigma[s_1]}, \dots, w_m|_{\sigma[s_m]}\}$$

eliminando cualquier par $w_i|_{\sigma[s_i]}$ tal que $w_i = \sigma[s_i]$ ($1 \leq i \leq m$), y cualquier par $v_j|_{t_j}$ tal que $v_j \in \{w_1, \dots, w_m\}$, ($1 \leq j \leq n$).

Observación 16. En otras palabras, para σ y ρ tal como en la Definición 1.35, su composición $\sigma\rho$, consiste en aplicar la sustitución σ a los términos s_i de ρ , con la condición que se elimine cada par resultante $w_i|_{\sigma[s_i]}$ tal que $w_i = \sigma[s_i]$ y se adicionan los pares $v_j|_{t_j}$ de σ tales que $v_j \neq w_i$, para todo i, j .

EJEMPLO 15. Sean a, b constantes, u, v, x, y, z variables y f, g funciones unarias. Sean las sustituciones σ, ρ, γ representadas por

$$\sigma = \{y|_{g(a)}, u|_z, v|_{f(f(a))}\} \quad , \quad \rho = \{x|_{f(y)}, y|_{f(a)}\} \quad , \quad \gamma = \{z|_b\}$$

y la expresión $\mathcal{E} = \mathcal{Q}(u, v, x, y, z)$.

Entonces, la acción de las sustituciones de manera independiente es:

$$\begin{aligned} \rho[\mathcal{E}] &= \mathcal{Q}(u, v, \rho[x], \rho[y], z) \\ &= \mathcal{Q}(u, v, f(y), f(a), z). \\ \sigma[\rho[\mathcal{E}]] &= \mathcal{Q}(\sigma[u], \sigma[v], \sigma[f(y)], f(a), z) \\ &= \mathcal{Q}(z, f(f(a)), f(g(a)), f(a), z). \\ \gamma[\sigma[\rho[\mathcal{E}]]] &= \mathcal{Q}(\gamma[z], f(f(a)), f(g(a)), f(a), \gamma[z]) \\ &= \mathcal{Q}(b, f(f(a)), f(g(a)), f(a), b). \end{aligned}$$

Ahora, calculemos la *composición* $\gamma\sigma\rho$ y su instancia sobre $\mathcal{E}$,

$$\begin{aligned} \gamma\sigma\rho &= \gamma(\sigma\rho) & (*) \\ &= \gamma \left(\left\{ \overset{\text{eliminar}}{y|_{g(a)}}, u|_z, v|_{f(f(a))}, x|_{\sigma[f(y)]}, y|_{\sigma[f(a)]} \right\} \right) \\ &= \gamma \left(\{u|_z, v|_{f(f(a))}, x|_{f(g(a))}, y|_{f(a)}\} \right) \\ &= \{z|_b, u|_{\gamma[z]}, v|_{f(f(a))}, x|_{f(g(a))}, y|_{f(a)}\} \\ &= \{z|_b, u|_b, v|_{f(f(a))}, x|_{f(g(a))}, y|_{f(a)}\} \\ \gamma\sigma\rho[\mathcal{E}] &= \mathcal{Q}(\gamma\sigma\rho[u], \gamma\sigma\rho[v], \gamma\sigma\rho[x], \gamma\sigma\rho[y], \gamma\sigma\rho[z]) \\ &= \mathcal{Q}(b, f(f(a)), f(g(a)), f(a), b). \end{aligned}$$

◆

Observación 17. En el Ejemplo 15 se marco con $(*)$ la primer igualdad del cálculo de la composición $\gamma\sigma\rho$ para destacar el uso de la propiedad asociativa. En general, se cumple la propiedad asociativa para la composición de sustituciones. Probaremos la propiedad para la composición de tres sustituciones, haciendo evidente la trivialidad de la prueba, veamos:

Sean γ, σ y ρ sustituciones representadas por

$$\gamma = \{x_1|_{r_1}, \dots, x_m|_{r_m}\} \quad , \quad \sigma = \{y_1|_{s_1}, \dots, y_n|_{s_n}\} \quad y \quad \rho = \{z_1|_{t_1}, \dots, z_p|_{t_p}\}.$$

Con base en la Definición 1.35, tenemos que:

$$\begin{aligned}
 \gamma(\sigma\rho) &= \gamma\left(\{y_1|_{s_1}, \dots, y_n|_{s_n}, z_1|_{\sigma[t_1]}, \dots, z_p|_{\sigma[t_p]}\}\right), y_i \neq z_j \text{ y } z_j \neq \sigma[t_j], \\
 &\quad \forall i, j, i \in \{1, \dots, n\}, j \in \{1, \dots, p\}. \\
 &= \{x_1|_{r_1}, \dots, x_m|_{r_m}, y_1|_{\gamma[s_1]}, \dots, y_n|_{\gamma[s_n]}, z_1|_{\gamma[\sigma[t_1]]}, \dots, z_p|_{\gamma[\sigma[t_p]]}\}, \\
 &\quad x_k \neq y_i \neq z_j \text{ y } y_i \neq \gamma[s_i], z_j \neq \gamma[\sigma[t_j]], \forall i, j, k, k \in \{1, \dots, m\}. \\
 &= \{x_1|_{r_1}, \dots, x_m|_{r_m}, y_1|_{\gamma[s_1]}, \dots, y_n|_{\gamma[s_n]}\} \cup \{z_1|_{\gamma[\sigma[t_1]]}, \dots, z_p|_{\gamma[\sigma[t_p]]}\}, \\
 &\quad x_k \neq y_i \neq z_j \text{ y } y_i \neq \gamma[s_i], z_j \neq \gamma[\sigma[t_j]], \forall i, j, k. \\
 &= (\gamma\sigma)\{z_1|_{\gamma[\sigma[t_1]]}, \dots, z_p|_{\gamma[\sigma[t_p]]}\}, x_k \neq y_i \neq z_j \text{ y } z_j \neq \gamma[\sigma[t_j]], \forall i, j, k. \quad (*) \\
 &= (\gamma\sigma)\rho \quad (**).
 \end{aligned}$$

Finalmente, observe que en (**) la composición $(\gamma\sigma)$ representa una función (sustitución), mientras que en (*) representa un conjunto, donde las variables (intrínsecamente) son x_k y y_i , $k \in \{1, \dots, m\}$ e $i \in \{1, \dots, n\}$.

Observación 18. La composición de sustituciones no cumple la propiedad conmutativa, veamos un ejemplo sencillo:

Sean γ y ρ sustituciones representadas por $\gamma = \{v|_{f(w)}\}$ y $\rho = \{w|_a\}$. Con base en la Definición 1.35, tenemos que:

$$\gamma\rho = \{v|_{f(w)}, w|_{\gamma[a]}\} = \{v|_{f(w)}, w|_a\} \neq \{w|_a, v|_{f(a)}\} = \{w|_a, v|_{\rho[f(w)]}\} = \rho\gamma$$

Definición 1.36. Sean σ y ρ sustituciones. Diremos que σ es **más general** que ρ si, para alguna sustitución γ , se tiene que $\rho = \gamma\sigma$.

EJEMPLO 16. Para las sustituciones σ y ρ representadas por

$$\rho = \{v|_{f(g(a, h(z)))}, w|_{g(h(v), b)}, z|_{h(v)}\} \text{ y } \sigma = \{v|_{f(g(v, w))}, w|_{g(z, b)}\}$$

tenemos que σ es **más general** que ρ , ya que $\rho = \gamma\sigma$, donde γ puede ser:

$$\gamma = \{v|_a, w|_{h(z)}, z|_{h(v)}\}.$$

◆

Observación 19. Note que toda sustitución σ es más general que ella misma, puesto que $\sigma = \epsilon\sigma$, donde ϵ es la sustitución identidad.

La siguiente definición juega un rol muy importante en la programación lógica, haciendo parte de las reglas del Principio de Resolución que estudiaremos; debido a que ayuda a exhibir concretamente las respuestas de un sistema deductivo de un programa lógico.

Definición 1.37. Sean $\mathcal{E}_1$ y $\mathcal{E}_2$ expresiones. Una sustitución σ es un **unificador** de $(\mathcal{E}_1, \mathcal{E}_2)$ sí, y solo sí, sus instancias son iguales, es decir, $\sigma(\mathcal{E}_1) = \sigma(\mathcal{E}_2)$; en tal caso $\mathcal{E}_1$ y $\mathcal{E}_2$ se dicen **unificables**.

EJEMPLO 17. Sean las expresiones $\mathcal{E}_1 = \mathcal{R}(f(v), g(w))$ y $\mathcal{E}_2 = \mathcal{R}(f(f(a)), g(x))$, donde a es constante, v, w, x variables y f, g funciones 1-arias. Observemos, en la Gráfica 1.4, tres tipos (diferentes) de unificadores σ :

UNIFICADOR σ	INSTANCIAS	
	$\sigma[\mathcal{R}(f(v), g(w))]$	$\sigma[\mathcal{R}(f(f(a)), g(x))]$
$\{v _{f(a)}, w _{f(g(a))}, x _{f(g(a))}\}$	$\mathcal{R}(f(f(a)), g(f(g(a))))$	$\mathcal{R}(f(f(a)), g(f(g(a))))$
$\{v _{f(a)}, w _a, x _a\}$	$\mathcal{R}(f(f(a)), g(a))$	$\mathcal{R}(f(f(a)), g(a))$
$\{v _{f(a)}, w _x\}$	$\mathcal{R}(f(f(a)), g(x))$	$\mathcal{R}(f(f(a)), g(x))$

Gráfica 1.4. Unificación (Ejemplo 17)



Para dos expresiones puede no existir unificador. Tal caso se debe a que ó el símbolo de relación no es el mismo ó el símbolo de relación coincide pero no tienen el mismo número de argumentos ó una variable de una expresión está incluida en otro término dentro de la otra expresión.

EJEMPLO 18. Las siguientes expresiones no son unificables:

- $\mathcal{P}(v, w, a)$ y $\mathcal{Q}(v, w, a)$, puesto que difieren del símbolo de relación.
- $\mathcal{Q}(v, w)$ y $\mathcal{Q}(v, w, x)$, puesto que difieren del número de argumentos.
- $\mathcal{Q}(f(v))$ y $\mathcal{Q}(g(w))$, puesto que difieren de símbolo de función.
- v y $f(v)$, puesto que no existe sustitución que pueda igualarlos simultáneamente.



Tal como se aprecia en la Gráfica 1.4, pueden existir varios unificadores para un conjunto de expresiones. Debido a que cada sustitución puede realizar cambios sobre algunas o todas las variables (diferentes), deseamos hallar aquellas sustituciones más generales que lo hagan solo para las variables que se requieran inmediatamente; para esto se define una característica muy importante denotada como *umg*.

Definición 1.38. Un unificador σ de $(\mathcal{E}_1, \mathcal{E}_2)$ es un ***unificador más general***, denotado por ***umg***, si σ es *más general* que cualquier otro unificador de $(\mathcal{E}_1, \mathcal{E}_2)$, es decir, si para cualquier otro unificador ρ de $(\mathcal{E}_1, \mathcal{E}_2)$ existe una sustitución γ tal que $\rho = \gamma\sigma$.

La existencia del *unificador más general* está garantizada por el algoritmo implementado por Robinson para tal fin; su prueba es constructiva, ya que exhibe el método para encontrarlo. Lo emplearemos tal como se exhibe en Ortíz[29, pág. 109] y Ben-Ari[1, pág. 158].

EJEMPLO 19. Sean $\mathcal{E}_1 = \mathcal{R}(f(v), g(w))$ y $\mathcal{E}_2 = \mathcal{R}(f(f(a)), g(x))$. Basados en el algoritmo de existencia para el *umg*, tenemos que el unificador más general de $(\mathcal{E}_1, \mathcal{E}_2)$ es $\sigma = \{v|_{f(a)}, w|_x\}$. Además, si $\sigma_1 = \{v|_{f(a)}, w|_{f(g(a))}, x|_{f(g(a))}\}$ y $\sigma_2 = \{v|_{f(a)}, w|_a, x|_a\}$, de la Gráfica 1.4 del Ejemplo 17, sabemos que σ_1 y σ_2 son *unificadores* de $(\mathcal{E}_1, \mathcal{E}_2)$, pero no son *umg*'s, ya que, $\sigma_1 = \gamma_1\sigma$, donde $\gamma_1 = \{x|_{f(g(a))}\}$ y $\sigma_2 = \gamma_2\sigma$, donde $\gamma_2 = \{x|_a\}$.

◆

1.5. Resolución

Con el desarrollo teórico en la Lógica de Primer Orden presentado hasta este momento, sintaxis, semántica, formas normales y sustitución, daremos paso al estudio de mayor interés, el *Principio de Resolución*. Hemos dejado a un lado el desarrollo logrado en la Lógica Proposicional solamente por el hecho de que, lo logrado en Primer Orden es una extensión enriquecida de lo Proposicional. Si se desea conocer más del Principio de Resolución en Lógica Proposicional ver Ben-Ari[1, pág. 64-81] o Fitting[5, pág. 45-49].

Partiendo de fórmulas de Primer Orden, hemos construido formas que permiten una manipulación lógica, las cláusulas, e interpretado su valor de verdad bajo ciertas estructuras, dando por definido el concepto de insatisfacibilidad. Luego, se encontró el modo de unificar los términos de las cláusulas y establecer así una relación sintáctica, por medio de las sustituciones.

Ahora, nuestro objetivo es describir el método que nos permita realizar deducciones lógicas a partir de un conjunto de cláusulas.

El *Principio de Resolución* es un método de deducción desarrollado por Robinson a mediados de los 60, que se aplica a las cláusulas y consiste en añadir la negación de una cláusula (objetivo) a un conjunto de cláusulas predefinidas conocidas como programa lógico, y realizar inferencias hasta obtener una contradicción (representada por la cláusula vacía), o equivalentemente, probar que el programa lógico junto a su objetivo es insatisfacible; este proceso es llamado refutación y será el pilar de nuestro método deductivo.

El objetivo de esta sección es realizar una definición formal de toda la estructura empleada en el *Principio de Resolución* de la Lógica de Primer Orden.

Sean S un conjunto (estándar) de cláusulas y Γ, Δ cláusulas (posiblemente vacías).

El ***Principio de Resolución*** consiste en la aplicación de las siguientes reglas sobre S .

Definición 1.39 (Regla de Resolución Binaria).

Sean C_1 y C_2 dos cláusulas que no tienen variables en común. Sean $l_1 \in C_1$ y $l_2 \in C_2$ dos literales, tales que l_1 y l_2 son unificables por un *umg* γ , entonces a través de la **regla de resolución binaria** se puede inferir la cláusula C_3 . Los literales l_1 y l_2 no aparecen en C_3 y se dice que son los *literales resueltos*.

$$\begin{array}{lcl} C_1 : & \Gamma_1 \vee l_1 \vee \Delta_1 \\ C_2 : & \Gamma_2 \vee \neg l_2 \vee \Delta_2 \\ \hline C_3 : & \gamma [\Gamma_1 \vee \Delta_1 \vee \Gamma_2 \vee \Delta_2] \end{array}$$

C_3 es llamada *resolvente binaria* de C_1 y C_2 .

Observación 20. Si C_3 es una resolvente binaria de C_1 y C_2 , llamaremos a las cláusulas C_1 y C_2 las **cláusulas padres**.

Observación 21. Si las cláusulas C_1 y C_2 tiene variables en común, antes de aplicar la regla de resolución binaria se deben renombrar las variables comunes en una de las dos cláusulas. Esto es posible gracias a que las variables son implícitamente cuantificadas universalmente.

EJEMPLO 20. Sean a, b constantes, u, v, w variables, f un símbolo de función unaria y $\mathcal{P}, \mathcal{Q}$ símbolos de relación 2-aria y 3-aria, respectivamente. Sean las cláusulas

$$\begin{aligned}\mathcal{C}_1 : & \quad \mathcal{P}(u, a) \vee \mathcal{Q}(u, v, w) \\ \mathcal{C}_2 : & \quad \mathcal{P}(b, v) \vee \neg \mathcal{P}(f(v), w) \vee \mathcal{Q}(u, v, w).\end{aligned}$$

Para hallar una resolvente binaria de $\mathcal{C}_1$ y $\mathcal{C}_2$, debemos antes de aplicar la regla, renombrar las variables u, v, w aplicando la sustitución $\{^u|_x, ^v|_y, ^w|_z\}$ a $\mathcal{C}_2$, obteniendo las cláusulas:

$$\begin{aligned}\mathcal{C}_1 : & \quad \mathcal{P}(u, a) \vee \mathcal{Q}(u, v, w) \\ \mathcal{C}_2 : & \quad \mathcal{P}(b, y) \vee \neg \mathcal{P}(f(y), z) \vee \mathcal{Q}(x, y, z).\end{aligned}$$

y como un *umg* de $\mathcal{P}(u, a)$ y $\neg \mathcal{P}(f(y), z)$ es $\gamma = \{^u|_{f(y)}, ^z|_a\}$ entonces, una resolvente binaria de $\mathcal{C}_1$ y $\mathcal{C}_2$ es:

$$\mathcal{C}_3 : \quad \gamma [\mathcal{Q}(u, v, w) \vee \mathcal{P}(b, y) \vee \mathcal{Q}(x, y, z)]$$

lo que equivale a, $\mathcal{C}_3 : \quad \mathcal{Q}(f(y), v, w) \vee \mathcal{P}(b, y) \vee \mathcal{Q}(x, y, a).$

◆

Observación 22. *El método enunciado como “renombrar variables” es conocido como estandarización (ver, Ben-Ari[pág. 160][1]) y como vimos en el ejemplo anterior, se emplea para evitar sustituciones inconsistentes.*

Observe que para hallar una resolvente binaria de dos cláusulas escritas de la forma $\mathcal{I}_1, \dots, \mathcal{I}_p \leftarrow \mathfrak{E}_1, \dots, \mathfrak{E}_q$ no se pueden tomar como literales resueltos dos literales del mismo signo que estén ambos o en la cabeza o en el cuerpo de las cláusulas. Por tanto, se tienen que tomar como literales resueltos dos del mismo signo, uno en la cabeza de una cláusula y otro en el cuerpo de la otra cláusula, o dos de distinto signo ambos de las cabezas o ambos de los cuerpos de cada cláusula. Así, dadas dos cláusulas:

$$\begin{aligned}\mathcal{A}_1, \dots, \mathcal{A}_p & \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_q \\ \mathcal{C}_1, \dots, \mathcal{C}_m & \leftarrow \mathcal{D}_1, \dots, \mathcal{D}_n\end{aligned}$$

que no tienen variables en común, si γ es un *umg* de $\mathcal{A}_i$ y $\mathcal{D}_j$, una resolvente binaria de ellas es:

$$\gamma[\mathcal{A}_1, \dots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \dots, \mathcal{A}_p, \mathcal{C}_1, \dots, \mathcal{C}_m \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_q, \mathcal{D}_1, \dots, \mathcal{D}_{j-1}, \mathcal{D}_{j+1}, \dots, \mathcal{D}_n].$$

Definición 1.40 (Regla de Factorización).

Dada una cláusula $\mathcal{C}$ con dos literales l_1 y l_2 unificables, a través de la **regla de factorización** se puede inferir $\mathcal{C}^*$, que se denomina un *factor* de $\mathcal{C}$.

$$\frac{\mathcal{C} : \quad \Gamma_1 \vee l_1 \vee \Gamma_2 \vee l_2 \vee \Gamma_3}{\mathcal{C}^* : \quad \gamma [\Gamma_1 \vee l_1 \vee \Gamma_2 \vee \Gamma_3]}$$

donde $\mu\gamma$ es el *umg* de l_1 y l_2 .

EJEMPLO 21. Con base en el Ejemplo 20, la resolvente binaria obtenida fue

$$\mathcal{C} : \quad \mathcal{Q}(f(y), v, w) \vee \mathcal{P}(b, y) \vee \mathcal{Q}(x, y, a).$$

Dado que un *umg* de $\mathcal{Q}(f(y), v, w)$ y $\mathcal{Q}(x, y, a)$ es $\gamma = \{x|_{f(y)}, y|_v, w|_a\}$ entonces, un factor de $\mathcal{C}$ es:

$$\mathcal{C}^* : \quad \mathcal{Q}(f(y), v, a) \vee \mathcal{P}(b, v).$$

◆

Observe que para hallar un factor de una cláusula escrita de la forma

$$l_1, \dots, l_p \leftarrow \mathfrak{l}_1, \dots, \mathfrak{l}_q$$

hay que unificar dos literales del mismo signo de la cabeza o del cuerpo, o dos literales de signo distinto uno en la cabeza y otro en el cuerpo de la cláusula.

Definición 1.41 (Regla de Resolución).

Dadas dos cláusulas $\mathcal{C}_1$ y $\mathcal{C}_2$, llamadas *cláusulas padres*, la aplicación de la **regla de resolución** sobre estas cláusulas consiste en uno de los cuatro casos siguientes:

- Resolución binaria entre $\mathcal{C}_1$ y $\mathcal{C}_2$.
- Resolución binaria entre $\mathcal{C}_1$ y un factor de $\mathcal{C}_2$.
- Resolución binaria entre $\mathcal{C}_2$ y un factor de $\mathcal{C}_1$
- Resolución binaria entre un factor de $\mathcal{C}_1$ y un factor de $\mathcal{C}_2$.

El resultado es llamado una *resolvente* de $\mathcal{C}_1$ y $\mathcal{C}_2$.

Definición 1.42. Sea $\mathcal{S}$ un conjunto de cláusulas. Una **derivación** de $\mathcal{C}$ a partir de $\mathcal{S}$, es una secuencia $\mathcal{C}_1, \dots, \mathcal{C}_k$ tal que $\mathcal{C}_k = \mathcal{C}$ y cada $\mathcal{C}_i$, $1 \leq i \leq k$, es bien ó una cláusula de $\mathcal{S}$ ó es una resolvente de cláusulas anteriores a $\mathcal{C}_i$.

Definición 1.43. Una **refutación** de $\mathcal{S}$ es una *derivación* de $\mathcal{C}$ a partir de $\mathcal{S}$, tal que $\mathcal{C}$ es una cláusula vacía, $\square$.

EJEMPLO 22. Sean a una constante, v, w variables, $\mathcal{N}, \mathcal{P}, \mathcal{Q}, \mathcal{T}$ y $\mathcal{R}$ símbolos de relación unarios y binario, respectivamente, y f un símbolo de función unario. Dado el conjunto de cláusulas $\mathcal{S}$ formado por:

$$\mathcal{S} = \left\{ \begin{array}{l} \bullet \neg\mathcal{P}(v) \vee \mathcal{Q}(v) \vee \mathcal{R}(v, f(v)) \\ \bullet \neg\mathcal{P}(v) \vee \mathcal{Q}(v) \vee \mathcal{N}(f(v)) \\ \bullet \mathcal{T}(a) \\ \bullet \mathcal{P}(a) \\ \bullet \neg\mathcal{R}(a, w) \vee \mathcal{T}(w) \\ \bullet \neg\mathcal{T}(v) \vee \neg\mathcal{Q}(v) \\ \bullet \neg\mathcal{T}(v) \vee \neg\mathcal{N}(v) \end{array} \right.$$

Una *refutación* de $\mathcal{S}$ por el *Principio de Resolución* se representa en la Gráfica 1.5.



Teorema 4 (Validez de la Resolución de Primer Orden).

Sean $\mathcal{S}$ un conjunto de cláusulas y $\mathcal{C}$ una cláusula, si de $\mathcal{S}$ se deriva $\mathcal{C}$ usando Resolución, entonces $\mathcal{S} \models \mathcal{C}$.

Teorema 5 (Completez de la Resolución de Primer Orden).

Un conjunto de cláusulas $\mathcal{S}$ es insatisfacible sí, y solo sí, existe una *refutación* de $\mathcal{S}$.

Las demostraciones de los dos teoremas anteriores se encuentran en Ben-Ari[pág. 164-166][1] y Ortíz[pág. 114-120][30], y se realizan de manera análoga a la prueba en Lógica Proposicional, empleando un tema llamado *árboles semánticos*, no tratado en este trabajo; por esta razón transferimos el lector a la referencia.

En conclusión, podemos decir que:

“El Principio de Resolución es un procedimiento que verifica la validez de una fórmula por refutación, es decir, prueba que la negación de una fórmula es insatisfacible.”

CLÁUSULAS PADRES	<i>umg</i>	RESOLVENTE
$\mathcal{C}_1 : \neg \mathcal{T}(v) \vee \neg \mathcal{Q}(v)$ $\mathcal{C}_2 : \mathcal{T}(a)$	$\{v _a\}$	$\mathcal{C}_3 : \neg \mathcal{Q}(a)$
$\mathcal{C}_4 : \neg \mathcal{P}(v) \vee \mathcal{Q}(v) \vee \mathcal{N}(f(v))$ $\mathcal{C}_5 : \mathcal{P}(a)$	$\{v _a\}$	$\mathcal{C}_6 : \mathcal{Q}(a) \vee \mathcal{N}(f(a))$
$\mathcal{C}_6 : \mathcal{Q}(a) \vee \mathcal{N}(f(a))$ $\mathcal{C}_3 : \neg \mathcal{Q}(a)$	$\{ \}$	$\mathcal{C}_7 : \mathcal{N}(f(a))$
$\mathcal{C}_8 : \neg \mathcal{P}(v) \vee \mathcal{Q}(v) \vee \mathcal{R}(v, f(v))$ $\mathcal{C}_5 : \mathcal{P}(a)$	$\{v _a\}$	$\mathcal{C}_9 : \mathcal{Q}(a) \vee \mathcal{R}(a, f(a))$
$\mathcal{C}_9 : \mathcal{Q}(a) \vee \mathcal{R}(a, f(a))$ $\mathcal{C}_3 : \neg \mathcal{Q}(a)$	$\{ \}$	$\mathcal{C}_{10} : \mathcal{R}(a, f(a))$
$\mathcal{C}_{11} : \neg \mathcal{R}(a, w) \vee \mathcal{T}(w)$ $\mathcal{C}_{10} : \mathcal{R}(a, f(a))$	$\{w _{f(a)}\}$	$\mathcal{C}_{12} : \mathcal{T}(f(a))$
$\mathcal{C}_{13} : \neg \mathcal{T}(v) \vee \neg \mathcal{N}(v)$ $\mathcal{C}_{12} : \mathcal{T}(f(a))$	$\{v _{f(a)}\}$	$\mathcal{C}_{14} : \neg \mathcal{N}(f(a))$
$\mathcal{C}_{14} : \neg \mathcal{N}(f(a))$ $\mathcal{C}_7 : \mathcal{N}(f(a))$	$\{ \}$	$\mathcal{C}_{15} : \square$

Gráfica 1.5. Refutación (Ejemplo 22)

1.6. Resolución lineal

Por muchos años, desde su aparición, el Principio de Resolución se ha considerado el eje para todos los métodos deductivos, por lo menos para los más utilizados, ya que han podido emplear su algoritmo de manera eficiente.

Sin embargo, como lo enuncia Ortíz[29], “... cuando el número de cláusulas crece, el número de pasos para llegar a la derivación de la cláusula vacía puede crecer mucho...”. Por esta razón, es necesario aplicar *criterios selectivos* que simplifiquen el proceso y lo hagan mucho más eficiente.

Un conjunto de criterios selectivos articulados de forma sistemática constituye una estrategia de resolución.

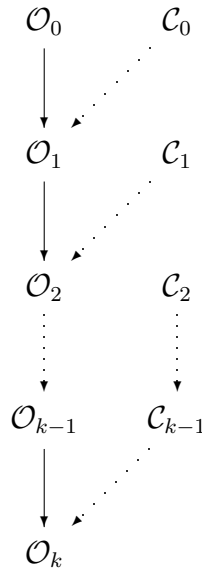
Estudiaremos una estrategia de “*refinamiento*” llamada **resolución lineal**. Es-

ta estrategia fue propuesta por Loveland (1970) y Luckham (1970) de manera independiente. Es exitosamente implementable y consiste en tomar una cláusula central (o de cabecera) y resolverla con otra cláusula, luego la cláusula resuelta se resuelve con otra cláusula y se aplica reiteradamente este proceso.

Definición 1.44. Sea $\mathcal{S}$ un conjunto de cláusulas y $\mathcal{O}_0$ una cláusula en $\mathcal{S}$. Una **derivación lineal** de $\mathcal{O}$ a partir de $\mathcal{S}$, con cláusula de cabecera $\mathcal{O}_0$, es una secuencia $\mathcal{O}_0, \mathcal{O}_1, \dots, \mathcal{O}_k$ tal que $\mathcal{O}_k = \mathcal{O}$ y, para cada $i = 1, \dots, k$, $\mathcal{O}_i$ es el resolvente de $\mathcal{C}_{i-1}$ y $\mathcal{O}_{i-1}$, donde $\mathcal{C}_{i-1}$ es bien ó una cláusula de $\mathcal{S}$ ó $\mathcal{C}_{i-1} = \mathcal{O}_j$ para algún $j \leq i - 1$.

Las cláusulas $\mathcal{O}_0, \mathcal{O}_1, \dots, \mathcal{O}_k$ son llamadas cláusulas **centrales** y $\mathcal{C}_0, \dots, \mathcal{C}_{k-1}$ son llamadas cláusulas **laterales**.

Una *derivación lineal* la representaremos de la forma expuesta en la Gráfica 1.6.



Gráfica 1.6. Derivación lineal

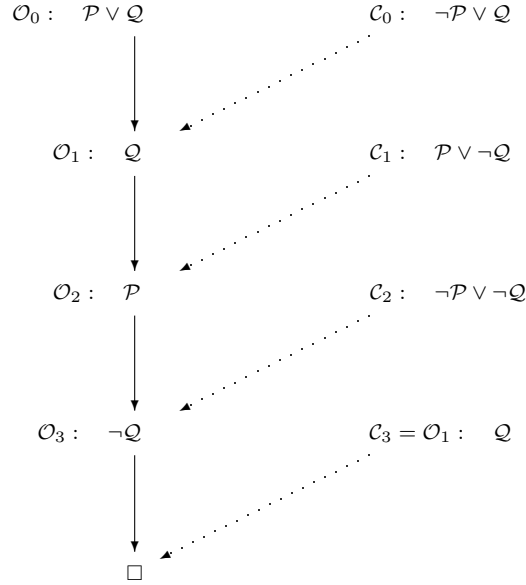
Definición 1.45. Una **refutación lineal** de $\mathcal{S}$ es una *derivación lineal* de $\mathcal{O}$ a partir de $\mathcal{S}$, tal que $\mathcal{O}$ es una cláusula vacía, $\square$.

EJEMPLO 23. Dado el conjunto de cláusulas

$$\mathcal{S} = \left\{ \begin{array}{ll} \bullet \mathcal{P} \vee \mathcal{Q} & \bullet \mathcal{P} \vee \neg \mathcal{Q} \\ \bullet \neg \mathcal{P} \vee \mathcal{Q} & \bullet \neg \mathcal{P} \vee \neg \mathcal{Q} \end{array} \right.$$

donde $\mathcal{P}$ y $\mathcal{Q}$ representan símbolos de relación de igual aridad y aplicados sobre una misma constante (arbitraria) para permitir la unificación entre símbolos.

Una *Refutación Lineal* de $\mathcal{S}$ con cláusula de cabecera $\mathcal{O}_0 : \mathcal{P} \vee \mathcal{Q}$, está representada en la Gráfica 1.7.



Gráfica 1.7. Refutación lineal (Ejemplo 23)

EJEMPLO 24. Sean a, b, c constantes, u, x, y, z variables, $\mathcal{P}, \mathcal{Q}, \mathcal{R}$ símbolos de relación 1-ario, 3-ario, 2-ario, respectivamente, y f un símbolo de función 1-ario.

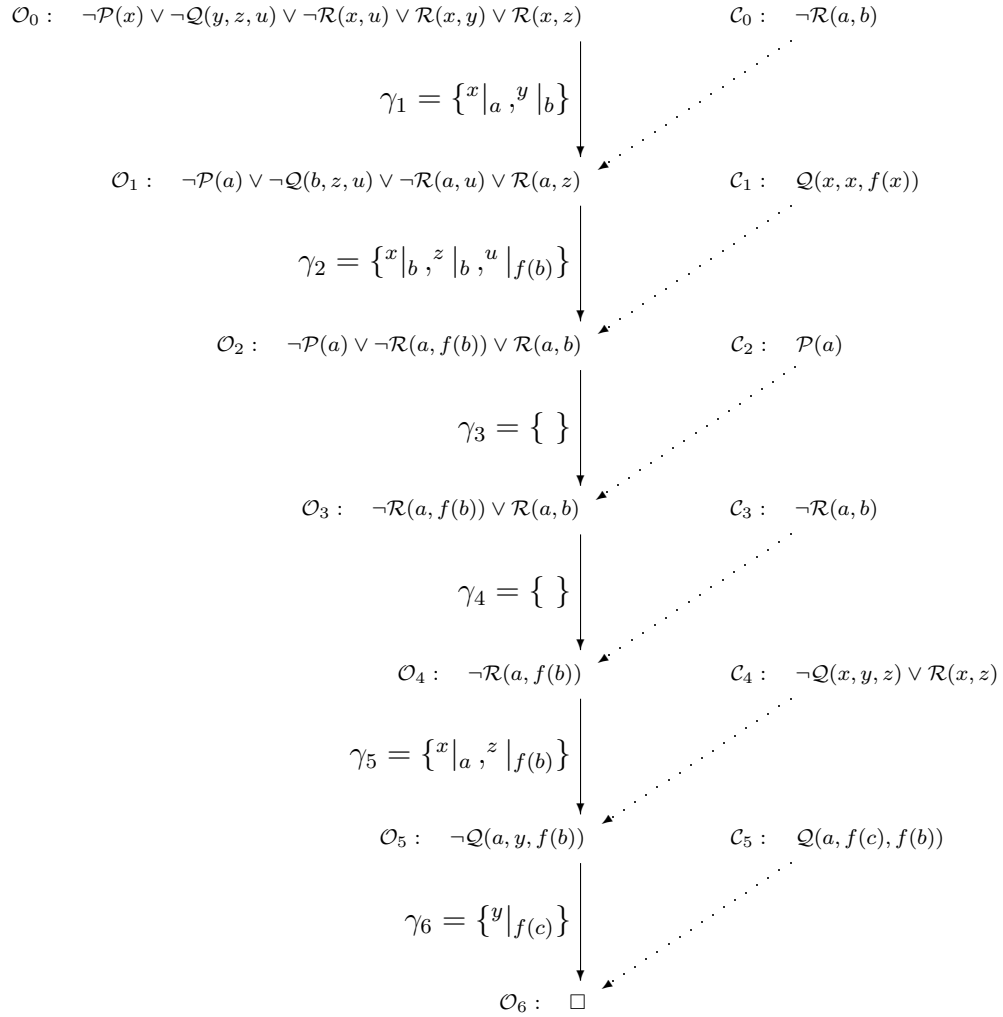
Sea $\mathcal{S}$ el conjunto de cláusulas,

$$\mathcal{S} = \left\{ \begin{array}{l} \bullet \neg \mathcal{P}(x) \vee \neg \mathcal{Q}(y, z, u) \vee \neg \mathcal{R}(x, u) \vee \mathcal{R}(x, y) \vee \mathcal{R}(x, z) \\ \bullet \neg \mathcal{R}(a, b) \\ \bullet \mathcal{Q}(x, x, f(x)) \\ \bullet \neg \mathcal{Q}(x, y, z) \vee \mathcal{Q}(x, y, z) \\ \bullet \neg \mathcal{Q}(x, y, z) \vee \mathcal{R}(x, z) \\ \bullet \mathcal{P}(a) \\ \bullet \mathcal{Q}(a, f(c), f(b)) \end{array} \right.$$

Una *Refutación Lineal* de $\mathcal{S}$ con cláusula de cabecera

$$\mathcal{O}_0 : \neg \mathcal{P}(x) \vee \neg \mathcal{Q}(y, z, u) \vee \neg \mathcal{R}(x, u) \vee \mathcal{R}(x, y) \vee \mathcal{R}(x, z)$$

está representada en la Gráfica 1.8.



Gráfica 1.8. Refutación lineal (Ejemplo 24)

En la sección anterior vimos que el Principio de Resolución es un método deductivo válido y completo. Una de las mayores ventajas de una estrategia de refinamiento como la *Resolución Lineal*, es que prevalece la completitud del Principio de Resolución.

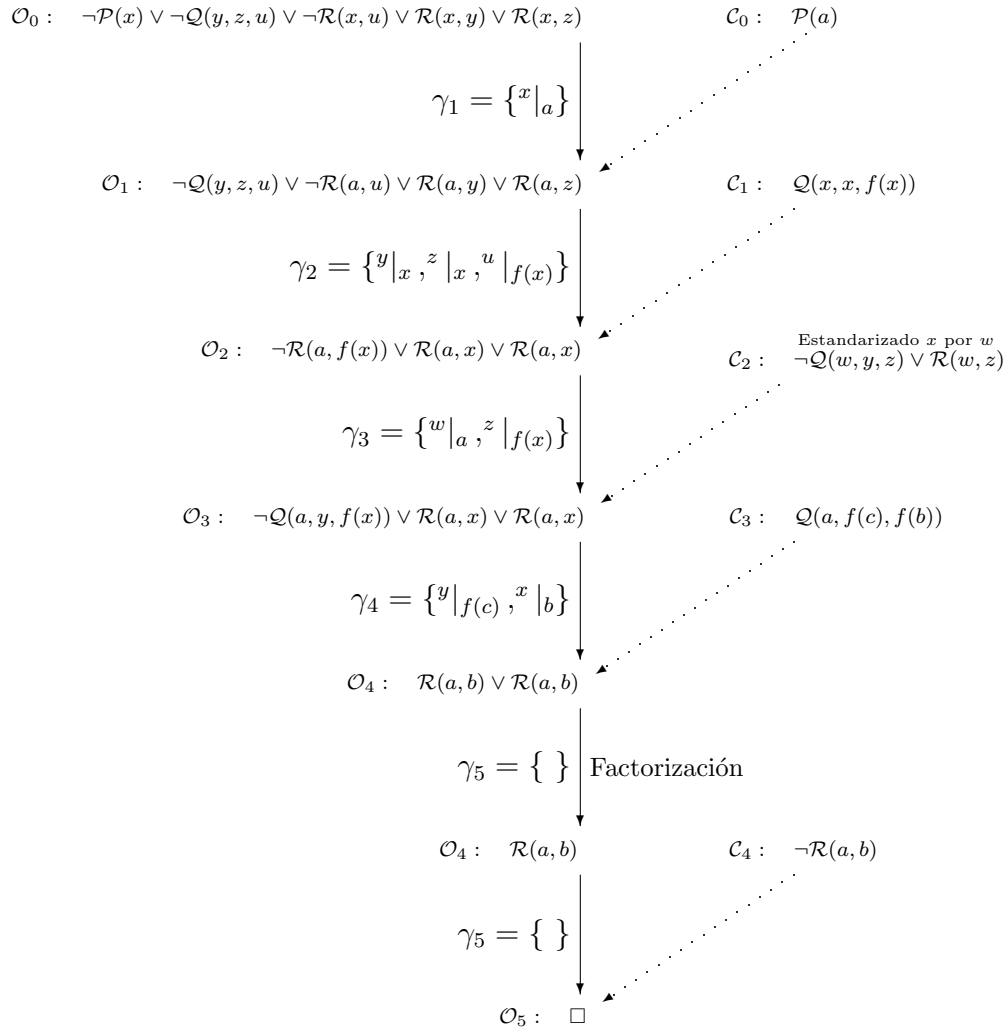
La prueba del siguiente teorema se encuentra en Ortíz[pág. 134-137][30] y la omitimos por su extensión.

Teorema 6 (Completez de la Resolución Lineal).

Si $\mathcal{S}$ es un conjunto insatisfacible de cláusulas y $\mathcal{C}$ es una cláusula de $\mathcal{S}$ tal que $\mathcal{S} - \{\mathcal{C}\}$ es satisfacible, entonces existe una *refutación lineal* de $\mathcal{S}$ con cláusula de cabecera $\mathcal{C}$.

La estrategia de resolución lineal puede combinarse con un criterio de ***selección de literal***. Este criterio elimina alternativas de resolución y puede significar una disminución considerable en los procesos de búsqueda, ya que para llegar a la derivación de la cláusula vacía es necesario ir eliminando cada literal de la cláusula central, entonces podemos seleccionar primero cuál es el literal que se quiere resolver, por medio de una ***regla de computación***, y luego considerar sólo las cláusulas laterales que lo permiten, por medio de una ***regla de búsqueda***. Ambas reglas, *de computación* y *de búsqueda* se definen en el capítulo siguiente.

EJEMPLO 25. Decidimos como un criterio de *selección de literal*, que los literales de cada cláusula se van resolver de izquierda a derecha; con base en el Ejemplo 24 obtenemos la siguiente *refutación lineal* representada en la Gráfica 1.9.



Gráfica 1.9. Refutación lineal con selección de literal (Ejemplo 25)

Hasta aquí, hemos definido todos los conceptos previos necesarios para el desarrollo de nuestro objetivo.

Definimos una característica estructural en el Principio de Resolución, llamada *Resolución lineal* y la enriquecimos adicionando una estrategia de refinamiento como lo es el criterio de *selección de literal*.

En el capítulo siguiente, unificamos todo este avance teórico en un método deductivo llamado *Resolución-SLD*, que consiste en una estrategia de *Resolución Lineal* junto con un criterio de *selección de literal*, aplicado a *cláusulas definidas* (o *cláusula de Horn*, tal como se definieron en [1.29](#)).

La sigla SLD para la Resolución de nuestro interés viene del inglés, **S**elective **L**inear for **D**efinite clauses.

Capítulo 2

Programación lógica

Aunque este capítulo debería llamarse *Resolución-SLD*, por que es aquí donde describimos esta importante estrategia de resolución, nuestro tema central de estudio; fue nombrado *Programación lógica* para destacar el fuerte vínculo existente entre estos dos temas, debido a que los métodos de deducción automática a partir del Principio de Resolución se han convertido en el eje operativo de los lenguajes de programación lógica y han logrado por medio de este, el mayor alcance en su desarrollo.

Green en 1969 realizó un aporte destacado en la historia de los métodos deductivos. Él se percató de que al aplicar la unificación se iba generando restricciones sobre los valores de las variables que, al llegar a la cláusula vacía, constituían una solución al problema y así dio origen a la programación lógica, por eso se considera su precursor (ver Green[12]). Todo este trabajo, alentó a Kowalski y otro grupo de investigadores a implementar un lenguaje de programación; y entre aquel grupo de investigadores se encontraba Colmerauer, el cual con sus avances investigativos dio inicio a una variedad de programas lógicos, que se han desarrollado rápidamente en muy poco tiempo, entre ellos “PROLOG”.

PROLOG (PROgramming in LOGic) es un lenguaje basado en la Lógica de Primer Orden, que trabaja sobre *cláusulas de Horn* (cláusulas definidas) e incorpora la *resolución-SLD* como su método de deducción. Es el lenguaje de programación lógica que más se ha desarrollado, ya que ha contado con extensivos esfuerzos por su implementación y lo han convertido en una herramienta práctica para el avance de la programación lógica.

Este capítulo lo dividimos en cuatro secciones en las cuales trataremos los siguientes temas: la Resolución-SLD, sus métodos de derivación (semántica operacional), su interpretación (semántica declarativa) y su punto de contacto entre la lógica y el álgebra (semántica de punto fijo).

2.1. Resolución-SLD

En esta sección vamos a abordar la aplicación de una estrategia de refinamiento del Principio de Resolución, conocida como *Resolución-SLD*, que es una especialización de la Resolución Lineal, diseñada para trabajar con programas lógicos definidos.

Tal como se dijo en la Definición 1.29, una cláusula la podemos expresar en la forma:

$$\mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_q \quad \text{con } q \geq 0,$$

donde todos los literales que aparecen son positivos, y fue nombrada *cláusula definida* (o cláusula de Horn). Usaremos la siguiente clasificación para las *cláusulas definidas*, dependiendo de la cantidad de literales en la cabeza y/o el cuerpo de la cláusula, veamos:

$\mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p$, con $p \geq 1$	: reglas .
$\leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p$, con $p \geq 1$	: <i>objetivos</i> (item (iv) Def. 1.29).
$\mathcal{A} \leftarrow$	: <i>hechos</i> (item (ii) Def. 1.29).
$\square$	: <i>cláusula vacía</i> (item (vii) Def. 1.29).

Sea $\mathfrak{P}$ un programa lógico definido, formado por *reglas* y *hechos*. Un *objetivo* según su definición adopta la expresión general $\leftarrow \mathfrak{l}_1, \dots, \mathfrak{l}_q$ con $q \geq 1$, lo que representa a la sentencia $(\forall v_1) \dots (\forall v_n) \neg(\mathfrak{l}_1 \wedge \dots \wedge \mathfrak{l}_q)$, donde $v_1, \dots, v_n$ son todas las variables de la fórmula; la cual es equivalente lógicamente a la fórmula

$$\neg((\exists v_1) \dots (\exists v_n)(\mathfrak{l}_1 \wedge \dots \wedge \mathfrak{l}_q)).$$

Es por esta razón, que cuando se quiere demostrar que

$$\mathfrak{P} \models (\exists v_1) \dots (\exists v_n)(\mathfrak{l}_1 \wedge \dots \wedge \mathfrak{l}_q)$$

es decir, que $(\exists v_1) \dots (\exists v_n)(\mathfrak{l}_1 \wedge \dots \wedge \mathfrak{l}_q)$ es una consecuencia lógica de $\mathfrak{P}$, lo que se hace, según la Proposición 1.2, es añadir su negación a las cláusulas de $\mathfrak{P}$ y demostrar que

$$\mathfrak{P} \cup \{\neg(\exists v_1) \dots (\exists v_n)(\mathfrak{l}_1 \wedge \dots \wedge \mathfrak{l}_q)\} \equiv \mathfrak{P} \cup \{\leftarrow \mathfrak{l}_1, \dots, \mathfrak{l}_q\}$$

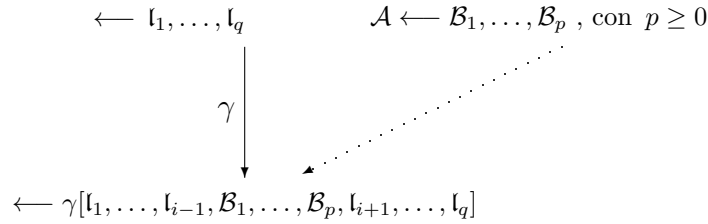
es insatisfacible.

Ahora, suponiendo que $\mathfrak{P} \cup \{\leftarrow \mathfrak{l}_1, \dots, \mathfrak{l}_q\}$ es insatisfacible y que $\mathfrak{P}$ es satisfacible entonces, según el Teorema 6, existe una *refutación lineal* de $\mathfrak{P}$ con cláusula de cabecera $\leftarrow \mathfrak{l}_1, \dots, \mathfrak{l}_q$.

Si tomamos como cláusula padre a nuestro objetivo $\leftarrow l_1, \dots, l_q$, las únicas cláusulas $\mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p$, con $p \geq 0$ que pueden actuar como laterales en $\mathfrak{P}$, son aquellas que la cabeza $\mathcal{A}$ es unificable con algún literal l_i , ($i = 1, \dots, q$) del objetivo. Así, siendo γ el *umg* de l_i y $\mathcal{A}$, obtenemos como resolvente (binaria):

$$\leftarrow \gamma[l_1, \dots, l_{i-1}, \mathcal{B}_1, \dots, \mathcal{B}_p, l_{i+1}, \dots, l_q].$$

Representado gráficamente tendremos algo como:



De esta forma, el proceso de Resolución Lineal se simplifica notablemente, ya que siempre manejamos objetivos y, para resolver, tenemos que buscar unificaciones sólo entre sus literales y las cabezas de las reglas o de los hechos.

A continuación, se definen algunos elementos necesarios para la teoría operacional de la Resolución-SLD. En adelante, $\mathfrak{P}$ denota un programa lógico definido y $\mathcal{O}$ un objetivo.

Definición 2.1. Una *regla de computación* es un criterio de selección de literales en un objetivo.

Definición 2.2. Una *regla de búsqueda* es una regla que selecciona una cláusula del programa lógico para resolverla con el literal seleccionado en un objetivo.

Como se mencionó antes, las sustituciones juegan un papel determinante en el objetivo de los programas lógicos, ya que por medio de ellas se exhiben los valores constantes que reemplazan las variables y satisfacen la fórmula.

Definición 2.3. Una *respuesta* θ para $\mathfrak{P} \cup \{\mathcal{O}\}$, es una sustitución de las variables de $\mathcal{O}$.

Definición 2.4. Una respuesta θ para $\mathfrak{P} \cup \{\mathcal{O}\}$, es una *respuesta correcta* para $\mathfrak{P} \cup \{\mathcal{O}\}$ sí, y solo sí, $\mathfrak{P} \models \theta[\mathcal{O}]$, es decir, si $\theta[\mathcal{O}]$ es una consecuencia lógica de $\mathfrak{P}$.

EJEMPLO 26. Sea $\mathfrak{P}$ el siguiente programa lógico definido:

$$\begin{aligned} \mathcal{C}_1 : \quad & \mathcal{Q}(u, v) \leftarrow \mathcal{R}(v) \\ \mathcal{C}_2 : \quad & \mathcal{R}(c) \leftarrow \end{aligned}$$

donde u, v, w son variables y c, d constantes. Algunas *respuestas correctas* para los objetivos $\mathcal{O}$ dados son:

- Si $\mathcal{O} : \leftarrow \mathcal{Q}(w, w)$ entonces $\theta = \{^w|_c\}$ es una *respuesta correcta*, ya que $\mathfrak{P} \models \mathcal{Q}(c, c)$, o lo que es equivalente, $\mathfrak{P} \models (\exists w)\mathcal{Q}(w, w)$.
- Una *respuesta correcta* no produce necesariamente una instancia básica, por ejemplo, si $\mathcal{O} : \leftarrow \mathcal{Q}(x, w)$ entonces $\theta = \{^x|_u, ^w|_c\}$ es una *respuesta correcta*, ya que $\mathfrak{P} \models \mathcal{Q}(u, c)$, o equivalentemente que $\mathfrak{P} \models (\forall x)(\exists w)\mathcal{Q}(x, w)$.
- Si $\mathcal{O} : \leftarrow \mathcal{R}(c)$ entonces $\theta = \epsilon$ (la sustitución identidad) es una *respuesta correcta*, ya que $\mathfrak{P} \models \mathcal{R}(c)$.
- Si $\mathcal{O} : \leftarrow \mathcal{R}(d)$ entonces, para este objetivo, no existe una *respuesta correcta*, ya que $\mathfrak{P} \cup \{\leftarrow \mathcal{R}(d)\} \equiv \mathfrak{P} \cup \{\neg \mathcal{R}(d)\}$ es satisfacible, es decir (según la Proposición 1.2), $\mathfrak{P} \not\models \mathcal{R}(d)$.

◆

Definición 2.5 (Resolución-SLD).

Sean $\mathcal{O} : \leftarrow \mathfrak{l}_1, \dots, \mathfrak{l}_p$ un objetivo (definido), con $p \geq 1$, y $\mathcal{C}_1, \dots, \mathcal{C}_n$ cláusulas definidas de la forma:

$$\begin{aligned} \mathcal{C}_1 : \quad & \mathcal{A}_1 \leftarrow \mathcal{B}_{1,1}, \dots, \mathcal{B}_{1,q_1} \\ \mathcal{C}_2 : \quad & \mathcal{A}_2 \leftarrow \mathcal{B}_{2,1}, \dots, \mathcal{B}_{2,q_2} \\ & \vdots \qquad \qquad \qquad \vdots \\ \mathcal{C}_n : \quad & \mathcal{A}_n \leftarrow \mathcal{B}_{n,1}, \dots, \mathcal{B}_{n,q_n} \end{aligned}$$

donde $\mathcal{O}$ y $\mathcal{C}_i$, $1 \leq i \leq n$, no tienen variables en común y si las tienen se *estandarizan*.

$\mathcal{O}^*$ es *derivado por Resolución-SLD* de $\mathcal{O}$ y $\mathcal{C}_1, \dots, \mathcal{C}_n$ si se determina por, la selección de un literal $\mathfrak{l}_m \in \mathcal{O}$, de acuerdo a la *regla de computación*, y por la selección de una cláusula $\mathcal{C}_i$, $1 \leq i \leq n$, de acuerdo a la *regla de búsqueda*; tal que θ sea un *umg* de $\mathcal{A}_i$ (la cabeza de $\mathcal{C}_i$) y $\mathfrak{l}_m$ (el literal seleccionado). Entonces, el objetivo

$$\mathcal{O}^* : \quad \theta[\leftarrow \mathfrak{l}_1, \dots, \mathfrak{l}_{m-1}, \mathcal{B}_{i,1}, \dots, \mathcal{B}_{i,q_i}, \mathfrak{l}_{m+1}, \dots, \mathfrak{l}_p]$$

es la resolvente binaria de $\mathcal{O}$ y $\mathcal{C}_i$.

Algoritmo SLD

El primer objetivo $\mathcal{O}_0$ es $\mathcal{O}$. Supongamos que $\mathcal{O}_j$ ha sido *derivado*.

$\mathcal{O}_{j+1}$ se deriva de $\mathcal{O}_j$ y $\mathcal{C}_i$, usando el *umg* θ_j , si se cumplen estas condiciones:

- $\mathfrak{l}_m$ es un literal de $\mathcal{O}_j$ seleccionado por una *regla de computación*,
- $\mathcal{C}_i$ es una cláusula definida seleccionada por una *regla de búsqueda*,
- θ_j es el *umg* de $\mathfrak{l}_m$ y $\mathcal{A}_i$ (la cabeza de $\mathcal{C}_i$),

y resolviendo:

$$\begin{aligned} \mathcal{O}_i : & \quad \leftarrow \mathfrak{l}_1, \dots, \mathfrak{l}_{m-1}, \mathfrak{l}_m, \mathfrak{l}_{m+1}, \dots, \mathfrak{l}_p \\ \mathcal{C}_i : & \quad \mathcal{A}_i \leftarrow \mathcal{B}_{i,1}, \dots, \mathcal{B}_{i,q_i} \\ & \quad \theta_j[\mathfrak{l}_m] = \theta_j[\mathcal{A}_i] \\ \mathcal{O}_{i+1} : & \quad \theta_j[\leftarrow \mathfrak{l}_1, \dots, \mathfrak{l}_{m-1}, \mathcal{B}_{i,1}, \dots, \mathcal{B}_{i,q_i}, \mathfrak{l}_{m+1}, \dots, \mathfrak{l}_p] \end{aligned}$$

2.2. Semántica operacional para la programación lógica

La semántica operacional define un programa lógico en términos de los resultados elaborados por la ejecución del algoritmo base del programa lógico mismo; en este caso, el *algoritmo-SLD*.

En esta sección se caracteriza la manera en que opera la *Resolución-SLD*, esto por medio de la derivación, y consideramos un caso particular llamado *refutación-SLD*; este último juega un rol muy importante en nuestro trabajo, ya que bajo la teoría desarrollada, nuestras deducciones se obtendrán a partir de procesos de contradicción.

Definición 2.6. Sea $\mathfrak{P}$ un programa lógico y $\mathcal{O}$ un objetivo. Una ***derivación-SLD*** de $\mathfrak{P} \cup \{\mathcal{O}\}$ esta formada por:

- una sucesión de objetivos $\mathcal{O}_0, \mathcal{O}_1, \mathcal{O}_2, \dots$,
- una sucesión de cláusulas $\mathcal{C}_1, \mathcal{C}_2, \dots$ de $\mathfrak{P}$ y,
- una sucesión $\theta_1, \theta_2, \dots$ de *umg*'s,

tal que $\mathcal{O} = \mathcal{O}_0$ y cada $\mathcal{O}_{i+1}$, $i \geq 0$, se *deriva* de $\mathcal{O}_i$ y $\mathcal{C}_{i+1}$ utilizando θ_{i+1} .

Observación 23. Cada par $\mathcal{O}_i$ y $\mathcal{C}_{i+1}$, no debe tener variables en común, en caso de tenerlas se deben estandarizar.

Observación 24. La sucesión de cláusulas esta formada por las cláusulas empleadas en cada paso de la derivación-*SLD*, es decir, la derivación-*SLD* se caracteriza únicamente por las cláusulas del programa lógico que se unifiquen con el objetivo (respectivo) de cada paso.

Definición 2.7. Una derivación-*SLD* de $\mathfrak{P} \cup \{\mathcal{O}\}$ puede ser de dos tipos:

- **Infinita:** si la derivación-*SLD* no finaliza, debido a que en cualquiera de los objetivos $\mathcal{O}_0, \mathcal{O}_1, \mathcal{O}_2, \dots$ el literal seleccionado se unifica únicamente con reglas de $\mathfrak{P}$.
- **Finita:** si la derivación finaliza en un número finito de pasos, es decir, las sucesiones de objetivos y *umg*'s son finitas. Si el número de pasos es m , es decir, si el último objetivo de la derivación-*SLD* es $\mathcal{O}_m$, diremos que la derivación-*SLD* tiene **longitud m** . La finalización de una derivación-*SLD finita* de longitud m puede ser de dos formas:
 - * **Fracaso:** si el átomo seleccionado en $\mathcal{O}_m$ (último objetivo), no es unificable con alguna cláusula de $\mathfrak{P}$.
 - * **Éxito:** si el último objetivo, $\mathcal{O}_m = \square$.

EJEMPLO 27. Sean $\mathfrak{P}$ un programa lógico formado por:

$$\mathfrak{P} = \begin{cases} \mathcal{C}_1 : & \mathcal{Q}(u, v) \leftarrow \mathcal{P}(u, v) \\ \mathcal{C}_2 : & \mathcal{Q}(u, v) \leftarrow \mathcal{P}(u, w), \mathcal{Q}(w, v) \\ \mathcal{C}_3 : & \mathcal{P}(a, a) \\ \mathcal{C}_4 : & \mathcal{P}(a, b) \end{cases}$$

y $\mathcal{O} : \leftarrow \mathcal{Q}(u, a)$ un objetivo. Observe que tanto $\mathfrak{P}$ como $\mathcal{O}$ son *definidos*.

A continuación, mostraremos una derivación-*SLD* de $\mathfrak{P} \cup \{\mathcal{O}\}$ de cada una de las formas descritas en la Definición 2.7. En todos los casos consideramos $\mathcal{O}_0 = \mathcal{O}$ y la regla de computación como el criterio que selecciona el literal más a la izquierda de la cláusula.

- Si la regla de búsqueda selecciona la cláusula $\mathcal{C}_1$, para resolverla con el literal $\mathcal{Q}(u, a)$, entonces:

$$\begin{array}{ll} \mathcal{O}_0 : & \leftarrow \mathcal{Q}(u, a) \\ \mathcal{C}_1 : & \mathcal{Q}(x, v) \leftarrow \mathcal{P}(x, v) \\ & \theta_1 = \{^u|_x, ^v|_a\} \\ \mathcal{O}_1 : & \leftarrow \mathcal{P}(x, a) \end{array}$$

luego, si la *regla de búsqueda* selecciona la cláusula $\mathcal{C}_3$, se tiene:

$$\begin{aligned}\mathcal{O}_1 : & \quad \leftarrow \mathcal{P}(x, a) \\ \mathcal{C}_3 : & \quad \mathcal{P}(a, a) \leftarrow \\ & \quad \theta_2 = \{x|_a\} \\ \mathcal{O}_2 : & \quad \square\end{aligned}$$

Por lo tanto, esta derivación-SLD es de tipo *finita*, de *longitud* 2 y finaliza en *éxito*. Observe que $\mathcal{C}_4$ no es unificable con $\mathcal{O}_2$.

- Si la *regla de búsqueda* selecciona la cláusula $\mathcal{C}_2$, para resolverla con el literal $\mathcal{Q}(u, a)$, entonces:

$$\begin{aligned}\mathcal{O}_0 : & \quad \leftarrow \mathcal{Q}(u, a) \\ \mathcal{C}_2 : & \quad \mathcal{Q}(x, v) \leftarrow \mathcal{P}(x, w), \mathcal{Q}(w, v) \\ & \quad \theta_1 = \{u|_x, v|_a\} \\ \mathcal{O}_1 : & \quad \leftarrow \mathcal{P}(x, w), \mathcal{Q}(w, a)\end{aligned}$$

luego, si la *regla de búsqueda* selecciona la cláusula $\mathcal{C}_3$, se tiene:

$$\begin{aligned}\mathcal{O}_1 : & \quad \leftarrow \mathcal{P}(x, w), \mathcal{Q}(w, a) \\ \mathcal{C}_3 : & \quad \mathcal{P}(a, a) \leftarrow \\ & \quad \theta_2 = \{x|_a, w|_a\} \\ \mathcal{O}_2 : & \quad \leftarrow \mathcal{Q}(a, a)\end{aligned}$$

así, si nuevamente seleccionamos la cláusula $\mathcal{C}_2$, se tiene:

$$\begin{aligned}\mathcal{O}_2 : & \quad \leftarrow \mathcal{Q}(a, a) \\ \mathcal{C}_2 : & \quad \mathcal{Q}(u, v) \leftarrow \mathcal{P}(u, w), \mathcal{Q}(w, v) \\ & \quad \theta_3 = \{u|_a, v|_a\} \\ \mathcal{O}_3 : & \quad \leftarrow \mathcal{P}(a, w), \mathcal{Q}(w, a)\end{aligned}$$

ahora, si la *regla de búsqueda* selecciona nuevamente $\mathcal{C}_3$, se tiene:

$$\begin{aligned}\mathcal{O}_3 : & \quad \leftarrow \mathcal{P}(a, w), \mathcal{Q}(w, a) \\ \mathcal{C}_3 : & \quad \mathcal{P}(a, a) \leftarrow \\ & \quad \theta_4 = \{w|_a\} \\ \mathcal{O}_4 : & \quad \leftarrow \mathcal{Q}(a, a)\end{aligned}$$

Observe que sea llegado a un punto en el cual, el objetivo $\mathcal{O}_4$ es igual al objetivo $\mathcal{O}_2$, lo que permite aplicar reiteradamente el proceso. Por lo tanto, esta derivación-SLD es de tipo *infinita*.

- Finalmente, si la *regla de búsqueda* selecciona primero la cláusula $\mathcal{C}_2$, para resolverla con el literal $\mathcal{Q}(u, a)$, entonces:

$$\begin{aligned}\mathcal{O}_0 : & \quad \leftarrow \mathcal{Q}(u, a) \\ \mathcal{C}_2 : & \quad \mathcal{Q}(x, v) \leftarrow \mathcal{P}(x, w), \mathcal{Q}(w, v) \\ & \quad \theta_1 = \{^u|_x, ^v|_a\} \\ \mathcal{O}_1 : & \quad \leftarrow \mathcal{P}(x, w), \mathcal{Q}(w, a)\end{aligned}$$

luego, si la *regla de búsqueda* selecciona la cláusula $\mathcal{C}_3$, se tiene:

$$\begin{aligned}\mathcal{O}_1 : & \quad \leftarrow \mathcal{P}(x, w), \mathcal{Q}(w, a) \\ \mathcal{C}_3 : & \quad \mathcal{P}(a, a) \leftarrow \\ & \quad \theta_2 = \{^x|_a, ^w|_a\} \\ \mathcal{O}_2 : & \quad \leftarrow \mathcal{Q}(a, a)\end{aligned}$$

y así, tenemos:

$$\begin{aligned}\mathcal{O}_2 : & \quad \leftarrow \mathcal{Q}(a, a) \\ \mathcal{C}_2 : & \quad \mathcal{Q}(u, v) \leftarrow \mathcal{P}(u, w), \mathcal{Q}(w, v) \\ & \quad \theta_3 = \{^u|_a, ^v|_a\} \\ \mathcal{O}_3 : & \quad \leftarrow \mathcal{P}(a, w), \mathcal{Q}(w, a)\end{aligned}$$

hasta aquí, el Principio de Resolución se ha empleado de manera idéntica al ítem anterior, pero ahora la *regla de búsqueda* seleccionará $\mathcal{C}_4$ y luego $\mathcal{C}_1$, así:

$$\begin{aligned}\mathcal{O}_3 : & \quad \leftarrow \mathcal{P}(a, w), \mathcal{Q}(w, a) \\ \mathcal{C}_4 : & \quad \mathcal{P}(a, b) \leftarrow \\ & \quad \theta_4 = \{^w|_b\} \\ \mathcal{O}_4 : & \quad \leftarrow \mathcal{Q}(b, a)\end{aligned}$$

lo que nos lleva,

$$\begin{aligned}\mathcal{O}_4 : & \quad \leftarrow \mathcal{Q}(b, a) \\ \mathcal{C}_1 : & \quad \mathcal{Q}(u, v) \leftarrow \mathcal{P}(u, v) \\ & \quad \theta_5 = \{^u|_b, ^v|_a\} \\ \mathcal{O}_5 : & \quad \leftarrow \mathcal{P}(b, a)\end{aligned}$$

donde este último objetivo $\mathcal{O}_5$ no es unificable con alguna cláusula de $\mathfrak{P}$. Por lo tanto, esta derivación-SLD es de tipo *finita*, de *longitud* 5 y finaliza en *fracaso*.

◆

En la siguiente definición, se clasifica uno de los tipos de derivación-SLD expuestos en la Definición 2.7; derivación que se convierte en la característica principal de nuestro método de deducción.

Definición 2.8. Una *refutación-SLD* de $\mathfrak{P} \cup \{\mathcal{O}\}$ es una *derivación-SLD finita* de $\mathfrak{P} \cup \{\mathcal{O}\}$ que finaliza en *éxito*, es decir, que tiene la cláusula vacía, $\square$, como último objetivo.

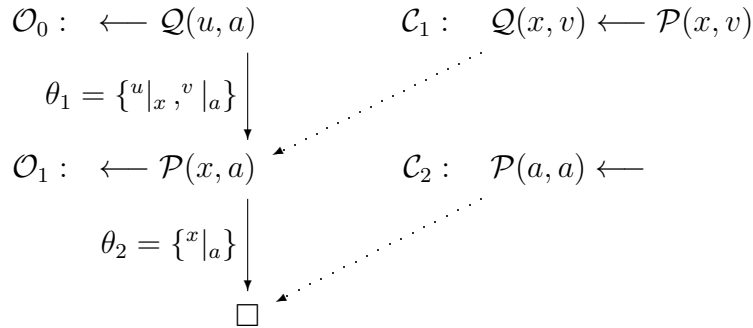
EJEMPLO 28. Con base en el Ejemplo 27, una *refutación-SLD* de $\mathfrak{P} \cup \{\mathcal{O}\}$, donde $\mathcal{O} : \leftarrow \mathcal{Q}(u, a)$, esta formada por:

- | | | |
|--|--|-----------------------------|
| • Los objetivos | • Las cláusulas | • Los <i>umg</i> 's |
| $\mathcal{O}_0 : \leftarrow \mathcal{Q}(u, a)$ | $\mathcal{C}_1 : \mathcal{Q}(u, v) \leftarrow \mathcal{P}(u, v)$ | $\theta_1 = \{u _x, v _a\}$ |
| $\mathcal{O}_1 : \leftarrow \mathcal{P}(x, a)$ | $\mathcal{C}_2 : \mathcal{P}(a, a) \leftarrow$ | $\theta_2 = \{x _a\}$ |
| $\mathcal{O}_2 : \square$ | | |

tal que $\mathcal{O} = \mathcal{O}_0$ y, $\mathcal{O}_1$ se deriva de $\mathcal{O}_0$ y $\mathcal{C}_1$ utilizando θ_1 , y $\mathcal{O}_2$ se deriva de $\mathcal{O}_1$ y $\mathcal{C}_2$ utilizando θ_2 . Esta refutación se puede representar tal como se muestra en la Gráfica 2.1.

◆

Como se indicó en la Observación 24, los subíndices empleados para las cláusulas en la refutación-SLD del Ejemplo 28 se asignaron de acuerdo al orden en que se emplearon en cada paso de la derivación, diferente (no necesariamente) al orden establecido en el programa lógico, a saber: la cláusula $\mathcal{C}_3$ del Ejemplo 27 se designó como $\mathcal{C}_2$ en el Ejemplo 28, ya que se empleó en el segundo paso de la refutación-SLD.



Gráfica 2.1. Refutación-SLD (Ejemplo 28)

Podemos agrupar todas las cláusulas (objetivos) que permiten calcular una refutación, usando el Principio de Resolución-SLD, de un programa lógico dado junto con la negación de la cláusula respectiva, es decir, agrupar las cláusulas que su negación junto a un programa lógico determinado tienen una derivación-SLD finita y finalizan en éxito; a este conjunto se le conocerá como **Conjunto de éxitos** y se define en la siguiente sección por requerir de un concepto determinado ahí.

En la Definición 2.4, hablamos de un tipo de *respuesta* que caracteriza la instancia de una cláusula que sea consecuencia lógica de un programa, llamada *respuesta correcta*; este tipo de respuesta está relacionada con la interpretación de la cláusula a partir de los modelos del programa lógico. Ahora definiremos una respuesta, equivalente a una *respuesta correcta* pero en contexto operativo, es decir, una respuesta que relaciona todos los *umg*'s (respuestas) empleados en cada paso de una derivación-SLD que finaliza en éxito, llamada *respuesta calculada*.

Definición 2.9. Una respuesta θ para $\mathfrak{P} \cup \{\mathcal{O}\}$, es una ***respuesta calculada*** para $\mathfrak{P} \cup \{\mathcal{O}\}$ si es la sustitución que se obtiene restringiendo la composición $\theta_n \cdots \theta_2 \theta_1$ a las variables de $\mathcal{O}$, donde $\theta_1, \theta_2, \dots, \theta_n$ es la secuencia de *umg*'s empleados en una refutación-SLD de $\mathfrak{P} \cup \{\mathcal{O}\}$ de longitud n .

EJEMPLO 29. Con base en la Gráfica 1.9 del Ejemplo 25, los unificadores más generales empleados en la refutación-SLD son:

$$\begin{aligned}\theta_1 &= \{x|_a\} \\ \theta_2 &= \{y|_x, z|_x, u|_{f(x)}\} \\ \theta_3 &= \{w|_a, z|_{f(x)}\} \\ \theta_4 &= \{y|_{f(c)}, x|_b\} \\ \theta_5 &= \{ \} \end{aligned}$$

luego, para hallar la *respuesta calculada* realizamos la composición

$$\begin{aligned}\theta_5 \theta_4 \theta_3 \theta_2 \theta_1 &= \{ \} \{y|_{f(c)}, x|_b\} \{w|_a, z|_{f(x)}\} \{y|_x, z|_x, u|_{f(x)}\} \{x|_a\} \\ &= \{ \} \{y|_{f(c)}, x|_b\} \{w|_a, z|_{f(x)}\} \{y|_x, z|_x, u|_{f(x)}, x|_a\} \\ &= \{ \} \{y|_{f(c)}, x|_b\} \left\{ w|_a, z|_{f(x)}, \overset{\text{eliminar}}{y|_x, z|_x, u|_{f(x)}, x|_a} \right\} \\ &= \{ \} \left\{ \overset{\text{eliminar}}{y|_{f(c)}}, \overset{\text{eliminar}}{x|_b}, w|_a, y|_{\theta_4[x]}, z|_{\theta_4[x]}, u|_{\theta_4[f(x)]}, x|_a \right\} \\ &= \{ \} \{w|_a, y|_b, z|_b, u|_{f(b)}, x|_a\} \\ &= \{w|_a, y|_b, z|_b, u|_{f(b)}, x|_a\} \end{aligned}$$

y con base en la Definición 2.9, restringiendo la composición a las variables u, x, y, z del objetivo, tenemos que

$$\theta = \{y|_b, z|_b, u|_{f(b)}, x|_a\}$$

es la *respuesta calculada*. ◆

EJEMPLO 30. Con base en el Ejemplo 28, los unificadores más generales empleados en la refutación-SLD de $\mathfrak{P} \cup \{\mathcal{O}\}$, donde $\mathcal{O} : \leftarrow Q(u, a)$, son:

$$\begin{aligned}\theta_1 &= \{^u|_x, ^v|_a\} \\ \theta_2 &= \{^x|_a\}\end{aligned}$$

luego, realizamos la composición

$$\theta_2\theta_1 = \{^x|_a\} \{^u|_x, ^v|_a\} = \{^x|_a, ^u|_{\theta_2[x]}, ^v|_a\} = \{^x|_a, ^u|_a, ^v|_a\}$$

y con base en la Definición 2.9, restringiendo la composición a las variables de $\mathcal{O}$, tenemos que

$$\theta = \{^u|_a\}$$

es la *respuesta calculada*. ◆

2.3. Semántica declarativa para la programación lógica

La semántica declarativa define un programa lógico en términos de los valores (o significados) obtenidos por las estructuras definidas para el programa lógico mismo. Desde este punto de vista podemos aplicar a los programas lógicos los conceptos de interpretación, asignación y modelos, vistos en la Sección 1.2.

Como se indicó en la página 32, la insatisfacibilidad de una fórmula (cláusula) representa el papel principal en la forma de operar nuestro método deductivo y, con base en la Proposición 1.2, para probar que una cláusula (objetivo) es derivada de un programa lógico, es decir, que es una consecuencia lógica, debemos probar que el programa lógico junto con la negación de la cláusula es un conjunto insatisfacible, lo que implica probar que toda cláusula de este conjunto es interpretada en cero (falso) para todas las estructuras y asignaciones que se puedan definir para este programa lógico, lo cual es una tarea casi imposible (en general) de lograr debido al sinnúmero de posibilidades que pueden asignarse tanto a las estructuras como a los dominios.

Un año antes de su muerte, Jacques Herbrand presentó en 1930, una teoría (que lleva su nombre) que garantiza de manera equivalente que cualquier conjunto de cláusulas es insatisfacible sí, y solo sí, es insatisfacible para un tipo especial de interpretaciones sobre un dominio particular. Este resultado permite así reducir el proceso de probar la insatisfacibilidad, para todas las posibles estructuras y dominios, a sólo las del tipo definido por Herbrand.

En esta sección veremos que para un conjunto de cláusulas (programa lógico), existe un conjunto de estructuras canónicas con un sistema extremadamente simple tal que, si un conjunto de cláusulas tiene un modelo entonces este tiene un modelo de la nueva forma definida; a este nuevo modelo lo conoceremos como *modelo de Herbrand* y además, construiremos un conjunto que logre representar a todos los modelos de Herbrand.

También definiremos el *conjunto de éxitos* como lo enunciamos en la página 47 (Sección 2.2) y describiremos un conjunto que tiene estructura de retículo, este último hecho será de gran utilidad en la siguiente sección.

Primero definiremos algunos conceptos básicos de la teoría de Herbrand. Para las siguientes definiciones, $\mathfrak{P}$ es un programa lógico.

Definición 2.10. Se define el *universo de Herbrand* para $\mathfrak{P}$, denotado por $\mathcal{U}_{\mathfrak{P}}$, como el conjunto de todos los *términos básicos* que pueden formarse a partir de las *constantes* y *símbolos de función* que aparecen en $\mathfrak{P}$. Si en $\mathfrak{P}$ no hay constantes, se debe agregar una constante arbitraria (pero fija) para poder construir los términos básicos.

EJEMPLO 31. Hallaremos el *universo de Herbrand* para los siguientes programas lógicos, donde a, b son constantes, u, v variables, $\mathcal{P}, \mathcal{Q}$ símbolos de relación y f, g símbolos de función unario y binario, respectivamente:

(i)

$$\mathfrak{P} = \left\{ \begin{array}{l} \bullet \mathcal{Q}(u, v) \leftarrow \mathcal{P}(u) \\ \bullet \mathcal{Q}(v, v) \leftarrow \\ \bullet \mathcal{P}(v) \leftarrow \end{array} \right.$$

$$\mathcal{U}_{\mathfrak{P}} = \{d\} \text{ (} d \text{ constante arbitraria)}$$

(ii)

$$\mathfrak{P} = \left\{ \begin{array}{l} \bullet \mathcal{Q}(a, v) \leftarrow \mathcal{P}(b) \\ \bullet \mathcal{Q}(u, b) \leftarrow \\ \bullet \mathcal{P}(v) \leftarrow \end{array} \right.$$

$$\mathcal{U}_{\mathfrak{P}} = \{a, b\}$$

(iii)

$$\mathfrak{P} = \left\{ \begin{array}{l} \bullet \mathcal{Q}(f(a), v) \leftarrow \mathcal{P}(f(v)) \\ \bullet \mathcal{Q}(u, f(b)) \leftarrow \\ \bullet \mathcal{P}(a) \leftarrow \end{array} \right.$$

$$\mathcal{U}_{\mathfrak{P}} = \{a, b, f(a), f(b), f(f(a)), f(f(b)), f(f(f(a))), f(f(f(b))), \dots\}$$

$$(iv) \quad \mathfrak{P} = \begin{cases} \bullet & \mathcal{Q}(f(u), v) \longleftarrow \mathcal{P}(u) \\ \bullet & \mathcal{Q}(g(u, f(u)), f(v)) \longleftarrow \\ \bullet & \mathcal{P}(g(u, v)) \longleftarrow \end{cases}$$

$$\mathfrak{U}_{\mathfrak{P}} = \{d, f(d), g(d, d), g(d, f(d)), g(f(d), d), g(f(d), f(d)), f(f(d)), \\ f(g(d, d)), f(g(d, f(d))), f(g(f(d), d)), f(g(f(d), f(d))), g(g(d, d), d), \dots\}$$

◆

Observación 25. *El universo de Herbrand de cualquier programa lógico $\mathfrak{P}$ es siempre no vacío, ya que en el peor de los casos se adiciona una constante cualquiera; además es finito sí, y solo sí, no aparecen símbolos de función en $\mathfrak{P}$.*

Existe una descripción para el *universo de Herbrand* un poco más general, tal como lo emplea Ortíz[pág. 89][29] y Ben-Ari[pág. 136][1], que claramente corresponde con la definición que hemos resuelto emplear, veamos:

“ Sea $\mathcal{C}_0$ el conjunto de constantes que aparecen en el programa lógico $\mathfrak{P}$; si no existe alguna constante, entonces se considera $\mathcal{C}_0 = \{d\}$, donde d es una constante cualquiera pero fija.

Para $i > 0$, se define $\mathcal{C}_i = \mathcal{C}_{i-1} \cup \{f(t_1, \dots, t_n) \mid f \text{ es un símbolo de función } n\text{-ario que aparece en } \mathfrak{P} \text{ y } t_1, \dots, t_n \text{ son elementos de } \mathcal{C}_{i-1}\}$.

Entonces, el universo de Herbrand para $\mathfrak{P}$ es

$$\mathfrak{U}_{\mathfrak{P}} = \lim_{i \rightarrow \infty} \mathcal{C}_i \text{ ”}$$

EJEMPLO 32. Con base en el comentario anterior, el *universo de Herbrand* para el programa lógico del ítem **(iii)** del Ejemplo 31 se forma así:

$$\begin{aligned} \mathcal{C}_0 &= \{a, b\} \\ \mathcal{C}_1 &= \mathcal{C}_0 \cup \{f(a), f(b)\} = \{a, b, f(a), f(b)\} \\ \mathcal{C}_2 &= \mathcal{C}_1 \cup \{f(a), f(b), f(f(a)), f(f(b))\} = \{a, b, f(a), f(b), f(f(a)), f(f(b))\} \\ &\vdots \\ \mathcal{C}_i &= \mathcal{C}_{i-1} \cup \{f(t) \mid t \in \mathcal{C}_{i-1}\} \\ &\vdots \\ \mathfrak{U}_{\mathfrak{P}} &= \lim_{i \rightarrow \infty} \mathcal{C}_i = \{a, b, f(a), f(b), f(f(a)), f(f(b)), f(f(f(a))), f(f(f(b))), \dots\} \end{aligned}$$

Para el programa lógico del ítem **(iv)** del Ejemplo 31 el *universo de Herbrand* será:

$$\begin{aligned}
\mathcal{C}_0 &= \{d\} \\
\mathcal{C}_1 &= \{d, f(d), g(d, d)\} \\
\mathcal{C}_2 &= \{d, f(d), g(d, d), f(f(d)), f(g(d, d)), g(d, f(d)), g(f(d), d), g(f(d), f(d)), \\
&\quad g(d, g(d, d)), g(g(d, d), d), g(f(d), g(d, d)), g(g(d, d), f(d)), g(g(d, d), g(d, d))\} \\
&\vdots \\
\mathcal{C}_i &= \mathcal{C}_{i-1} \cup \{f(t) \mid t \in \mathcal{C}_{i-1}\} \cup \{g(t, t) \mid t \in \mathcal{C}_{i-1}\} \\
&\vdots \\
\mathcal{U}_{\mathfrak{P}} &= \{d, f(d), g(d, d), g(d, f(d)), g(f(d), d), g(f(d), f(d)), f(f(d)), f(g(d, d)), \\
&\quad f(g(d, f(d))), f(g(f(d), d)), f(g(f(d), f(d))), g(g(d, d), d), \dots\}
\end{aligned}$$

◆

Definición 2.11. Sea $\mathcal{U}_{\mathfrak{P}}$ el universo de Herbrand de $\mathfrak{P}$. La **base de Herbrand** de $\mathfrak{P}$, denotada por $\mathfrak{B}_{\mathfrak{P}}$, es el conjunto de todos los *átomos* que pueden formarse a partir de los símbolos de relación que aparecen en $\mathfrak{P}$ y los términos que aparecen en $\mathcal{U}_{\mathfrak{P}}$.

EJEMPLO 33. Con base en el Ejemplo 31, la *base de Herbrand* para cada ítem, respectivamente, será:

- (i) $\mathcal{U}_{\mathfrak{P}} = \{d\}$ (d una constante arbitraria)
 $\mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(d), \mathcal{Q}(d, d)\}$
- (ii) $\mathcal{U}_{\mathfrak{P}} = \{a, b\}$
 $\mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(a), \mathcal{P}(b), \mathcal{Q}(a, a), \mathcal{Q}(a, b), \mathcal{Q}(b, a), \mathcal{Q}(b, b)\}$
- (iii) $\mathcal{U}_{\mathfrak{P}} = \{a, b, f(a), f(b), f(f(a)), f(f(b)), f(f(f(a))), f(f(f(b))), \dots\}$
 $\mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(a), \mathcal{P}(b), \mathcal{P}(f(a)), \mathcal{P}(f(b)), \mathcal{P}(f(f(a))), \mathcal{P}(f(f(b))), \dots, \\ \mathcal{Q}(a, a), \mathcal{Q}(a, b), \mathcal{Q}(b, a), \mathcal{Q}(b, b), \mathcal{Q}(a, f(a)), \mathcal{Q}(a, f(b)), \dots\}$
- (iv) $\mathcal{U}_{\mathfrak{P}} = \{d, f(d), g(d, d), g(d, f(d)), g(f(d), d), g(f(d), f(d)), \dots, \\ f(f(d)), f(g(d, d)), f(g(d, f(d))), g(g(d, d), d), \dots\}$
 $\mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(d), \mathcal{P}(f(d)), \mathcal{P}(g(d, d)), \mathcal{P}(g(d, f(d))), \mathcal{P}(g(f(d), d)), \dots, \\ \mathcal{Q}(d, d), \mathcal{Q}(d, f(d)), \mathcal{Q}(d, g(d, d)), \mathcal{Q}(d, g(d, f(d))), \dots\}$

◆

Observación 26. *La base de Herbrand de $\mathfrak{P}$ es siempre no vacía, ya que cualquier programa lógico debe contener por lo menos un literal; además es finita sí, y solo sí, el universo de Herbrand para $\mathfrak{P}$ es finito.*

Observación 27. *Debido a la descripción de los elementos del universo de Herbrand, los átomos que se forman en la base de Herbrand son átomos básicos, es decir, átomos que no contienen variables.*

Hasta aquí hemos descrito dos conjuntos de la Teoría de Herbrand el *Universo* y la *Base*, los cuales están formados por elementos básicos que se encuentran explícitos en un programa lógico; y este hecho los convertirá en uno de los conjuntos más destacados en nuestro estudio.

A continuación, se define el conjunto al que se hizo referencia en la Sección 2.2 página 47. Aunque este conjunto constituye la semántica operacional de un programa lógico, sólo se define hasta este momento por que se relaciona con uno de los conjuntos más representativos de la teoría desarrollada en esta sección: la *Base de Herbrand*.

Definición 2.12. Sea $\mathfrak{P}$ un programa lógico definido. El **conjunto de éxitos** (*Success Set*) de $\mathfrak{P}$, denotado por $\mathcal{SS}_{\mathfrak{P}}$, es el conjunto de todos los átomos básicos tal que su negación junto al programa lógico tienen una refutación-SLD, es decir,

$$\mathcal{SS}_{\mathfrak{P}} = \{\mathcal{R} \in \mathfrak{B}_{\mathfrak{P}} \mid \mathfrak{P} \cup \{\neg \mathcal{R}\} \text{ tiene una refutación-SLD}\}$$

donde $\mathcal{R}$ denota un elemento típico de $\mathfrak{B}_{\mathfrak{P}}$.

La razón por la cual se definió el *conjunto de éxitos* para programas lógicos *definidos* se analizará más adelante y se relaciona con la prueba de *completitud* del Principio de Resolución-SLD. Ahora, se define una estructura con una característica particular que le brinda la teoría de Herbrand.

Definición 2.13. Una **estructura de Herbrand** o **$\mathfrak{H}$ -estructura** para $\mathfrak{P}$ es una estructura $\mathfrak{E} = \langle \mathcal{D}, \mathcal{I} \rangle$ de $\mathfrak{P}$, tal que $\mathcal{D} = \mathfrak{U}_{\mathfrak{P}}$, es decir, el dominio es el universo de Herbrand para $\mathfrak{P}$ e $\mathcal{I} = \mathcal{I}_{\mathfrak{H}}$, donde $\mathcal{I}_{\mathfrak{H}}$ denota la interpretación, llamada **interpretación de Herbrand** o **$\mathfrak{H}$ -interpretación**, que asocia:

- A cada constante c , la misma constante, es decir, $c^{\mathcal{I}_{\mathfrak{H}}} = c$.
- A cada símbolo de función n -ario f , la misma función pero definida en $\mathfrak{U}_{\mathfrak{P}}$, es decir, $f^{\mathcal{I}_{\mathfrak{H}}} = f$ donde $f^{\mathcal{I}_{\mathfrak{H}}} : \mathfrak{U}_{\mathfrak{P}}^n \rightarrow \mathfrak{U}_{\mathfrak{P}}$.
- A cada símbolo de relación n -ario $\mathcal{R}$, una relación n -aria $\mathcal{R}^{\mathcal{I}_{\mathfrak{H}}} \subseteq \mathfrak{U}_{\mathfrak{P}}^n$, es decir, $\mathcal{R}^{\mathcal{I}_{\mathfrak{H}}}$ será un conjunto formado por n -uplas de elementos de $\mathfrak{U}_{\mathfrak{P}}$.

En conclusión, denotamos una $\mathfrak{H}$ -estructura por $\mathfrak{E}_{\mathfrak{H}}$, donde $\mathfrak{E}_{\mathfrak{H}} = \langle \mathfrak{U}_{\mathfrak{P}}, \mathcal{I}_{\mathfrak{H}} \rangle$.

Observación 28. *En la Definición 2.13, no se coloca ninguna restricción a la interpretación de símbolos de relación.*

Note que una $\mathfrak{H}$ -estructura establece el dominio, interpreta las constantes y símbolos de función, por ellas mismas, dando como resultado elementos del universo de Herbrand; y para los símbolos de función, los interpreta como un conjunto de tuplas del universo de Herbrand; es decir, las $\mathfrak{H}$ -estructuras interpretan sobre su dominio, de la misma forma que se definió en 1.10; por esta razón, las $\mathfrak{H}$ -estructuras son también estructuras en el sentido clásico.

Observe que las $\mathfrak{H}$ -estructuras definidas para un programa lógico se diferencian únicamente por la interpretación que se asigne a los símbolos de relación; por esta razón, podemos definir tantas $\mathfrak{H}$ -estructuras como $\mathfrak{H}$ -interpretaciones se asignen a estos símbolos. Convenimos en denotar las $\mathfrak{H}$ -estructuras únicamente por las interpretaciones de los símbolos de relación, formando así un conjunto de instancias básicas de los átomos presentes en el programa lógico, que se *interpretan en 1* (verdadero) por la $\mathfrak{H}$ -interpretación respectiva. De esta forma si un átomo no aparece en una $\mathfrak{H}$ -estructura implica que es *interpretado en 0* (falso) por la $\mathfrak{H}$ -interpretación respectiva.

Debido a que la base de Herbrand contiene todas las combinaciones posibles que forman átomos básicos de un programa lógico, entonces cada subconjunto de la base de Herbrand puede verse como una $\mathfrak{H}$ -estructura, con la característica de que cada uno de estos átomos se interpretan como verdaderos.

Por lo anterior, y por lo expuesto en Ortíz[29], representaremos simplemente una $\mathfrak{H}$ -estructura para $\mathfrak{P}$ de la siguiente forma:

$$\mathfrak{E}_{\mathfrak{H}} = \langle \mathfrak{U}_{\mathfrak{P}}, \mathfrak{I}_{\mathfrak{H}} \rangle = \{ \mathcal{R}(t_1, \dots, t_n) \in \mathfrak{B}_{\mathfrak{P}} \mid [\mathcal{R}(t_1, \dots, t_n)]^{\mathfrak{I}_{\mathfrak{H}}} = 1 \}$$

donde $\mathcal{R}$ denota un símbolo de relación presente en $\mathfrak{P}$ y $(t_1, \dots, t_n)$, por la definición de $\mathfrak{B}_{\mathfrak{P}}$, son claramente elementos de $\mathfrak{U}_{\mathfrak{P}}$.

Por lo tanto, para conocer el valor de verdad de un átomo básico en una $\mathfrak{H}$ -estructura simplemente debemos verificar si pertenece o no al subconjunto de $\mathfrak{B}_{\mathfrak{P}}$ equivalente a la $\mathfrak{H}$ -estructura.

Observación 29. *A diferencia de lo descrito en la Sección 1.2, donde se requería de una función que interpretará las variables por elementos del dominio (Definición 1.11), las $\mathfrak{H}$ -interpretaciones definirán de manera intrínseca la forma de interpretar las variables, asignándole un elemento del universo de Herbrand y caracterizándose así misma por esta asignación.*

Observación 30. *Toda $\mathfrak{H}$ -estructura viene dada por la asignación de valores de verdad a cada uno de los elementos de la base de Herbrand.*

EJEMPLO 34. Sea $\mathfrak{P}$ el programa lógico formado por las cláusulas:

$$(i) \quad \mathfrak{P} = \left\{ \begin{array}{l} \bullet \quad \mathcal{Q}(u, v) \longleftarrow \mathcal{P}(u) \\ \bullet \quad \mathcal{R}(v) \longleftarrow \mathcal{P}(u) \end{array} \right.$$

El universo y la base de Herbrand de $\mathfrak{P}$ son:

$$\mathfrak{U}_{\mathfrak{P}} = \{d\} \text{ (} d \text{ una constante arbitraria)}$$

$$\mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(d), \mathcal{Q}(d, d), \mathcal{R}(d)\}$$

Algunas $\mathfrak{H}$ -estructuras de $\mathfrak{P}$ son:

$$\begin{array}{ll} \mathfrak{E}_{\mathfrak{H}_1} = \mathfrak{B}_{\mathfrak{P}} & \mathfrak{E}_{\mathfrak{H}_2} = \emptyset \\ \mathfrak{E}_{\mathfrak{H}_3} = \{\mathcal{Q}(d, d)\} & \mathfrak{E}_{\mathfrak{H}_4} = \{\mathcal{P}(d), \mathcal{R}(d)\} \\ \mathfrak{E}_{\mathfrak{H}_5} = \{\mathcal{P}(d), \mathcal{Q}(d, d)\} & \mathfrak{E}_{\mathfrak{H}_6} = \{\mathcal{P}(d)\} \end{array}$$

Note que $\mathfrak{E}_{\mathfrak{H}_6}$ interpreta en 1 a $\mathcal{P}(d)$ e interpreta en 0 a $\mathcal{Q}(d, d)$ y a $\mathcal{R}(d)$.

$$(ii) \quad \mathfrak{P} = \left\{ \begin{array}{l} \bullet \quad \mathcal{Q}(f(a), v) \longleftarrow \mathcal{P}(f(v)) \\ \bullet \quad \mathcal{Q}(u, f(u)) \longleftarrow \end{array} \right.$$

El universo y la base de Herbrand de $\mathfrak{P}$ son:

$$\mathfrak{U}_{\mathfrak{P}} = \{a, f(a), f(f(a)), f(f(f(a))), \dots\}$$

$$\mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(a), \mathcal{Q}(a, a), \mathcal{P}(f(a)), \mathcal{Q}(a, f(a)), \mathcal{Q}(f(a), a), \mathcal{Q}(f(a), f(a)), \dots, \\ \mathcal{P}(f(f(a))), \mathcal{Q}(a, f(f(a))), \mathcal{Q}(f(a), f(f(a))), \dots\}$$

Algunas $\mathfrak{H}$ -estructuras de $\mathfrak{P}$ son:

$$\begin{array}{l} \mathfrak{E}_{\mathfrak{H}_1} = \mathfrak{B}_{\mathfrak{P}} \\ \mathfrak{E}_{\mathfrak{H}_2} = \{\mathcal{P}(a), \mathcal{Q}(f(a), a), \mathcal{P}(f(f(f(a))))\} \\ \mathfrak{E}_{\mathfrak{H}_3} = \{\mathcal{P}(f(a)), \mathcal{Q}(f(a), a), \mathcal{Q}(a, f(a))\} \\ \mathfrak{E}_{\mathfrak{H}_4} = \{\mathcal{Q}(f(a), f(f(a)))\} \\ \mathfrak{E}_{\mathfrak{H}_5} = \emptyset \end{array}$$

◆

Observación 31. Debido a que las $\mathfrak{H}$ -estructuras son subconjuntos de $\mathfrak{B}_{\mathfrak{P}}$, existen tantas $\mathfrak{H}$ -estructuras como la cardinalidad de $\wp(\mathfrak{B}_{\mathfrak{P}})$, donde $\wp(\cdot)$ denota el **conjunto de partes** o **conjunto potencia** usual. Por tanto, $\wp(\mathfrak{B}_{\mathfrak{P}})$ está formado por todas las $\mathfrak{H}$ -estructuras de $\mathfrak{P}$.

Haremos un paréntesis en el desarrollo de la teoría de Herbrand para hablar un poco de $\wp(\mathfrak{B}_{\mathfrak{P}})$, denotando una característica muy relevante y útil para el contenido desarrollado en la siguiente sección.

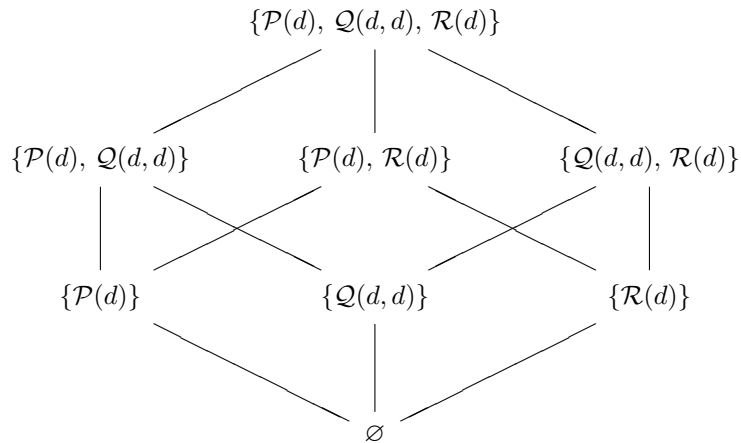
En Rodríguez[35] y Komendantskaya[17] se encuentra la definición de *retículo* y *retículo completo*; para este último dicen:

“ Un retículo en el cual cada subconjunto (incluyendo los infinitos) tiene tanto un supremo como un ínfimo se llama un **retículo completo**. ”

En particular, si para un conjunto dado ordenamos los subconjuntos con la relación de inclusión conjuntista “ $\subseteq$ ”, entonces obtenemos un *retículo completo*, donde el supremo (sup) está dado por la unión y el ínfimo (ínf) por la intersección de subconjuntos. Por consiguiente, los *retículos completos* siempre son acotados.

Entonces, podemos concluir por las Observaciones 26 y 31 y bajo el orden parcial “ $\subseteq$ ”, que $\wp(\mathfrak{B}_{\mathfrak{P}})$ posee una estructura de retículo, en particular, es un *retículo completo*, donde su ínfimo (bottom) es $\emptyset$ y su supremo (top) es $\mathfrak{B}_{\mathfrak{P}}$.

EJEMPLO 35. Haciendo uso de los diagramas de Hasse, podemos representar el *retículo completo* formado por $\wp(\mathfrak{B}_{\mathfrak{P}})$, para el ítem (i) del Ejemplo 34, donde $\mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(d), \mathcal{Q}(d, d), \mathcal{R}(d)\}$ (Ver Gráfica 2.2).



Gráfica 2.2. Diagrama de Hasse del retículo completo $\wp(\mathfrak{B}_{\mathfrak{P}})$ (Ejemplo 35)



Como lo indica Simari[36], la siguiente definición especifica cuando una *regla* es *interpretada en 1* por una $\mathfrak{H}$ -estructura; tal como se indicó en la Definición 1.15 para el caso clásico. Nos referimos a *reglas* y *hechos* tal como se designaron en la página 40.

Observación 32. *Los hechos son interpretados en 1 por una $\mathfrak{H}$ -estructura si, la instancia básica del átomo que corresponde al hecho, pertenece a la $\mathfrak{H}$ -estructura.*

Definición 2.14. Una $\mathfrak{H}$ -estructura $\mathfrak{E}_{\mathfrak{H}}$ se dice que *interpreta en 1* a una regla

$$\mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p, \text{ con } p \geq 1,$$

si para toda instancia básica $\mathcal{A}^* \leftarrow \mathcal{B}_1^*, \dots, \mathcal{B}_p^*$ de la regla se tiene; siempre que $\mathcal{B}_i^* \in \mathfrak{E}_{\mathfrak{H}}$, para todo i , $1 \leq i \leq p$, entonces $\mathcal{A}^* \in \mathfrak{E}_{\mathfrak{H}}$.

Definición 2.15. Un *modelo de Herbrand* o *$\mathfrak{H}$ -modelo* para $\mathfrak{P}$ es una $\mathfrak{H}$ -estructura que es un modelo de $\mathfrak{P}$, es decir, es una $\mathfrak{H}$ -estructura que *interpreta en 1* a todas las reglas y hechos presentes en $\mathfrak{P}$.

EJEMPLO 36. Con base en el Ejemplo 34, son algunos $\mathfrak{H}$ -modelos de $\mathfrak{P}$ las $\mathfrak{H}$ -estructuras:

- para el ítem (i), $\mathfrak{E}_{\mathfrak{H}_2} = \emptyset$, $\mathfrak{E}_{\mathfrak{H}_3} = \{\mathcal{Q}(d, d)\}$ y $\mathfrak{E}_{\mathfrak{H}_4} = \{\mathcal{P}(d), \mathcal{R}(d)\}$; note que $\mathfrak{E}_{\mathfrak{H}_6} = \{\mathcal{P}(d)\}$ no es un $\mathfrak{H}$ -modelo por que $\mathcal{Q}(d, d)$ y $\mathcal{R}(d)$ no pertenecen a $\mathfrak{E}_{\mathfrak{H}_6}$, condición necesaria para que las dos reglas sean *interpretadas en 1*, según la Definición 2.14.
- para el ítem (ii), $\mathfrak{E}_{\mathfrak{H}_1} = \mathfrak{B}_{\mathfrak{P}}$, $\mathfrak{E}_{\mathfrak{H}_3} = \{\mathcal{P}(f(a)), \mathcal{Q}(f(a), a), \mathcal{Q}(a, f(a))\}$ y $\mathfrak{E}_{\mathfrak{H}_4} = \{\mathcal{Q}(f(a), f(f(a)))\}$.

◆

EJEMPLO 37. Sea $\mathfrak{P}$ el programa lógico formado por las cláusulas:

$$\mathfrak{P} = \left\{ \begin{array}{l} \bullet \mathcal{P}(u, v) \leftarrow \mathcal{Q}(u) \\ \bullet \mathcal{R}(v) \leftarrow \\ \bullet \mathcal{P}(u, u) \leftarrow \\ \bullet \mathcal{R}(v) \leftarrow \mathcal{S}(v, u) \end{array} \right.$$

El universo y la base de Herbrand de $\mathfrak{P}$ son:

$$\begin{aligned} \mathfrak{U}_{\mathfrak{P}} &= \{d\} \text{ (} d \text{ una constante arbitraria)} \\ \mathfrak{B}_{\mathfrak{P}} &= \{\mathcal{P}(d, d), \mathcal{Q}(d), \mathcal{R}(d), \mathcal{S}(d, d)\} \end{aligned}$$

Consideraremos todas las $\mathfrak{H}$ -estructuras de $\mathfrak{P}$. Dado que $\mathfrak{B}_{\mathfrak{P}}$ tiene cuatro elementos, por la Observación 31, existen 2^4 $\mathfrak{H}$ -estructuras, veamos:

$$\begin{array}{ll}
 \mathfrak{E}_{\mathfrak{H}_1} = \emptyset & \mathfrak{E}_{\mathfrak{H}_9} = \{\mathcal{P}(d, d)\} \\
 \mathfrak{E}_{\mathfrak{H}_2} = \{\mathcal{Q}(d)\} & \mathfrak{E}_{\mathfrak{H}_{10}} = \{\mathcal{R}(d)\} \\
 \mathfrak{E}_{\mathfrak{H}_3} = \{\mathcal{S}(d, d)\} & \mathfrak{E}_{\mathfrak{H}_{11}} = \{\mathcal{P}(d, d), \mathcal{Q}(d)\} \\
 \mathfrak{E}_{\mathfrak{H}_4} = \{\mathcal{P}(d, d), \mathcal{R}(d)\} & \mathfrak{E}_{\mathfrak{H}_{12}} = \{\mathcal{P}(d, d), \mathcal{S}(d, d)\} \\
 \mathfrak{E}_{\mathfrak{H}_5} = \{\mathcal{Q}(d), \mathcal{R}(d)\} & \mathfrak{E}_{\mathfrak{H}_{13}} = \{\mathcal{Q}(d), \mathcal{S}(d, d)\} \\
 \mathfrak{E}_{\mathfrak{H}_6} = \{\mathcal{R}(d), \mathcal{S}(d, d)\} & \mathfrak{E}_{\mathfrak{H}_{14}} = \{\mathcal{P}(d, d), \mathcal{Q}(d), \mathcal{R}(d)\} \\
 \mathfrak{E}_{\mathfrak{H}_7} = \{\mathcal{P}(d, d), \mathcal{Q}(d), \mathcal{S}(d, d)\} & \mathfrak{E}_{\mathfrak{H}_{15}} = \{\mathcal{P}(d, d), \mathcal{R}(d), \mathcal{S}(d, d)\} \\
 \mathfrak{E}_{\mathfrak{H}_8} = \{\mathcal{Q}(d), \mathcal{R}(d), \mathcal{S}(d, d)\} & \mathfrak{E}_{\mathfrak{H}_{16}} = \mathfrak{B}_{\mathfrak{P}}
 \end{array}$$

Finalmente, seleccionamos únicamente las $\mathfrak{H}$ -estructuras que son $\mathfrak{H}$ -modelos de $\mathfrak{P}$, a saber:

$$\begin{array}{ll}
 \mathfrak{E}_{\mathfrak{H}_4} = \{\mathcal{P}(d, d), \mathcal{R}(d)\} & \mathfrak{E}_{\mathfrak{H}_{14}} = \{\mathcal{P}(d, d), \mathcal{Q}(d), \mathcal{R}(d)\} \\
 \mathfrak{E}_{\mathfrak{H}_{15}} = \{\mathcal{P}(d, d), \mathcal{R}(d), \mathcal{S}(d, d)\} & \mathfrak{E}_{\mathfrak{H}_{16}} = \mathfrak{B}_{\mathfrak{P}}
 \end{array}$$

◆

Observación 33.

- Cada una de las $\mathfrak{H}$ -estructuras constituyen un $\mathfrak{H}$ -modelo para el subconjunto de $\mathfrak{B}_{\mathfrak{P}}$ que representa.
- $\mathfrak{B}_{\mathfrak{P}}$ siempre será $\mathfrak{H}$ -modelo de $\mathfrak{P}$, para todo $\mathfrak{P}$.
- $\emptyset$ será $\mathfrak{H}$ -modelo de $\mathfrak{P}$ sí, y solo sí, $\mathfrak{P}$ no posee hechos.

Es importante notar que en la definición 2.15 se define un $\mathfrak{H}$ -modelo para un programa lógico (o equivalentemente, un conjunto estándar de cláusulas), programa lógico que consiste de sentencias cuantificadas universalmente, tal como lo describimos en las páginas 18 y 21; ya que, no para todo conjunto de sentencias existe un $\mathfrak{H}$ -modelo, como podemos verlo en el siguiente ejemplo.

EJEMPLO 38. Sea $\mathcal{S}$ el conjunto de sentencias $\{\mathcal{P}(a), \exists v \neg \mathcal{P}(v)\}$. El universo y la base de Herbrand de $\mathcal{S}$ son:

$$\mathcal{U}_{\mathfrak{P}} = \{a\} \quad \text{y} \quad \mathcal{B}_{\mathfrak{P}} = \{\mathcal{P}(a)\}.$$

Como la base de Herbrand sólo tiene un elemento, sólo hay dos posibles $\mathfrak{H}$ -estructuras de $\mathcal{S}$, a saber:

$$\mathfrak{E}_{\mathfrak{H}_1} = \emptyset \quad \text{y} \quad \mathfrak{E}_{\mathfrak{H}_2} = \mathcal{B}_{\mathfrak{P}}.$$

Ahora, $\mathfrak{E}_{\mathfrak{H}_2}$ no es modelo de $\exists v \neg \mathcal{P}(v)$ y $\mathfrak{E}_{\mathfrak{H}_1}$ no es modelo de $\mathcal{P}(a)$. Por lo tanto, ni $\mathfrak{E}_{\mathfrak{H}_1}$ ni $\mathfrak{E}_{\mathfrak{H}_2}$ son $\mathfrak{H}$ -modelos, es decir, $\mathcal{S}$ no tiene $\mathfrak{H}$ -modelos.

Sin embargo, $\mathcal{S}$ si posee modelos. Consideremos la estructura $\mathfrak{E}_{\mathfrak{H}} = \langle \mathcal{D}, \mathcal{I} \rangle$, siendo:

$$\mathcal{D} = \{7, 8\} \quad a^{\mathcal{I}} = 7 \quad \mathcal{P}^{\mathcal{I}} = \{7\}$$

Entonces, $\mathcal{P}(7)$ es verdadero y $\exists v \neg \mathcal{P}(v)$ es verdadero para $v = 8$, por lo tanto, $\mathfrak{E}$ es un modelo de $\mathcal{S}$. ◆

En el ejemplo anterior podemos ver que, la no existencia de un modelo de Herbrand para $\mathcal{S}$ no significa que $\mathcal{S}$ no pueda tener un modelo, es decir, que $\mathcal{S}$ sea insatisfacible. Sin embargo, Herbrand demostró que si $\mathcal{S}$ es un conjunto estándar de cláusulas (programa lógico), la situación del Ejemplo 38 no puede suceder, es decir, siempre podemos encontrar un $\mathfrak{H}$ -modelo para $\mathcal{S}$.

Teorema 7.

Sea $\mathfrak{P}$ un programa lógico. $\mathfrak{P}$ tiene un modelo sí, y solo sí, $\mathfrak{P}$ tiene un modelo de Herbrand.

La demostración del Teorema 7 se encuentra en forma explícita en Ben-Ari[pág. 138][1] y Geuvers[8]. En Ortíz[pág. 92][29] se prueba usando la definición de $\mathfrak{H}$ -estructura asociada o correspondiente.

Hasta ahora, la teoría de Herbrand que hemos estudiado nos ha permitido establecer una equivalencia semántica entre los modelos y los $\mathfrak{H}$ -modelos de un programa lógico.

Sin embargo, la teoría de Herbrand no la emplearemos como método deductivo, debido a que al aplicar el **Teorema de Herbrand**, que dice:

“ Un conjunto estándar de cláusulas $\mathcal{S}$ es insatisfacible sí, y solo sí, existe un conjunto finito de instancias básicas de cláusulas de $\mathcal{S}$ que es insatisfacible.”

y para demostrar que una cláusula es insatisfacible, el teorema:

“ Una cláusula de $\mathcal{S}$ es insatisfacible sí, y solo sí, la cláusula es interpretada en 0 (falsa) por todas las $\mathfrak{H}$ -estructuras de $\mathcal{S}$. ”

se requiere evaluar todos los posibles conjuntos finitos de instancias básicas de $\mathcal{S}$, conjuntos que pueden ser muy variados y extensos; por todas las posibles $\mathfrak{H}$ -estructuras definidas para $\mathcal{S}$ para probar así que $\mathcal{S}$ es insatisfacible; todo esto hace al método de Herbrand ineficiente comparado con la Resolución-SLD expuesta en la Sección 2.1, que no emplea este tipo de procedimientos, ya que su algoritmo procede con las cláusulas del programa lógico que requiere inmediatamente en cada paso.

La razón de emplear parte de la teoría de Herbrand radica en que el tipo de estructura que define Herbrand ($\mathfrak{H}$ -estructura) permite operar de manera más fácil, ya que están compuestas por elementos básicos del programa lógico, y emulan las interpretaciones clásicas de una cláusula para efectos de insatisfacibilidad.

Nuestro propósito es definir un conjunto que represente todos las cláusulas que son consecuencias lógicas de un programa lógico, resultado que nos permitirá probar la *completitud* de nuestro método deductivo en estudio: la *Resolución-SLD*.

La demostración de los dos teoremas (de Herbrand) que aparecen en esta página y en la anterior, se encuentran en Ben-Ari[pág. 140][1] y Ortíz[29, págs. 92 y 97].

Dado que los $\mathfrak{H}$ -modelos de un programa lógico $\mathfrak{P}$, por ser $\mathfrak{H}$ -estructuras, son subconjuntos de la base de Herbrand de $\mathfrak{P}$; tenemos que, por la Observación 33 (segundo ítem) el conjunto que reúne a todos los $\mathfrak{H}$ -modelos es distinto de vacío y por ser $\mathcal{O}(\mathfrak{B}_{\mathfrak{P}})$ un retículo completo (ver pág. 56) podemos pensar en la existencia de un $\mathfrak{H}$ -modelo mínimo de $\mathfrak{P}$.

Teorema 8. Sea $\mathcal{M} = \{\mathfrak{E}_{\mathfrak{H}_1}, \mathfrak{E}_{\mathfrak{H}_2}, \mathfrak{E}_{\mathfrak{H}_3}, \dots\}$ una familia no vacía de $\mathfrak{H}$ -modelos de un programa lógico *definido* $\mathfrak{P}$. Entonces

$$\mathfrak{E}_{\mathfrak{H}} = \bigcap_{\mathfrak{E}_{\mathfrak{H}_i} \in \mathcal{M}} \mathfrak{E}_{\mathfrak{H}_i}$$

es un $\mathfrak{H}$ -modelo de $\mathfrak{P}$.

Demostración.

Supongamos que $\mathfrak{E}_{\mathfrak{H}}$ no es un $\mathfrak{H}$ -modelo de $\mathfrak{P}$. Entonces, existe una cláusula definida $\mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p$, con $p \geq 0$, de $\mathfrak{P}$, tal que para alguna instancia básica $\mathcal{A}^* \leftarrow \mathcal{B}_1^*, \dots, \mathcal{B}_p^*$ de la cláusula, es *interpretada en 0* por $\mathfrak{E}_{\mathfrak{H}}$.

Lo que implica que $\mathfrak{E}_{\mathfrak{H}} \supseteq \{\mathcal{B}_1^*, \dots, \mathcal{B}_p^*\}$ pero $\mathfrak{E}_{\mathfrak{H}} \not\supseteq \{\mathcal{A}^*\}$. Por tanto, para todo $\mathfrak{E}_{\mathfrak{H}_j} \in \mathcal{M}$ se cumple que $\mathfrak{E}_{\mathfrak{H}_j} \supseteq \{\mathcal{B}_1^*, \dots, \mathcal{B}_p^*\}$ y, para algún $j = q$, existe $\mathfrak{E}_{\mathfrak{H}_q} \in \mathcal{M}$ tal que $\mathcal{A}^* \notin \mathfrak{E}_{\mathfrak{H}_q}$.

Por consiguiente, la instancia básica $\mathcal{A}^* \leftarrow \mathcal{B}_1^*, \dots, \mathcal{B}_p^*$ es *interpretada en 0* por $\mathfrak{E}_{\mathfrak{H}_q}$, lo cual es una contradicción pues $\mathfrak{E}_{\mathfrak{H}_q}$ es un $\mathfrak{H}$ -modelo de $\mathfrak{P}$. ■

Obsérvese que en el Teorema 8, se realiza la prueba para programas lógicos *definidos*, la razón es que para programas lógicos indefinidos, en general la intersección de los $\mathfrak{H}$ -modelos no es un $\mathfrak{H}$ -modelo; veamos el siguiente ejemplo.

EJEMPLO 39. Sea $\mathcal{S}$ el conjunto de sentencias $\{\mathcal{P}(a) \vee \mathcal{Q}(a)\}$, que representa de manera equivalente a la cláusula *indefinida* $\mathcal{P}(a), \mathcal{Q}(a) \leftarrow$. El universo y la base de Herbrand de $\mathcal{S}$ son:

$$\mathcal{U}_{\mathfrak{P}} = \{a\} \quad \text{y} \quad \mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(a), \mathcal{Q}(a)\}.$$

La base de Herbrand sólo tiene dos elementos, entonces las posibles $\mathfrak{H}$ -estructuras de $\mathcal{S}$ son:

$$\mathfrak{E}_{\mathfrak{H}_1} = \emptyset \quad \mathfrak{E}_{\mathfrak{H}_2} = \{\mathcal{P}(a)\} \quad \mathfrak{E}_{\mathfrak{H}_3} = \{\mathcal{Q}(a)\} \quad \mathfrak{E}_{\mathfrak{H}_4} = \mathfrak{B}_{\mathfrak{P}}$$

de las cuales, las únicas $\mathfrak{H}$ -estructuras que son $\mathfrak{H}$ -modelos son:

$$\mathfrak{E}_{\mathfrak{H}_2} = \{\mathcal{P}(a)\} \quad \mathfrak{E}_{\mathfrak{H}_3} = \{\mathcal{Q}(a)\} \quad \mathfrak{E}_{\mathfrak{H}_4} = \mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(a), \mathcal{Q}(a)\}$$

de donde podemos observar que la intersección es igual a vacío, pero vacío no es un $\mathfrak{H}$ -modelo de $\mathcal{S}$. ◆

Definición 2.16. Sea $\mathfrak{P}$ un programa lógico definido. El **modelo mínimo de Herbrand** de $\mathfrak{P}$, se denota por $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$, y se define como la intersección de todos los $\mathfrak{H}$ -modelos de $\mathfrak{P}$, es decir,

$$\mathbf{mm}\mathfrak{H}_{\mathfrak{P}} = \bigcap \{ \mathfrak{E}_{\mathfrak{H}} \mid \mathfrak{E}_{\mathfrak{H}} \text{ es un } \mathfrak{H}\text{-modelo de } \mathfrak{P} \}$$

EJEMPLO 40. Con base en el Ejemplo 37, donde los $\mathfrak{H}$ -modelos de $\mathfrak{P}$ son:

$$\begin{aligned} \mathfrak{E}_{\mathfrak{H}_4} &= \{\mathcal{P}(d, d), \mathcal{R}(d)\} & \mathfrak{E}_{\mathfrak{H}_{14}} &= \{\mathcal{P}(d, d), \mathcal{Q}(d), \mathcal{R}(d)\} \\ \mathfrak{E}_{\mathfrak{H}_{15}} &= \{\mathcal{P}(d, d), \mathcal{R}(d), \mathcal{S}(d, d)\} & \mathfrak{E}_{\mathfrak{H}_{16}} &= \mathfrak{B}_{\mathfrak{P}} \\ & & &= \{\mathcal{P}(d, d), \mathcal{Q}(d), \mathcal{R}(d), \mathcal{S}(d, d)\} \end{aligned}$$

calculamos $\text{mm}\mathfrak{H}_{\mathfrak{P}}$ intersectando todos los $\mathfrak{H}$ -modelos de $\mathfrak{P}$ y se obtiene:

$$\text{mm}\mathfrak{H}_{\mathfrak{P}} = \mathfrak{E}_{\mathfrak{H}_4} \cap \mathfrak{E}_{\mathfrak{H}_{14}} \cap \mathfrak{E}_{\mathfrak{H}_{15}} \cap \mathfrak{E}_{\mathfrak{H}_{16}} = \{\mathcal{P}(d, d), \mathcal{R}(d)\}.$$

◆

A continuación, estableceremos una forma de representar al *modelo mínimo de Herbrand* a partir de la característica declarativa de los $\mathfrak{H}$ -modelos, es decir, a partir la característica que se asigna a cada cláusula al ser *interpretadas en 1* por las $\mathfrak{H}$ -estructuras.

La demostración del siguiente teorema se realiza de la misma forma que lo presentan Emden y Kowalski[pág. 738][43].

Teorema 9.

Sea $\mathfrak{P}$ un programa lógico definido. El *modelo mínimo de Herbrand* de $\mathfrak{P}$, es el conjunto de todos los átomos básicos que son consecuencia lógica de $\mathfrak{P}$, es decir,

$$\text{mm}\mathfrak{H}_{\mathfrak{P}} = \{\mathcal{R} \in \mathfrak{B}_{\mathfrak{P}} \mid \mathfrak{P} \models \mathcal{R}\}$$

donde $\mathcal{R}$ denota un elemento típico de $\mathfrak{B}_{\mathfrak{P}}$.

Demostración.

Sea $\mathcal{R}(t_1, \dots, t_n)$ una instancia básica de un átomo presente en $\mathfrak{P}$. Denotamos $\mathcal{R}(t_1, \dots, t_n)$ por $\mathcal{R}$ para simplicidad en la prueba. Así, $\mathcal{R}$ denota un elemento típico de $\mathfrak{B}_{\mathfrak{P}}$.

$\mathcal{R}$ es una consecuencia lógica de $\mathfrak{P}$

sí, y solo sí, $\mathfrak{P} \cup \{\neg\mathcal{R}\}$ es insatisfacible (por Proposición 1.2)

sí, y solo sí, $\mathfrak{P} \cup \{\neg\mathcal{R}\}$ no tiene modelos (por Definición 1.18 y 1.20)

sí, y solo sí, $\mathfrak{P} \cup \{\neg\mathcal{R}\}$ no tiene $\mathfrak{H}$ -modelos (por Teorema 7)

sí, y solo sí, $\neg\mathcal{R}$ es *interpretado en 0* por todos los $\mathfrak{H}$ -modelos de $\mathfrak{P}$

sí, y solo sí, $\mathcal{R}$ es *interpretado en 1* por todos los $\mathfrak{H}$ -modelos de $\mathfrak{P}$

sí, y solo sí, $\mathcal{R}$ pertenece a $\text{mm}\mathfrak{H}_{\mathfrak{P}}$.

■

2.4. Semántica de Punto Fijo

Como dice Simari[36], la semántica de Punto Fijo se origina del estudio de la conexión existente entre la lógica y el álgebra. Conceptos familiares en álgebra tales como retículos, mapeos y puntos fijos permiten dotar de significado a un programa lógico.

Dado un programa lógico definido, ¿cómo podríamos obtener su modelo mínimo de Herbrand?. Según Celma[pág. 26][2] la definición declarativa (Sección 2.3) de $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$ no nos dice mucho ya que para calcularlo deberíamos encontrar todos los $\mathfrak{H}$ -modelos del programa lógico y hallar su intersección, por esta razón daremos una visión constructiva de $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$.

En esta sección, se define el *operador de consecuencia inmediata* $\mathcal{T}_{\mathfrak{P}}$, se muestra como se caracteriza los *modelos mínimos* de $\mathfrak{P}$ a partir $\mathcal{T}_{\mathfrak{P}}$ y se prueba la equivalencia entre estos objetos. Sólo para que el texto sea autosuficiente, daremos algunas definiciones previas.

Para las siguientes definiciones usaremos terminología básica de la teoría de retículos y se refiere el texto de Rodríguez[35].

Sean $\mathcal{J}$ y $\mathcal{K}$ conjuntos parcialmente ordenados y $\mathcal{T}$ una función definida entre $\mathcal{J}$ y $\mathcal{K}$, es decir, $\mathcal{T} : \mathcal{J} \longrightarrow \mathcal{K}$.

Definición 2.17. $\mathcal{T}$ es una función **monótona** sí, y solo sí, para todo $a, b \in \mathcal{J}$ se tiene que $a \leq_1 b$ implica $\mathcal{T}(a) \leq_2 \mathcal{T}(b)$, donde “ $\leq_1$ ” y “ $\leq_2$ ” denotan el orden parcial definido en $\mathcal{J}$ y $\mathcal{K}$ respectivamente.

Definición 2.18. $\mathcal{T}$ es una función **continua** sí, y solo sí, $\mathcal{T}$ preserva supremos sobre cadenas, es decir, para todo cadena $a_1 \leq a_2 \leq \dots$ en $\mathcal{J}$ se cumple que

$$\mathcal{T}\left(\sup_{i \geq 0} a_i\right) = \sup_{i \geq 0} \mathcal{T}(a_i).$$

La siguiente proposición nos indica que es suficiente cumplir con la propiedad de continuidad en una función para alcanzar su monotonía. Realizamos la prueba tal como lo hace Ramírez[pág. 9][33].

Proposición 2.1.

Si $\mathcal{T}$ es un operador *continuo*, entonces $\mathcal{T}$ es un operador *monótono*.

Demostración.

Sea $\mathcal{T} : \mathcal{J} \longrightarrow \mathcal{K}$ un operador continuo y sean $a, b \in \mathcal{J}$ tal que $a \leq_1 b$. Entonces, $\{a, b\}$ es una cadena. Luego, $\mathcal{T}(b) = \mathcal{T}(\sup\{a, b\}) = \sup\{\mathcal{T}(a), \mathcal{T}(b)\}$, es decir, $\mathcal{T}(a) \leq_2 \mathcal{T}(b)$. ■

Definición 2.19. Sea a un elemento de $\mathcal{L}$. Diremos que a es un **punto fijo** de $\mathcal{T}$ sí, y solo sí, $\mathcal{T}(a) = a$.

Definición 2.20. Sea a un elemento de $\mathcal{L}$. Diremos que a es el **menor punto fijo** de $\mathcal{T}$, y se denota por $\mathbf{mpf}_{\mathcal{T}}$, si para todo otro punto fijo b de $\mathcal{T}$ se tiene que $a \leq b$.

Podemos construir un *modelo mínimo de Herbrand* a través de un proceso iterativo, definiendo una serie de conjuntos formados recursivamente, tal como lo describe Ramírez[33, pág. 8], veamos:

- Las instancias básicas de los *hechos* deben estar en todo $\mathfrak{H}$ -modelo y por tanto, también en $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$. Llamaremos a este conjunto $\mathcal{C}_0$, el cual será un subconjunto de $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$.
- Sea $\mathcal{A}^* \longleftarrow \mathcal{B}_1^*, \dots, \mathcal{B}_p^*$ una instancia básica de una regla $\mathcal{A} \longleftarrow \mathcal{B}_1, \dots, \mathcal{B}_p$ con $p \geq 1$. Por la Definición 2.14 sabemos que si $\mathcal{B}_1^*, \dots, \mathcal{B}_p^*$ pertenecen a $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$ entonces $\mathcal{A}^*$ también tiene que pertenecer. Por tanto, si $\mathcal{B}_1^*, \dots, \mathcal{B}_p^*$ pertenecen a $\mathcal{C}_k$, definimos

$$\mathcal{C}_{k+1} = \{\mathcal{A}^* \mid \mathcal{A}^* \longleftarrow \mathcal{B}_1^*, \dots, \mathcal{B}_p^* \text{ es una instancia básica de } \mathfrak{P} \text{ y } \mathcal{B}_1^*, \dots, \mathcal{B}_p^* \in \mathcal{C}_k\} \cup \mathcal{C}_k$$

y tendremos que $\mathcal{C}_{k+1}$ será un subconjunto de $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$.

A partir de esta idea intuitiva, daremos la definición de una función que se forma a partir de un proceso recursivo aplicando un operador especial que definimos a continuación.

Definición 2.21. Sea $\mathfrak{P}$ un programa lógico definido. Definimos el **operador de consecuencia inmediata** para $\mathfrak{P}$, denotado por $\mathcal{T}_{\mathfrak{P}}$, como la función

$$\mathcal{T}_{\mathfrak{P}} : \wp(\mathfrak{B}_{\mathfrak{P}}) \longrightarrow \wp(\mathfrak{B}_{\mathfrak{P}})$$

$$\mathfrak{E}_{\mathfrak{H}} \longrightarrow \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}})$$

definida por:

$$\mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}}) = \{\mathcal{A} \in \mathfrak{B}_{\mathfrak{P}} \mid \mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p \text{ es una instancia básica de una cláusula en } \mathfrak{P} \text{ y } \{\mathcal{B}_1, \dots, \mathcal{B}_p\} \subseteq \mathfrak{E}_{\mathfrak{H}}\}.$$

EJEMPLO 41. Sea $\mathfrak{P}$ el programa lógico definido formado por las cláusulas:

$$\mathfrak{P} = \left\{ \begin{array}{l} \bullet \mathcal{P}(w) \leftarrow \mathcal{R}(w) \\ \bullet \mathcal{R}(b) \leftarrow \\ \bullet \mathcal{Q}(v) \leftarrow \mathcal{S}(w), \mathcal{R}(v) \\ \bullet \mathcal{S}(v) \leftarrow \mathcal{R}(v) \\ \bullet \mathcal{P}(a) \leftarrow \end{array} \right.$$

El universo y la base de Herbrand de $\mathfrak{P}$ son:

$$\mathfrak{U}_{\mathfrak{P}} = \{a, b\}$$

$$\mathfrak{B}_{\mathfrak{P}} = \{\mathcal{P}(a), \mathcal{P}(b), \mathcal{Q}(a), \mathcal{Q}(b), \mathcal{R}(a), \mathcal{R}(b), \mathcal{S}(a), \mathcal{S}(b)\}$$

Consideraremos algunas de las $\mathfrak{H}$ -estructuras de $\mathfrak{P}$ y aplicaremos para ellas el *operador de consecuencia inmediata*, veamos:

$$\mathfrak{E}_{\mathfrak{H}_1} = \{\mathcal{R}(b)\} \longrightarrow \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}_1}) = \{\mathcal{P}(a), \mathcal{R}(b), \mathcal{S}(b), \mathcal{P}(b)\}$$

$$\mathfrak{E}_{\mathfrak{H}_2} = \{\mathcal{R}(a), \mathcal{P}(a)\} \longrightarrow \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}_2}) = \{\mathcal{P}(a), \mathcal{R}(b), \mathcal{S}(a)\}$$

$$\mathfrak{E}_{\mathfrak{H}_3} = \emptyset \longrightarrow \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}_3}) = \{\mathcal{P}(a), \mathcal{R}(b)\}$$

$$\mathfrak{E}_{\mathfrak{H}_4} = \{\mathcal{S}(b), \mathcal{R}(a)\} \longrightarrow \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}_4}) = \{\mathcal{P}(a), \mathcal{R}(b), \mathcal{Q}(a), \mathcal{S}(a)\}$$

$$\mathfrak{E}_{\mathfrak{H}_5} = \{\mathcal{Q}(a), \mathcal{Q}(b), \mathcal{S}(a), \mathcal{S}(b)\} \longrightarrow \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}_5}) = \{\mathcal{P}(a), \mathcal{R}(b)\}$$

$$\mathfrak{E}_{\mathfrak{H}_6} = \{\mathcal{P}(a), \mathcal{Q}(b), \mathcal{R}(a), \mathcal{S}(a)\} \longrightarrow \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}_6}) = \{\mathcal{P}(a), \mathcal{R}(b), \mathcal{Q}(a), \mathcal{S}(a)\}$$

◆

A continuación describimos dos propiedades del operador de consecuencia inmediata, muy importantes para la construcción de la forma equivalente al modelo mínimo de Herbrand de un programa lógico. Las demostraciones se realizan de la misma manera presentada por Emden-Kowalski[43, pág. 739].

Proposición 2.2.

El *operador de consecuencia inmediata* $\mathcal{T}_{\mathfrak{P}}$ es continuo para cada programa lógico definido $\mathfrak{P}$.

Demostración.

Dado que $\wp(\mathfrak{B}_{\mathfrak{P}})$ es un retículo completo (ver pág. 56), para cada subconjunto $\mathcal{X} (\neq \emptyset)$ de $\wp(\mathfrak{B}_{\mathfrak{P}})$, que por definición es una $\mathfrak{H}$ -estructura, existe el supremo y el ínfimo. Observe que, una $\mathfrak{E}_{\mathfrak{H}} \in \mathcal{X}$ existe y para todo $\{\mathcal{B}_1, \dots, \mathcal{B}_p\}$ tenemos que, $\{\mathcal{B}_1, \dots, \mathcal{B}_p\} \subseteq \mathfrak{E}_{\mathfrak{H}}$ sí, y solo sí, $\{\mathcal{B}_1, \dots, \mathcal{B}_p\} \subseteq \sup(\mathcal{X})$, es decir, todos los elementos que pertenezcan al supremo de $\mathcal{X}$ pertenecen a un subconjunto ($\mathfrak{H}$ -estructura) de $\mathcal{X}$ y viceversa.

Probaremos que, $\mathcal{T}_{\mathfrak{P}}(\sup(\mathcal{X})) = \sup(\mathcal{T}_{\mathfrak{P}}(\mathcal{X}))$, para cada $\mathcal{X} \subseteq \wp(\mathfrak{B}_{\mathfrak{P}})$.

$\mathcal{A} \in \mathcal{T}_{\mathfrak{P}}(\sup(\mathcal{X}))$

sí, y solo sí, $\mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p$ es una instancia básica de una cláusula en $\mathfrak{P}$ y $\{\mathcal{B}_1, \dots, \mathcal{B}_p\} \subseteq \sup(\mathcal{X})$

sí, y solo sí, $\mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p$ es una instancia básica de una cláusula en $\mathfrak{P}$ y $\{\mathcal{B}_1, \dots, \mathcal{B}_p\} \subseteq \mathfrak{E}_{\mathfrak{H}}$, para algún $\mathfrak{E}_{\mathfrak{H}} \in \mathcal{X}$

sí, y solo sí, $\mathcal{A} \in \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}})$, para algún $\mathfrak{E}_{\mathfrak{H}} \in \mathcal{X}$

sí, y solo sí, $\mathcal{A} \in \sup(\mathcal{T}_{\mathfrak{P}}(\mathcal{X}))$. ■

Lema 1.

Sea $\mathfrak{P}$ un programa lógico definido y $\mathfrak{E}_{\mathfrak{H}}$ una $\mathfrak{H}$ -estructura de $\mathfrak{P}$. Entonces $\mathfrak{E}_{\mathfrak{H}}$ es un $\mathfrak{H}$ -modelo de $\mathfrak{P}$ sí, y solo sí, $\mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}}) \subseteq \mathfrak{E}_{\mathfrak{H}}$.

Demostración.

($\Rightarrow$) Sea $\mathcal{A} \in \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}})$. Por definición de $\mathcal{T}_{\mathfrak{P}}$, existe una instancia básica de una cláusula de $\mathfrak{P}$ que tiene como cabeza el átomo $\mathcal{A}$, $\mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p$, tal que $\{\mathcal{B}_1, \dots, \mathcal{B}_p\} \subseteq \mathfrak{E}_{\mathfrak{H}}$. Dado que $\mathfrak{E}_{\mathfrak{H}}$ es un $\mathfrak{H}$ -modelo (por hipótesis), esto implica que $\mathcal{A} \in \mathfrak{E}_{\mathfrak{H}}$, y por tanto, $\mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}}) \subseteq \mathfrak{E}_{\mathfrak{H}}$.

($\Leftarrow$) Supongamos que $\mathfrak{E}_{\mathfrak{H}}$ no es un $\mathfrak{H}$ -modelo de $\mathfrak{P}$. Entonces, existe una instancia básica $\mathcal{A} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_p$ de alguna cláusula de $\mathfrak{P}$ que es *interpretada en 0* por $\mathfrak{E}_{\mathfrak{H}}$. Por consiguiente, $\{\mathcal{B}_1, \dots, \mathcal{B}_p\} \subseteq \mathfrak{E}_{\mathfrak{H}}$ y $\mathcal{A} \notin \mathfrak{E}_{\mathfrak{H}}$, de aquí se tiene que $\mathcal{A} \in \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}})$ y $\mathcal{A} \notin \mathfrak{E}_{\mathfrak{H}}$, pero $\mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}}) \subseteq \mathfrak{E}_{\mathfrak{H}}$ (por hipótesis), con lo cual se llega a una contradicción y por lo tanto, $\mathfrak{E}_{\mathfrak{H}}$ debe ser $\mathfrak{H}$ -modelo de $\mathfrak{P}$. ■

Observación 34. Con base en el Lema 1, para el Ejemplo 41, ninguna de las $\mathfrak{H}$ -estructuras de las seis exhibidas es $\mathfrak{H}$ -modelo de $\mathfrak{P}$.

Tal como se enunció en la página 64, a partir de un proceso recursivo podemos calcular el modelo mínimo de Herbrand. Consideremos la siguiente sucesión (transfinita) creciente de $\mathfrak{H}$ -estructuras, la cual inicia aplicando el operador de consecuencia inmediata al conjunto vacío, y repetimos su aplicación sobre la $\mathfrak{H}$ -estructura obtenida, hasta llegar a un *punto fijo*:

$$\begin{aligned}
\mathcal{T}_{\mathfrak{P}} \uparrow 0 &= \emptyset (= \mathcal{T}_{\mathfrak{P}}^0(\emptyset)) \\
\mathcal{T}_{\mathfrak{P}} \uparrow 1 &= \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow 0) = \mathcal{T}_{\mathfrak{P}}(\emptyset) = \mathcal{T}_{\mathfrak{P}}^1(\emptyset) \\
\mathcal{T}_{\mathfrak{P}} \uparrow 2 &= \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow 1) = \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}}(\emptyset)) = \mathcal{T}_{\mathfrak{P}}^2(\emptyset) \\
\mathcal{T}_{\mathfrak{P}} \uparrow 3 &= \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow 2) = \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}}(\emptyset))) = \mathcal{T}_{\mathfrak{P}}^3(\emptyset) \\
&\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\
\mathcal{T}_{\mathfrak{P}} \uparrow (i+1) &= \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow i) = \underbrace{\mathcal{T}_{\mathfrak{P}}(\cdots(\mathcal{T}_{\mathfrak{P}}(\emptyset)))}_{(i+1)\text{-veces}} = \mathcal{T}_{\mathfrak{P}}^{i+1}(\emptyset) \\
&\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\
\mathcal{T}_{\mathfrak{P}} \uparrow \omega &= \bigcup_{i=0}^{\infty} \mathcal{T}_{\mathfrak{P}} \uparrow i = \sup_{k \in \mathbb{N}} \{ \mathcal{T}_{\mathfrak{P}}^k(\emptyset) \}
\end{aligned}$$

obteniendo la siguiente sucesión creciente:

$$\emptyset \subseteq \mathcal{T}_{\mathfrak{P}}(\emptyset) \subseteq \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}}(\emptyset)) \subseteq \cdots \subseteq \sup_{k \in \mathbb{N}} \{ \mathcal{T}_{\mathfrak{P}}^k(\emptyset) \}$$

o de manera equivalente, empleando la notación estándar:

$$\mathcal{T}_{\mathfrak{P}} \uparrow 0 \subseteq \mathcal{T}_{\mathfrak{P}} \uparrow 1 \subseteq \mathcal{T}_{\mathfrak{P}} \uparrow 2 \subseteq \cdots \subseteq \mathcal{T}_{\mathfrak{P}} \uparrow \omega.$$

Observemos que si para algún i se cumple que $\mathcal{T}_{\mathfrak{P}} \uparrow i = \mathcal{T}_{\mathfrak{P}} \uparrow (i+1)$, se tiene que

$$\underbrace{\mathcal{T}_{\mathfrak{P}} \uparrow i}_w = \mathcal{T}_{\mathfrak{P}} \uparrow (i+1) = \underbrace{\mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow i)}_{\mathcal{T}_{\mathfrak{P}}(w)}$$

entonces $\mathcal{T}_{\mathfrak{P}} \uparrow i$ es un *punto fijo* de $\mathcal{T}_{\mathfrak{P}}$ y se cumple que $\mathcal{T}_{\mathfrak{P}} \uparrow \omega = \mathcal{T}_{\mathfrak{P}} \uparrow i$.

EJEMPLO 42. Con base en el Ejemplo 37, donde $\mathfrak{P}$ es el programa lógico definido

$$\mathfrak{P} = \left\{ \begin{array}{l} \bullet \mathcal{P}(u, v) \longleftarrow \mathcal{Q}(u) \\ \bullet \mathcal{R}(v) \longleftarrow \\ \bullet \mathcal{P}(u, u) \longleftarrow \\ \bullet \mathcal{R}(v) \longleftarrow \mathcal{S}(v, u) \end{array} \right.$$

Entonces,

$$\begin{aligned}\mathcal{T}_{\mathfrak{P}} \uparrow 0 &= \emptyset \\ \mathcal{T}_{\mathfrak{P}} \uparrow 1 &= \{\mathcal{R}(d), \mathcal{P}(d, d)\} \\ \mathcal{T}_{\mathfrak{P}} \uparrow 2 &= \mathcal{T}_{\mathfrak{P}} \uparrow 1 = \mathcal{T}_{\mathfrak{P}} \uparrow \omega\end{aligned}$$

donde, con base en el Ejemplo 40, vemos claramente que $\mathcal{T}_{\mathfrak{P}} \uparrow \omega$ corresponde con $\text{mm}\mathfrak{H}_{\mathfrak{P}}$. ◆

EJEMPLO 43. Con base en el Ejemplo 41, donde $\mathfrak{P}$ es el programa lógico definido

$$\mathfrak{P} = \left\{ \begin{array}{l} \bullet \mathcal{P}(w) \leftarrow \mathcal{R}(w) \\ \bullet \mathcal{R}(b) \leftarrow \\ \bullet \mathcal{Q}(v) \leftarrow \mathcal{S}(w), \mathcal{R}(v) \\ \bullet \mathcal{S}(v) \leftarrow \mathcal{R}(v) \\ \bullet \mathcal{P}(a) \leftarrow \end{array} \right.$$

Entonces,

$$\begin{aligned}\mathcal{T}_{\mathfrak{P}} \uparrow 0 &= \emptyset \\ \mathcal{T}_{\mathfrak{P}} \uparrow 1 &= \{\mathcal{R}(b), \mathcal{P}(a)\} \\ \mathcal{T}_{\mathfrak{P}} \uparrow 2 &= \mathcal{T}_{\mathfrak{P}} \uparrow 1 \cup \{\mathcal{P}(b), \mathcal{S}(b)\} = \{\mathcal{R}(b), \mathcal{P}(a), \mathcal{P}(b), \mathcal{S}(b)\} \\ \mathcal{T}_{\mathfrak{P}} \uparrow 3 &= \mathcal{T}_{\mathfrak{P}} \uparrow 2 \cup \{\mathcal{Q}(b)\} = \{\mathcal{R}(b), \mathcal{P}(a), \mathcal{P}(b), \mathcal{S}(b), \mathcal{Q}(b)\} \\ \mathcal{T}_{\mathfrak{P}} \uparrow 4 &= \mathcal{T}_{\mathfrak{P}} \uparrow 3 = \mathcal{T}_{\mathfrak{P}} \uparrow \omega\end{aligned}$$
◆

EJEMPLO 44. Sea $\mathfrak{P}$ el programa lógico definido formado por la cláusulas:

$$\mathfrak{P} = \left\{ \begin{array}{l} \bullet \mathcal{Q}(a) \leftarrow \\ \bullet \mathcal{Q}(g(g(v))) \leftarrow \mathcal{Q}(v) \end{array} \right.$$

Entonces,

$$\begin{aligned}\mathcal{T}_{\mathfrak{P}} \uparrow 0 &= \emptyset \\ \mathcal{T}_{\mathfrak{P}} \uparrow 1 &= \{\mathcal{Q}(a)\} \\ \mathcal{T}_{\mathfrak{P}} \uparrow 2 &= \mathcal{T}_{\mathfrak{P}} \uparrow 1 \cup \{\mathcal{Q}(g(g(a)))\} = \{\mathcal{Q}(a), \mathcal{Q}(g(g(a)))\} \\ \mathcal{T}_{\mathfrak{P}} \uparrow 3 &= \mathcal{T}_{\mathfrak{P}} \uparrow 2 \cup \{\mathcal{Q}(g^4(a))\} = \{\mathcal{Q}(a), \mathcal{Q}(g^2(a)), \mathcal{Q}(g^4(a))\} \\ \mathcal{T}_{\mathfrak{P}} \uparrow 4 &= \mathcal{T}_{\mathfrak{P}} \uparrow 3 \cup \{\mathcal{Q}(g^6(a))\} = \{\mathcal{Q}(a), \mathcal{Q}(g^2(a)), \mathcal{Q}(g^4(a)), \mathcal{Q}(g^6(a))\} \\ &\vdots \\ \mathcal{T}_{\mathfrak{P}} \uparrow \omega &= \{\mathcal{Q}(g^{2k}(a)) \mid k \in \mathbb{N}_0\}\end{aligned}$$
◆

Ahora, enunciaremos el *Teorema de Kleene* y realizaremos su prueba. Luego, aplicando el resultado al operador de consecuencia inmediata $\mathcal{T}_{\mathfrak{P}}$, podremos mostrar que el modelo mínimo de Herbrand de un programa lógico es precisamente un *punto fijo* especial del operador $\mathcal{T}_{\mathfrak{P}}$.

El resultado de Kleene, es frecuentemente atribuido a Alfred Tarski, pero el enunciado original del *Teorema de punto fijo de Tarski* es acerca de funciones monótonas. La prueba del siguiente lema se realiza de manera análoga a la presentada por Ramírez[33, pág. 10].

Lema 2 (Teorema de Kleene).

Sea $\mathcal{L}$ un retículo completo y $\mathcal{T} : \mathcal{L} \rightarrow \mathcal{L}$ una función continua. Entonces, el *menor punto fijo* de $\mathcal{T}$ es el supremo de la sucesión creciente de $\mathfrak{H}$ -estructuras de $\mathcal{T}$, es decir,

$$\mathbf{mpf}_{\mathcal{T}} = \sup_{k \in \mathbb{N}} \{ \mathcal{T}^k(\emptyset) \}.$$

Demostración.

Iniciamos indicando que la notación $\mathcal{T}^k(\emptyset)$ empleada en el lema está definida por:

- $\mathcal{T}^0(\emptyset) = \emptyset$
- $\mathcal{T}^{k+1}(\emptyset) = \mathcal{T}(\mathcal{T}^k(\emptyset))$

Dado que, para todo $\mathcal{X} \in \mathcal{L}$ se cumple $\emptyset \subseteq \mathcal{X}$, entonces $\mathcal{T}^0(\emptyset) = \emptyset \subseteq \mathcal{T}^1(\emptyset)$; y por la monotonía de $\mathcal{T}$ (Proposición 2.2) se tiene que:

$$\begin{aligned} \text{si } \mathcal{T}^k(\emptyset) \subseteq \mathcal{T}^{k+1}(\emptyset) \text{ entonces } \mathcal{T}^{k+1}(\emptyset) &= \mathcal{T}(\mathcal{T}^k(\emptyset)) \\ &\subseteq \mathcal{T}(\mathcal{T}^{k+1}(\emptyset)) \\ &= \mathcal{T}^{k+2}(\emptyset). \end{aligned}$$

Por lo tanto, por inducción sobre $k \in \mathbb{N}$, se concluye que

$$\mathcal{T}^k(\emptyset) \subseteq \mathcal{T}^{k+1}(\emptyset), \forall k \in \mathbb{N}.$$

Es decir, los elementos $\mathcal{T}^k(\emptyset)$ forman una cadena en $\mathcal{L}$ y como $\mathcal{L}$ es un retículo completo, el $\sup_{k \in \mathbb{N}} \{ \mathcal{T}^k(\emptyset) \}$ existe y tiene sentido usarlo para definir $\mathbf{mpf}_{\mathcal{T}}$.

Ahora, probaremos que $\mathbf{mpf}_{\mathcal{T}} = \sup_{k \in \mathbb{N}} \{ \mathcal{T}^k(\emptyset) \}$ es un punto fijo de $\mathcal{T}$ y luego que ese punto fijo es mínimo, veamos:

$$\begin{aligned}
\mathcal{T}(\text{mpf}_{\mathcal{T}}) &= \mathcal{T}\left(\sup_{k \in \mathbb{N}} \{\mathcal{T}^k(\emptyset)\}\right) \\
&= \sup_{k \in \mathbb{N}} \mathcal{T}(\mathcal{T}^k(\emptyset)) && \text{por continuidad de } \mathcal{T} \\
&= \sup_{k \in \mathbb{N}} \mathcal{T}^{k+1}(\emptyset) && \text{por definición de } \mathcal{T}^k(\emptyset) \\
&= \sup_{k \in \mathbb{N}} \mathcal{T}^k(\emptyset) && \text{pues } \sup_{k \in \mathbb{N}} \mathcal{X}_k = \sup_{k \in \mathbb{N}} \mathcal{X}_{j+k}, \forall j \in \mathbb{N} \\
&= \text{mpf}_{\mathcal{T}}
\end{aligned}$$

De aquí, $\text{mpf}_{\mathcal{T}}$ es realmente un punto fijo de $\mathcal{T}$.

Ahora, supongamos que $\mathcal{Z}$ es un punto fijo de $\mathcal{T}$, así $\mathcal{T}(\mathcal{Z}) = \mathcal{Z}$. Como $\mathcal{T}^0(\emptyset) = \emptyset$, se tiene que $\mathcal{T}^0(\emptyset) \subseteq \mathcal{Z}$.

Como $\mathcal{T}$ es monótona:

$$\begin{array}{rcl}
& \mathcal{T}^0(\emptyset) & \subseteq \mathcal{Z} \\
\mathcal{T}^1(\emptyset) & = & \mathcal{T}(\mathcal{T}^0(\emptyset)) \subseteq \mathcal{T}(\mathcal{Z}) = \mathcal{Z} \\
\mathcal{T}^2(\emptyset) & = & \mathcal{T}(\mathcal{T}^1(\emptyset)) \subseteq \mathcal{T}(\mathcal{Z}) = \mathcal{Z} \\
\mathcal{T}^3(\emptyset) & = & \mathcal{T}(\mathcal{T}^2(\emptyset)) \subseteq \mathcal{T}(\mathcal{Z}) = \mathcal{Z} \\
& \vdots & \vdots \\
\mathcal{T}^{i+1}(\emptyset) & = & \mathcal{T}(\mathcal{T}^i(\emptyset)) \subseteq \mathcal{T}(\mathcal{Z}) = \mathcal{Z} \\
& \vdots & \vdots \\
\sup_{k \in \mathbb{N}} \{\mathcal{T}^{k+1}(\emptyset)\} & = & \sup_{k \in \mathbb{N}} \{\mathcal{T}^k(\emptyset)\} \subseteq \mathcal{T}(\mathcal{Z}) = \mathcal{Z}
\end{array}$$

Por lo tanto,

$$\text{mpf}_{\mathcal{T}} = \sup_{k \in \mathbb{N}} \{\mathcal{T}^k(\emptyset)\} \subseteq \mathcal{Z}$$

■

Aplicando, la Proposición 2.2, el Lema 2 y la notación de la página 67; al operador de consecuencia inmediata $\mathcal{T}_{\mathfrak{P}}$, podemos afirmar que

$$\text{mpf}_{\mathcal{T}_{\mathfrak{P}}} = \mathcal{T}_{\mathfrak{P}} \uparrow \omega.$$

y concluimos que,

“ El menor punto fijo del operador de consecuencia inmediata constituye el significado asociado a un programa lógico de acuerdo a la semántica de Punto Fijo. ”

Finalizamos esta sección, demostrando la equivalencia entre la semántica declarativa y la semántica de punto fijo, es decir, probando que el *modelo mínimo de Herbrand* es igual al *menor punto fijo del operador de consecuencia inmediata* de un programa lógico.

Teorema 10.

Para un programa lógico definido $\mathfrak{P}$, tenemos que

$$\mathbf{mm}\mathfrak{H}_{\mathfrak{P}} = \mathbf{mpf}_{\mathcal{T}_{\mathfrak{P}}} = \mathcal{T}_{\mathfrak{P}} \uparrow \omega$$

Demostración.

$$\begin{aligned} \mathbf{mm}\mathfrak{H}_{\mathfrak{P}} &= \inf \{ \mathfrak{E}_{\mathfrak{H}} \mid \mathfrak{E}_{\mathfrak{H}} \text{ es un } \mathfrak{H}\text{-modelo de } \mathfrak{P} \} && \text{por la Definición 2.16} \\ &= \inf \{ \mathfrak{E}_{\mathfrak{H}} \mid \mathcal{T}_{\mathfrak{P}}(\mathfrak{E}_{\mathfrak{H}}) \subseteq \mathfrak{E}_{\mathfrak{H}} \} && \text{por el Lema 1} \\ &= \mathbf{mpf}_{\mathcal{T}_{\mathfrak{P}}} && \text{por def. del menor punto fijo} \\ &= \mathcal{T}_{\mathfrak{P}} \uparrow \omega && \text{por la continuidad de } \mathcal{T}_{\mathfrak{P}} \end{aligned}$$

■

Este capítulo finaliza con la siguiente sección que emplea todos los conceptos de la lógica de Primer Orden desarrollada en el Capítulo 1 y toda la teoría semántica desarrollada hasta ahora, para demostrar la *validez* y *completez* del Principio de Resolución-SLD.

2.5. Validez y Completez de la Resolución-SLD para programas lógicos

Desde que iniciamos los estudios de este capítulo nos hemos encaminado a desarrollar nuestra teoría sobre un programa lógico definido junto con la negación de una cláusula para probar que esta unión es un conjunto insatisfacible, es decir, mostrar que siempre que podamos encontrar una refutación-SLD de un programa lógico definido junto con un objetivo, se tenga que el objetivo es una consecuencia lógica del programa lógico; o de forma equivalente, que el objetivo pertenece al modelo mínimo de Herbrand.

En otras palabras, queremos probar que los cálculos ejecutados por el algoritmo de resolución-SLD son correctos.

Para todas las enunciados y pruebas de esta sección, $\mathfrak{P}$ denota un programa lógico definido y $\mathcal{O}$ un objetivo, tal y como se dijo en las Definiciones 1.29 y 1.30.

Teorema 11 (Validez de la Resolución-SLD).

Cada *respuesta calculada* para $\mathfrak{P} \cup \{\mathcal{O}\}$ es una *respuesta correcta* para $\mathfrak{P} \cup \{\mathcal{O}\}$.

Demostración.

Sean $\mathcal{O} : \quad \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_q$ el objetivo y θ una respuesta calculada para $\mathfrak{P} \cup \{\mathcal{O}\}$, donde $\theta_1, \dots, \theta_n$ es la secuencia de *umg's* empleados en una refutación-SLD de $\mathfrak{P} \cup \{\mathcal{O}\}$ de longitud n .

Tenemos que probar que $\theta[\mathcal{B}_1 \wedge \dots \wedge \mathcal{B}_q] = \theta_n \dots \theta_2 \theta_1[\mathcal{B}_1 \wedge \dots \wedge \mathcal{B}_q]$ es una consecuencia lógica de $\mathfrak{P}$. Lo haremos aplicando inducción sobre la longitud n de la refutación-SLD.

Paso básico

Supongamos que $n = 1$, esto significa que el objetivo es de la forma $\mathcal{O} : \quad \leftarrow \mathcal{B}$ y existe un hecho (cláusula unitaria positiva), $\mathcal{A} \leftarrow \quad$, en $\mathfrak{P}$ tal que $(\theta = \theta_1)$ $\theta[\mathcal{A}] = \theta[\mathcal{B}]$. Dado que $\theta[\mathcal{A}] \leftarrow \quad$ es una instancia de un hecho de $\mathfrak{P}$, entonces $\theta[\mathcal{A}]$ es una consecuencia lógica de $\mathfrak{P}$. Por consiguiente, $\theta[\mathcal{B}]$ es una consecuencia lógica de $\mathfrak{P}$.

Paso inductivo

Supongamos que el resultado se cumple para respuestas calculadas obtenidas de refutaciones-SLD de longitud $(n - 1)$, donde el objetivo es $\mathcal{O} : \quad \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_q$.

Supongamos que $\theta_1, \dots, \theta_n$ es la secuencia de *umg's* usados en una refutación-SLD de $\mathfrak{P} \cup \{\mathcal{O}\}$ de longitud n .

Sea $\mathcal{C}_1 : \quad \mathcal{A} \leftarrow \mathcal{D}_1, \dots, \mathcal{D}_p$ la primer cláusula de entrada para la refutación-SLD y $\mathcal{B}_m$, $1 \leq m \leq q$, el literal seleccionado en $\mathcal{O}$ en el primer paso de la resolución-SLD, tal que $\theta[\mathcal{A}] = \theta[\mathcal{B}_m]$.

Entonces, por la hipótesis inductiva

$$\theta_n \dots \theta_2 \theta_1 [\mathcal{B}_1 \wedge \dots \wedge \mathcal{B}_{m-1} \wedge \mathcal{D}_1 \wedge \dots \wedge \mathcal{D}_p \wedge \mathcal{B}_{m+1} \wedge \dots \wedge \mathcal{B}_q]$$

es una consecuencia lógica de $\mathfrak{P}$, debido a que solamente son necesarios $n - 1$ pasos para obtener una refutación-SLD. Por consiguiente, $\theta_n \dots \theta_2 \theta_1 [\mathcal{D}_1 \wedge \dots \wedge \mathcal{D}_p]$ es una consecuencia lógica de $\mathfrak{P}$, de aquí que, $\theta_n \dots \theta_2 \theta_1 [\mathcal{A}] = \theta_n \dots \theta_2 \theta_1 [\mathcal{B}_m]$ es una consecuencia lógica de $\mathfrak{P}$. Dado que cada m , $1 \leq m \leq q$, es arbitrario, concluimos que $\theta_n \dots \theta_2 \theta_1 [\mathcal{B}_1 \wedge \dots \wedge \mathcal{B}_q]$ es una consecuencia lógica de $\mathfrak{P}$. ■

Ahora, probaremos parte de la relación existente entre el *conjunto de éxitos* (Definición 2.12) y el *modelo mínimo de Herbrand* (Definición 2.16) de un programa lógico; de hecho, como veremos más adelante, estos dos conjuntos son equivalentes.

Corolario 11.1.

Para un programa lógico definido $\mathfrak{P}$, el *conjunto de éxitos* de $\mathfrak{P}$ está contenido en el *modelo mínimo de Herbrand* de $\mathfrak{P}$, es decir,

$$\begin{aligned} \mathfrak{ES}_{\mathfrak{P}} &= \{ \mathcal{R} \in \mathfrak{B}_{\mathfrak{P}} \mid \mathfrak{P} \cup \{ \neg \mathcal{R} \} \text{ tiene una refutación-SLD} \} \\ &\subseteq \\ \mathfrak{mmH}_{\mathfrak{P}} &= \{ \mathcal{R} \in \mathfrak{B}_{\mathfrak{P}} \mid \mathfrak{P} \models \mathcal{R} \} \end{aligned}$$

donde $\mathcal{R}$ denota un elemento típico de $\mathfrak{B}_{\mathfrak{P}}$.

Demostración.

Sea $\mathcal{R} \in \mathfrak{B}_{\mathfrak{P}}$ y suponga que $\mathfrak{P} \cup \{ \leftarrow \mathcal{R} \}$ tiene una refutación-SLD. Por consiguiente, existe una *respuesta calculada* θ para $\mathfrak{P} \cup \{ \leftarrow \mathcal{R} \}$. Por el Teorema 11, θ es una *respuesta correcta* para $\mathfrak{P} \cup \{ \leftarrow \mathcal{R} \}$; luego, $\theta[\mathcal{R}]$ es una consecuencia lógica de $\mathfrak{P}$. Pero $\mathcal{R}$ es una instancia básica de una cláusula en $\mathfrak{P}$, debido a que $\mathcal{R} \in \mathfrak{B}_{\mathfrak{P}}$, entonces $\theta[\mathcal{R}] = \mathcal{R}$, es decir, $\theta = \epsilon$ (sustitución identidad) y así, $\mathcal{R}$ es una consecuencia lógica de $\mathfrak{P}$, es decir, $\mathfrak{P} \models \mathcal{R}$. ■

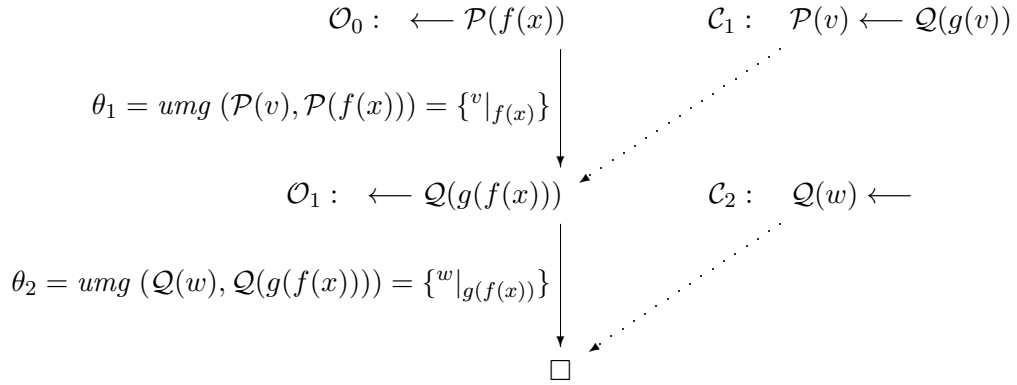
Hasta aquí, hemos probado que el algoritmo de Resolución-SLD definido para programas lógicos es **válido**. Finalizaremos la discusión formal de la Resolución-SLD para programas lógicos mostrando que la forma de operar es **completa**. Antes de esto, definiremos una característica de las refutaciones-SLD; a la cual nos referimos en el lema siguiente, que será de gran importancia para la prueba de completez de la Resolución-SLD.

Definición 2.22. Una **refutación-SLD sin restricciones** de $\mathfrak{P} \cup \{ \mathcal{O} \}$ es una refutación-SLD de $\mathfrak{P} \cup \{ \mathcal{O} \}$, excepto que omitimos el requerimiento de que las sustituciones $\theta_1, \dots, \theta_n$ sean los *unificadores mas generales*. Solo se requiere que ellos sean unificadores.

EJEMPLO 45. Sean $\mathfrak{P}$ un programa lógico formado por:

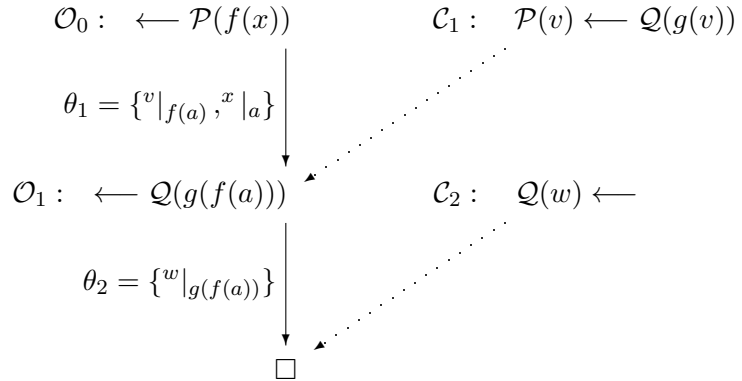
$$\mathfrak{P} = \left\{ \begin{array}{ll} \mathcal{C}_1 : & \mathcal{P}(v) \leftarrow \mathcal{Q}(g(v)) \\ \mathcal{C}_2 : & \mathcal{Q}(w) \end{array} \right.$$

y $\mathcal{O} : \leftarrow \mathcal{P}(f(x))$ un objetivo. Entonces, una representación gráfica de una refutación-SLD de $\mathfrak{P} \cup \{ \mathcal{O} \}$ es:



Luego, por la Definición 2.9, la composición $\theta = \theta_2\theta_1 = \{w|_{g(f(x))}, v|_{f(x)}\}$ restringida a las variables de $\mathcal{O} : \leftarrow \mathcal{P}(f(x))$ es una respuesta *calculada* de $\mathfrak{P} \cup \{\mathcal{O}\}$ y $\theta = \epsilon$, es decir, θ es la sustitución identidad.

Ahora, observemos la representación gráfica de una *refutación-SLD sin restricciones*, donde los unificadores no son necesariamente los *umg*'s:



De aquí, por la Definición 2.4, la composición $\theta = \theta_2\theta_1 = \{w|_{g(f(a))}, v|_{f(a)}, x|_a\}$ restringida a las variables de $\mathcal{O} : \leftarrow \mathcal{P}(f(x))$ es una respuesta *correcta* de $\mathfrak{P} \cup \{\mathcal{O}\}$ y resulta en $\theta = \{x|_a\}$. ◆

Lema 3 (Lema del *umg*).

Sea $\mathfrak{P}$ un programa lógico definido y $\mathcal{O}$ un objetivo. Suponga que $\mathfrak{P} \cup \{\mathcal{O}\}$ tiene una refutación-SLD *sin restricciones* de longitud n , donde $\theta_1, \dots, \theta_n$ son los unificadores empleados (no necesariamente *umg*'s). Entonces, $\mathfrak{P} \cup \{\mathcal{O}\}$ tiene una refutación-SLD de la misma longitud, con los *umg*'s $\theta_1^*, \dots, \theta_n^*$ y existe una sustitución γ , tal que $\theta_n \cdots \theta_2 \theta_1 = \gamma \theta_n^* \cdots \theta_2^* \theta_1^*$.

La prueba del Lema 3 se encuentra en Staab-Schenk[38]. Queremos mostrar que para un programa lógico definido $\mathfrak{P}$ junto con un objetivo $\mathcal{O}$, siempre que $\mathcal{O}$ es una consecuencia lógica de $\mathfrak{P}$, podemos encontrar una refutación-SLD de $\mathfrak{P} \cup \{\mathcal{O}\}$.

Teorema 12 (Completez de la Resolución-SLD).

Sea $\mathfrak{P}$ un programa lógico definido. El *conjunto de éxitos* de $\mathfrak{P}$ es igual al *modelo mínimo de Herbrand* de $\mathfrak{P}$, es decir,

$$\mathfrak{S}\mathfrak{S}_{\mathfrak{P}} = \mathbf{mm}\mathfrak{H}_{\mathfrak{P}}.$$

Demostración.

Por el Corolario 11.1, es suficiente probar para este teorema que $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}} \subseteq \mathfrak{S}\mathfrak{S}_{\mathfrak{P}}$.

Supongamos que $\mathcal{A}$ está en $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$. Por el Teorema 10, $\mathcal{A} \in \mathcal{T}_{\mathfrak{P}} \uparrow n$, para algún $n \in \omega$. Afirmamos que $\mathcal{A} \in \mathcal{T}_{\mathfrak{P}} \uparrow n$ implica que $\mathfrak{P} \cup \{\leftarrow \mathcal{A}\}$ tiene una refutación-SLD, y por tanto, que $\mathcal{A}$ está en $\mathfrak{S}\mathfrak{S}_{\mathfrak{P}}$.

Probaremos esta afirmación aplicando inducción sobre la longitud n de la refutación-SLD.

Paso básico

Supongamos que $n = 1$. Entonces, $\mathcal{A} \in \mathcal{T}_{\mathfrak{P}} \uparrow 1$ significa que $\mathcal{A} \leftarrow$ es una instancia básica de un hecho en $\mathfrak{P}$; luego, por el Algoritmo de Resolución-SLD (ver página 43), $\mathfrak{P} \cup \{\leftarrow \mathcal{A}\}$ tiene una refutación-SLD y así, $\mathcal{A} \in \mathfrak{S}\mathfrak{S}_{\mathfrak{P}}$.

Paso inductivo

Supongamos que la afirmación se cumple para $(n-1)$. Entonces, $\mathcal{A} \in \mathcal{T}_{\mathfrak{P}} \uparrow (n-1)$ implica que $\mathfrak{P} \cup \{\leftarrow \mathcal{A}\}$ tiene una refutación-SLD.

Sea $\mathcal{A} \in \mathcal{T}_{\mathfrak{P}} \uparrow n$. Por la definición de $\mathcal{T}_{\mathfrak{P}}$, existe una instancia básica de una cláusula $\mathcal{B} \leftarrow \mathcal{B}_1, \dots, \mathcal{B}_q$ de $\mathfrak{P}$, tal que $\mathcal{A} = \theta[\mathcal{B}]$ y $\{\theta[\mathcal{B}_1], \dots, \theta[\mathcal{B}_q]\} \subseteq \mathcal{T}_{\mathfrak{P}} \uparrow (n-1)$, para algún θ .

Por la hipótesis inductiva, cada $\mathfrak{P} \cup \{\leftarrow \theta[\mathcal{B}_i]\}$, $1 \leq i \leq q$, tiene una refutación-SLD; y dado que cada $\theta[\mathcal{B}_i]$ es un átomo básico, aquellas refutaciones-SLD pueden ser combinadas en una refutación-SLD *sin restricciones* de $\mathfrak{P} \cup \{\leftarrow \theta[\mathcal{B}_1, \dots, \mathcal{B}_q]\}$ y por consiguiente, $\mathfrak{P} \cup \{\leftarrow \theta[\mathcal{B}]\}$ también tendrá una refutación de este tipo.

Finalmente, dado que $\mathcal{A} = \theta[\mathcal{B}]$, entonces $\mathfrak{P} \cup \{\leftarrow \mathcal{A}\}$ tiene una refutación-SLD *sin restricciones* y aplicando el Lema del *umg* (pág. 74), obtenemos una refutación-SLD de $\mathfrak{P} \cup \{\leftarrow \mathcal{A}\}$. ■

Capítulo 3

Lógicas anotadas

Hasta este momento, desarrollamos en los dos capítulos anteriores toda la teoría necesaria para demostrar que el Principio de Resolución-SLD en lógicas de Primer Orden es un método deductivo *válido* y *completo*, tal como lo exhibimos en la Sección 2.5.

Ahora, nuestro propósito es relacionar y desarrollar toda esta teoría en un tipo de lógica diferente a la estudiada, ya que haremos uso de una anotación adicional para la elaboración de fórmulas; a estas lógicas se les conoce como: “*lógicas anotadas*”.

Las ***lógicas anotadas*** son lógicas no-clásicas que han contribuido fuertemente al desarrollo de la programación lógica. Esto se debe a que el conjunto de valores que puede tomar la interpretación de sus términos no se reduce al clásico $\{0, 1\}$ que bien conocemos, si no que lo enriquece tomando sus valores sobre un retículo (conjunto parcialmente ordenado), dándole más expresividad a las deducciones realizadas.

Siendo su gestor Subrahmanian[16], quien constituye estas lógicas luego de realizar ciertas modificaciones a la teoría de lógicas cuantitativas expuesta inicialmente por Van Emden[42] en su artículo “*Quantitative deduction and its fixpoint theory*”. Él introduce un nuevo término; la fórmula anotada (ó átomo anotado) que consiste de una fórmula junto con un valor de interpretación tomado del intervalo $[0, 1]$. En general, uno podría anotar con cualquier tipo de conjunto con la estructura que se nos ocurra.

Buscando una presentación que sirviera de base para la teoría de lógicas anotadas, se presentaron las lógicas P_τ ; haciendo referencia por P a la lógica Proposicional junto con las anotaciones en un conjunto τ con estructura de retículo. Un estudio detallado de las lógicas P_τ se encuentra en Muñoz[28].

Sin embargo, como se describió en el Capítulo 1, nuestro avance en la teoría se desarrolló en lógicas de Primer Orden, por ser esta una extensión, con más

expresividad que la lógica Proposicional; por esta razón la terminología de lógicas anotadas enunciada en este capítulo será adecuada a lógicas de Primer Orden, justificando así la gran relevancia del trabajo presentado con anterioridad. A este tipo de lógicas se le conocen como lógicas Q_τ (ver Ortíz[30]) y serán presentadas en la siguiente sección.

3.1. Lógicas anotadas Q_τ

El sistema Q_τ es la lógica anotada básica de Primer Orden, debido a que todo su desarrollo se realiza en un lenguaje de primer orden, e igual que para las lógicas P_τ , se considera las anotaciones en un conjunto con estructura de retículo.

Con base al desarrollo teórico de la lógica de Primer Orden se forma la teoría para Q_τ , con un hecho muy importante la adjunción de un conjunto para las anotaciones y una función (negación) que opere con ellas.

Consideramos un retículo completo τ fijo, y una función $\sim : \tau \rightarrow \tau$ también fija, que representa una negación para anotaciones. Tanto el retículo como la función son en principio absolutamente arbitrarios. De esta manera, estaremos hablando de infinitos sistemas lógicos, puesto que cada sistema depende del retículo τ y la función $\sim$.

Definición 3.1. Un *alfabeto anotado*, $\mathcal{A}$, consiste de los mismos elementos de un *alfabeto de Primer Orden*, es decir, variables, conectivos, cuantificadores y símbolos auxiliares.

Observación 35. Tal como lo indica Komendantskaya[18], se permiten varios conectivos para representar operaciones sobre el retículo, a saber, seis conectivos $\wedge, \vee, \otimes, \oplus, \sim, \neg$ y cuatro cuantificadores $\exists, \forall, \Pi, \Sigma$.

Definición 3.2. El *lenguaje de Primer Orden anotado* o simplemente *lenguaje anotado*, $\mathcal{L}$ de Q_τ , es el lenguaje de Primer Orden junto con las constantes de anotación $\lambda, \kappa, \omega, \dots$, pertenecientes a τ .

Observación 36. La definición de término se emplea con el significado convencional del lenguaje de Primer Orden.

Definición 3.3. Si $\mathcal{R}$ es un símbolo de relación de aridad n y $\xi \in \tau$ entonces, el par $\mathcal{R}_{\langle \xi \rangle}$ lo llamaremos *relación anotada*.

Definición 3.4. Si $\mathcal{R}_{\langle \xi \rangle}$ es una relación anotada de aridad n y $t_1, \dots, t_n$ son términos de $\mathcal{L}$ entonces, $\mathcal{R}_{\langle \xi \rangle}(t_1, \dots, t_n)$ será llamado *átomo anotado*.

Las fórmulas para un lenguaje anotado son formadas exactamente como se forman para el lenguaje de Primer Orden, reemplazando únicamente la definición de átomo por la de átomo anotado, veamos:

Definición 3.5. Una *fórmula anotada* se define inductivamente como:

- Un átomo anotado $\mathcal{R}_{\langle \xi \rangle}(t_1, \dots, t_n)$ es una *fórmula anotada*.
- Si Φ y Ψ son átomos anotados, entonces así lo son $(\Phi \vee \Psi), (\Phi \wedge \Psi), (\Phi \oplus \Psi), (\Phi \otimes \Psi), (\sim \Phi)$ y $(\neg \Phi)$.
- Si ϕ y ψ son fórmulas (de primer orden) y $\langle \xi \rangle$ es un término de anotación, entonces $(\phi \vee \psi)_{\langle \xi \rangle}, (\phi \wedge \psi)_{\langle \xi \rangle}, (\phi \oplus \psi)_{\langle \xi \rangle}, (\phi \otimes \psi)_{\langle \xi \rangle}$ son átomos anotados.
- Si Ψ es un átomo anotado y x es un símbolo de variable, entonces las fórmulas cuantificadas $(\forall x \Psi), (\exists x \Psi), (\Pi x \Psi), (\Sigma x \Psi)$ son átomos anotados.

Como podemos observar el desarrollo de la teoría de las lógicas anotadas dependen del conjunto de anotación. En cumplimiento con el objetivo de este trabajo, de probar la validez y completez de la Resolución-SLD sobre birretículos, consideramos apropiado describir la estructura del birretículo en la siguiente sección, para luego dar paso a las definiciones de la lógica anotada exactamente sobre el conjunto de anotaciones de nuestro interés.

3.2. Birretículos ($\mathcal{B}$)

En esta sección se hablará solamente de los conceptos básicos y algunas de las propiedades de los *birretículos*. Un desarrollo amplio de la teoría de birretículos se encuentra en Fitting[4].

Definición 3.6. Un *birretículo* $\mathcal{B}$ es una sextupla $(\mathcal{B}, \vee, \wedge, \oplus, \otimes, \neg)$ tal que $(\mathcal{B}, \vee, \wedge)$ y $(\mathcal{B}, \oplus, \otimes)$ son ambos retículos completos; y $\neg : \mathcal{B} \rightarrow \mathcal{B}$ es una función que satisface las siguientes tres propiedades:

- $\neg \circ \neg = Id_{\mathcal{B}}$,
- $\neg$ es un homomorfismo de $(\mathcal{B}, \vee, \wedge)$ en $(\mathcal{B}, \wedge, \vee)$, y
- $\neg$ es un automorfismo de $(\mathcal{B}, \oplus, \otimes)$.

Observación 37. *La definición de homomorfismo u homomorfismo de retículo se encuentra en Rodríguez[35].*

Observación 38. *La función $\neg : \mathcal{B} \rightarrow \mathcal{B}$ definida por $\neg \langle \mu, \nu \rangle = \langle \nu, \mu \rangle$ satisface las condiciones de la Definición 3.6. A la función $\neg$ la llamaremos **negación**.*

Cierto tipo de birretículos puede ser obtenido formando un producto de dos retículos y esta propiedad hace más fácil describir e investigar sus estructuras.

Para las siguientes definiciones, suponga que $\mathcal{L}_1 = (L_1, \leq_1)$ y $\mathcal{L}_2 = (L_2, \leq_2)$ son retículos completos, donde L_1 y L_2 representan dos conjuntos no vacíos. Sean μ_1, μ_2 elementos arbitrarios del retículo $\mathcal{L}_1$ y sean ν_1, ν_2 elementos arbitrarios del retículo $\mathcal{L}_2$. Además, sean $\cap_1, \cup_1$ el sup y el ínf definido respectivamente en el retículo $\mathcal{L}_1$; y sean $\cap_2, \cup_2$ el sup y el ínf definido respectivamente en el retículo $\mathcal{L}_2$.

Definición 3.7. Se forma el conjunto de puntos $L_1 \times L_2$ y se define, el **orden de verdad** $\leq_v$ y el **orden de conocimiento** $\leq_c$ sobre $L_1 \times L_2$ como sigue:

- $\langle \mu_1, \nu_1 \rangle \leq_v \langle \mu_2, \nu_2 \rangle$ si y solo si $\mu_1 \leq_1 \mu_2$ y $\nu_2 \leq_2 \nu_1$.
- $\langle \mu_1, \nu_1 \rangle \leq_c \langle \mu_2, \nu_2 \rangle$ si y solo si $\mu_1 \leq_1 \mu_2$ y $\nu_1 \leq_2 \nu_2$.

Se denota la estructura resultante por $\mathcal{L}_1 \odot \mathcal{L}_2 = (L_1 \times L_2, \leq_v, \leq_c)$.

En la Definición 3.7, convenimos en denotar como $\langle \mu, \nu \rangle$ a los elementos de $L_1 \times L_2$, notación que seguiremos empleando a lo largo de este trabajo, ya que, nuestro propósito ahora es mostrar que un birretículo se puede formar por el producto de dos retículos.

Habiendo definido $\mathcal{L}_1 \odot \mathcal{L}_2$, se define las operaciones entre birretículos como sigue.

Definición 3.8. Las cuatro **operaciones entre birretículos** asociadas con los ordenes establecidos son $\wedge, \vee, \otimes, \oplus$ y se definen de la siguiente manera:

- Para el ordenamiento $\leq_v$
 - * $\langle \mu_1, \nu_1 \rangle \wedge \langle \mu_2, \nu_2 \rangle = \langle \mu_1 \cap_1 \mu_2, \nu_1 \cup_2 \nu_2 \rangle,$
 - * $\langle \mu_1, \nu_1 \rangle \vee \langle \mu_2, \nu_2 \rangle = \langle \mu_1 \cup_1 \mu_2, \nu_1 \cap_2 \nu_2 \rangle,$
- Para el ordenamiento $\leq_c$
 - * $\langle \mu_1, \nu_1 \rangle \otimes \langle \mu_2, \nu_2 \rangle = \langle \mu_1 \cap_1 \mu_2, \nu_1 \cap_2 \nu_2 \rangle,$
 - * $\langle \mu_1, \nu_1 \rangle \oplus \langle \mu_2, \nu_2 \rangle = \langle \mu_1 \cup_1 \mu_2, \nu_1 \cup_2 \nu_2 \rangle.$

Las operaciones infinitas con respecto al v - y c - ordenamientos, que consisten simplemente de la aplicación iterativa de la operación, se denotan respectivamente por $\bigwedge, \bigvee$ y $\bigotimes, \bigoplus$.

Proposición 3.1. Suponga que $\mathcal{B}$ es un birretículo distributivo. Entonces existen retículos distributivos $\mathcal{L}_1$ y $\mathcal{L}_2$ tal que $\mathcal{B}$ es isomorfo a $\mathcal{L}_1 \odot \mathcal{L}_2$.

El resultado de esta proposición, es bastante significativo, ya que nos garantiza que todos los avances alcanzados para retículos se pueden emplear ahora como birretículo con la estructura definida por $\mathcal{L}_1 \odot \mathcal{L}_2$. A continuación damos un breve ejemplo de un birretículo formado a partir de dos retículos y una representación gráfica del mismo.

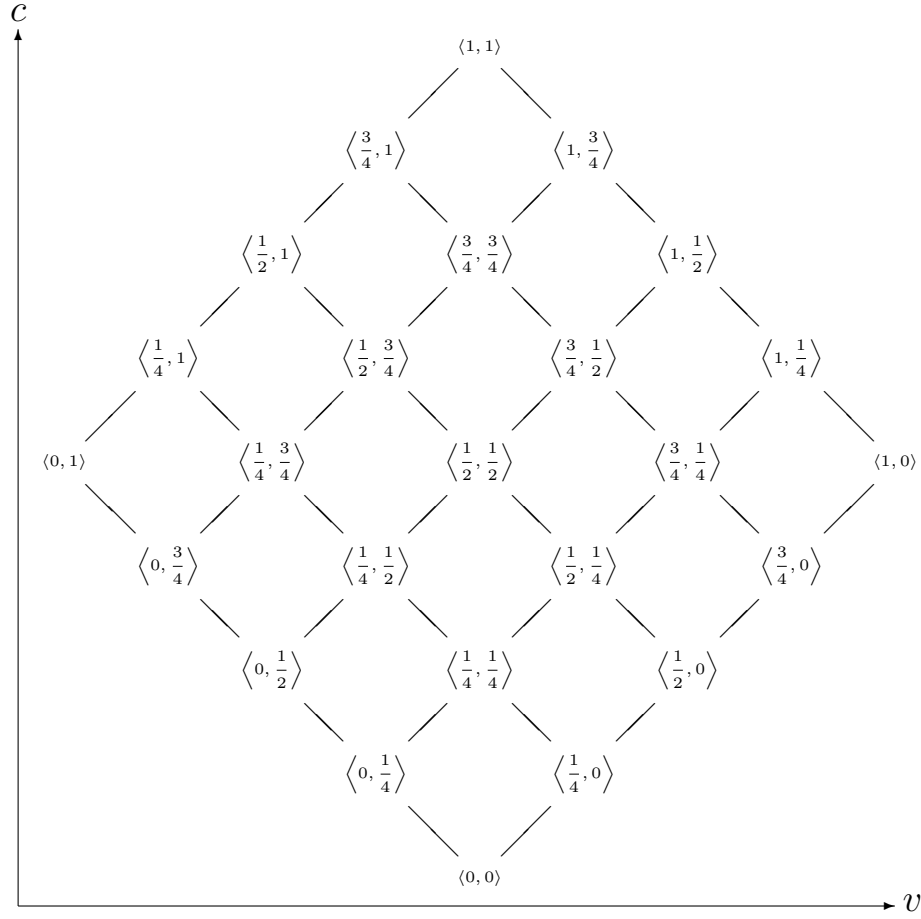
EJEMPLO 46. Considere el birretículo $\mathcal{B}_{25} = \mathcal{L}_1 \odot \mathcal{L}_2$, definido por

$$\mathcal{L}_1 = (\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}, \leq) = \mathcal{L}_2$$

con $0 \leq \frac{1}{4} \leq \frac{1}{2} \leq \frac{3}{4} \leq 1$. Los dos ordenes $\leq_c$ y $\leq_v$ son determinados de acuerdo a la Definición 3.7 para $\mathcal{L}_1 \odot \mathcal{L}_2$ y su representación es la Gráfica 3.1.



Observación 39. En general, el análogo de los valores clásicos de verdad, 1 (verdadero) y 0 (falso), en nuestro conjunto de valores de verdad son representados por $\langle 1, 0 \rangle$ y $\langle 0, 1 \rangle$ respectivamente, que son el mayor y el menor de los elementos del birretículo respecto a $\leq_v$.

Gráfica 3.1. Birretículo $\mathcal{B}_{25}$ (Ejemplo 46)

3.3. Lógica anotada sobre el birretículo $\mathcal{B}$

De acuerdo con el comentario de la página 78, en esta sección haremos la descripción de las definiciones restantes de la lógica anotada, precisamente con sus anotaciones sobre birretículos.

Para todas las definiciones de esta sección, y en adelante para este texto, $\mathcal{B}$ denota un birretículo.

Definición 3.9. Un *término de anotación* de $\mathcal{B}$, se forma únicamente de la siguiente manera: ó es un elemento (constante) de $\mathcal{B}$, ó es una variable de la anotación (que toma valores en $\mathcal{B}$), ó tiene la forma $\zeta(\langle \mu_1, \nu_1 \rangle, \dots, \langle \mu_n, \nu_n \rangle)$ donde $\zeta : \mathcal{B}^n \rightarrow \mathcal{B}$ es una *función de anotación* y los $\langle \mu_i, \nu_i \rangle$, $i = 1, \dots, n$, son anotaciones variables y/o constantes.

Definición 3.10. Una *fórmula anotada* $\mathfrak{F}$ en $\mathcal{B}$ se define inductivamente como:

- Si $\mathcal{R}$ es un símbolo de relación n -ario, $t_1, \dots, t_n$ son términos y $\langle \mu, \nu \rangle$ es un término de anotación de $\mathcal{B}$, entonces $R_{\langle \mu, \nu \rangle}(t_1, \dots, t_n)$ es una *fórmula anotada en $\mathcal{B}$* (llamada un *átomo anotado*).
- Si Φ y Ψ son átomos anotados en $\mathcal{B}$, entonces así lo son $(\Phi \vee \Psi)$, $(\Phi \wedge \Psi)$, $(\Phi \oplus \Psi)$, $(\Phi \otimes \Psi)$, $(\sim \Phi)$ y $(\neg \Phi)$.
- Si ϕ y ψ son fórmulas (de primer orden) y $\langle \mu, \nu \rangle$ es un término de anotación de $\mathcal{B}$, entonces $(\phi \vee \psi)_{\langle \mu, \nu \rangle}$, $(\phi \wedge \psi)_{\langle \mu, \nu \rangle}$, $(\phi \oplus \psi)_{\langle \mu, \nu \rangle}$, $(\phi \otimes \psi)_{\langle \mu, \nu \rangle}$ son átomos anotados en $\mathcal{B}$.
- Si Ψ es un átomo anotado en $\mathcal{B}$ y v es un símbolo de variable, entonces las fórmulas cuantificadas $(\forall v \Psi)$, $(\exists v \Psi)$, $(\Pi v \Psi)$, $(\Sigma v \Psi)$ son átomos anotados en $\mathcal{B}$.

Observación 40. Para simplicidad de la notación se denota arbitrariamente por $R_{\langle \mu, \nu \rangle}$ al átomo anotado $R_{\langle \mu, \nu \rangle}(t_1, \dots, t_n)$.

Definición 3.11. Un *lenguaje anotado en birretículos* $\mathfrak{L}$ sobre $\mathcal{B}$ consiste en el conjunto de todas las *fórmulas anotadas* en $\mathcal{B}$ construidas a partir de los símbolos del alfabeto anotado y los términos de anotación en $\mathcal{B}$.

3.4. Semántica anotada en $\mathcal{B}$

Sea $\mathcal{D}$ un dominio de interpretación para un lenguaje anotado $\mathfrak{L}$ sobre el birretículo $\mathcal{B}$.

Definición 3.12. Una *estructura anotada en $\mathcal{B}$* para $\mathfrak{L}$ es un par $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{I} \rangle$ donde $\mathcal{D}$ es el *dominio de interpretación* de $\mathfrak{L}$, y $\mathfrak{I}$ es una *interpretación anotada en $\mathcal{B}$* para $\mathfrak{L}$ que consiste de las siguientes dos aplicaciones:

- La primer aplicación $\mathfrak{I}$, que interpreta los elementos del lenguaje de primer orden que conforma el lenguaje anotado $\mathfrak{L}$, asocia:
 - * a cada símbolo constante c , un elemento $[c]^{\mathfrak{I}}$ en $\mathcal{D}$.
 - * a cada símbolo de función n -ario f , una función $[f]^{\mathfrak{I}} : \mathcal{D}^n \longrightarrow \mathcal{D}$, y
 - * a cada símbolo de relación n -ario R , una función $[R]^{\mathfrak{I}} : \mathcal{D}^n \longrightarrow \mathcal{B}$.

- La segunda aplicación $\mathfrak{X}$ asocia a cada término de anotación $\langle \mu, \nu \rangle$ de $\mathcal{B}$, una función $\mathfrak{X}_{\langle \mu, \nu \rangle} : \mathcal{B} \longrightarrow \{\langle 0, 1 \rangle, \langle 1, 0 \rangle\}$, definida por

$$\mathfrak{X}_{\langle \mu, \nu \rangle}(\langle \alpha, \beta \rangle) = \begin{cases} \langle 1, 0 \rangle, & \text{si } \langle \mu, \nu \rangle \leq_c \langle \alpha, \beta \rangle, \\ \langle 0, 1 \rangle, & \text{en otro caso.} \end{cases}$$

para todo $\langle \alpha, \beta \rangle \in \mathcal{B}$.

Observación 41. *La aplicación $\mathfrak{X}$ es usada para evaluar la interpretación de la fórmula.*

La *asignación* $\mathfrak{A}$ es definida de la forma convencional usada en lógica de primer orden.

Por simplicidad de la notación, se denota por $\mathfrak{E}$ a la estructura $\mathfrak{E} = \langle \mathcal{D}, \mathfrak{T} \rangle$.

Observación 42. *Nuevamente, gracias al Principio de Recursión, se garantiza la asignación de un único valor en $\mathcal{D}$ a cada término del lenguaje $\mathfrak{L}$.*

Definición 3.13. Sean $\mathfrak{E}$ una estructura anotada en $\mathcal{B}$ y $\mathfrak{A}$ una asignación para $\mathfrak{L}$. Entonces, a cada término t de $\mathfrak{L}$ se asocia el valor $\mathfrak{T}(t) = [t]^{\mathfrak{T}, \mathfrak{A}}$ en $\mathcal{D}$ como sigue:

- para un símbolo constante c , $\mathfrak{T}(c) = [c]^{\mathfrak{T}, \mathfrak{A}} = [c]^{\mathfrak{T}}$.
- para una variable x , $\mathfrak{T}(x) = [x]^{\mathfrak{T}, \mathfrak{A}} = [x]^{\mathfrak{A}}$.
- para un símbolo de función n -ario f , $\mathfrak{T}(f(t_1, \dots, t_n)) = [f(t_1, \dots, t_n)]^{\mathfrak{T}, \mathfrak{A}} = [f]^{\mathfrak{T}}([t_1]^{\mathfrak{T}, \mathfrak{A}}, \dots, [t_n]^{\mathfrak{T}, \mathfrak{A}})$, donde $t_1, \dots, t_n$ son términos de $\mathfrak{L}$.

Definición 3.14. Sean $\mathfrak{E}$ y $\mathfrak{A}$ como antes, entonces a cada fórmula anotada $\mathfrak{F}$ de $\mathcal{L}$, se puede dar un valor de verdad $\mathfrak{T}(\mathfrak{F})$ en $\{\langle 0, 1 \rangle, \langle 1, 0 \rangle\}$ como sigue:

- Si $\mathfrak{F}$ es un átomo anotado $R_{\langle \mu, \nu \rangle}(t_1, \dots, t_n)$, entonces su valor de verdad $\mathfrak{T}(\mathfrak{F})$ esta dado por

$$\begin{aligned} * \quad & \mathfrak{T}(R_{\langle \mu, \nu \rangle}(t_1, \dots, t_n)) = \mathfrak{X}_{\langle \mu, \nu \rangle}([R]^{\mathfrak{T}}([t_1]^{\mathfrak{T}, \mathfrak{A}}, \dots, [t_n]^{\mathfrak{T}, \mathfrak{A}})). \\ * \quad & \mathfrak{T}(\neg R_{\langle \mu, \nu \rangle}) = \mathfrak{T}(R_{\neg \langle \mu, \nu \rangle}) = \mathfrak{T}(R_{\langle \nu, \mu \rangle}). \\ * \quad & \mathfrak{T}(\neg(R_{\langle \mu, \nu \rangle})) = \sim \mathfrak{T}(R_{\langle \mu, \nu \rangle}). \end{aligned}$$

- Si $\mathfrak{F}$ tiene la forma $(\Phi \circledast \Psi)$, donde Φ y Ψ son átomos anotados, entonces $\mathfrak{I}(\Phi \circledast \Psi) = \mathfrak{I}(\Phi) \circledast \mathfrak{I}(\Psi)$.
- Si $\mathfrak{F}$ tiene la forma $(\phi \circledast \psi)_{\langle \mu, \nu \rangle}$, donde ϕ y ψ son fórmulas de primer orden en $\mathfrak{L}$, entonces $\mathfrak{I}((\phi \circledast \psi)_{\langle \mu, \nu \rangle}) = \mathfrak{X}_{\langle \mu, \nu \rangle}(\mathfrak{I}(\phi) \circledast \mathfrak{I}(\psi))$.
- Si $\mathfrak{F}$ tiene la forma $\Pi x R_{\langle \mu, \nu \rangle}(x)$, entonces

$$\mathfrak{I}(\Pi x R_{\langle \mu, \nu \rangle}(x)) = \mathfrak{X}_{\langle \mu, \nu \rangle} \left(\prod_{d \in \mathcal{D}} [R]^{\mathfrak{I}}(d) \right)$$

Observación 43. Π denota un cuantificador en $\mathfrak{L}$ y $\prod$ denota la operación infinita en $\mathcal{B}$, correspondiente a Π respecto a uno de los ordenes establecidos, ya sean $\vee, \wedge, \bigvee, \bigwedge$ ó $\oplus, \otimes, \bigoplus, \bigotimes$. Además, $[R]^{\mathfrak{I}}(d)$ recibe la interpretación con respecto a $\mathfrak{I}$ y $d = x^{\mathfrak{A}}$. Note que al lado izquierdo de los dos items anteriores, el símbolo $\circledast$ denota un conectivo binario en $\mathfrak{L}$, y al lado derecho denota una operación del birretículo $\mathcal{B}$.

En el primer subitem de la Definición 3.14; si $\langle \alpha, \beta \rangle$, un valor constante de $\mathcal{B}$, es el valor de verdad asignado a $R_{\langle \mu, \nu \rangle}(t_1, \dots, t_n)$ por $\mathfrak{I}$ y $\mathfrak{A}$, entonces $\mathfrak{I}(\mathfrak{F}) = \mathfrak{X}_{\langle \mu, \nu \rangle}(\langle \alpha, \beta \rangle) = \langle 1, 0 \rangle$ si $\langle \mu, \nu \rangle \leq_c \langle \alpha, \beta \rangle$, y $\mathfrak{I}(\mathfrak{F}) = \langle 0, 1 \rangle$ en otro caso. En segundo item, se debe tener cuidado a los dos sentidos en que es usado $\neg$. En un caso se emplea de manera convencional sobre el predicado de la fórmula anotada, es decir, como un conectivo unario de $\mathfrak{L}$ y en otro como la función definida para $\mathcal{B}$, es decir, como un homomorfismo de $\mathcal{B}$ en $\mathcal{B}$.

Observación 44. Los conectivos $\wedge, \vee, \otimes, \oplus$ y los cuantificadores $\exists, \forall, \Sigma, \Pi$ del lenguaje anotado $\mathfrak{L}$ son puestos en correspondencia con las operaciones finitas e infinitas definidas en el birretículo $\mathcal{B}$, respectivamente.

Definición 3.15. Sean $\mathfrak{E}$ y $\mathfrak{A}$ como antes y x una variable arbitraria del lenguaje de primer orden $\mathcal{L}$ asociado a $\mathfrak{L}$. Se interpretan los cuantificadores de $\mathfrak{L}$ relacionados con las operaciones infinitas de $\mathcal{B}$, de la siguiente forma:

- $\mathfrak{I}(\exists x R_{\langle \mu, \nu \rangle}(x)) = \mathfrak{X}_{\langle \mu, \nu \rangle} \left(\bigvee_{d \in \mathcal{D}} [R]^{\mathfrak{I}}(d) \right)$
- $\mathfrak{I}(\forall x R_{\langle \mu, \nu \rangle}(x)) = \mathfrak{X}_{\langle \mu, \nu \rangle} \left(\bigwedge_{d \in \mathcal{D}} [R]^{\mathfrak{I}}(d) \right)$

- $\mathfrak{I}(\Pi x R_{\langle \mu, \nu \rangle}(x)) = \mathfrak{X}_{\langle \mu, \nu \rangle} \left(\bigotimes_{d \in \mathcal{D}} [R]^{\mathfrak{I}}(d) \right)$
- $\mathfrak{I}(\Sigma x R_{\langle \mu, \nu \rangle}(x)) = \mathfrak{X}_{\langle \mu, \nu \rangle} \left(\bigoplus_{d \in \mathcal{D}} [R]^{\mathfrak{I}}(d) \right)$

donde $R_{\langle \mu, \nu \rangle}(x)$ es un átomo anotado de $\mathcal{L}$ y $d = [x]^{\mathfrak{I}}$.

Observación 45. *Tal como se indicó en la Observación 39, las definiciones clásicas de fórmulas satisfacible, insatisfacible, valido y no-valido, son estándar si los valores de verdad 1 y 0 en estas definiciones son reemplazadas por $\langle 1, 0 \rangle$ y $\langle 0, 1 \rangle$.*

Ahora, presentaremos un conjunto de propiedades semánticas de las fórmulas anotadas que involucran las operaciones entre birretículos (ver Definición 3.8); propiedades que serán de gran importancia para desarrollar las pruebas de validez y completez del Principio de Resolución-SLD.

Proposición 3.2. Sean $F, F_1, \dots, F_p$ fórmulas de primer orden, y fijemos una interpretación de primer orden $\mathfrak{I}$ para ellos. Entonces cualquier interpretación $\mathfrak{I}$ construida sobre $\mathfrak{I}$ como en la definición 3.12 tiene las siguientes propiedades:

- i) Si $\mathfrak{I}(F_{\langle \mu, \nu \rangle}) = \langle 1, 0 \rangle$, entonces $\mathfrak{I}(F_{\langle \mu^*, \nu^* \rangle}) = \langle 1, 0 \rangle$ para todo $\langle \mu^*, \nu^* \rangle \leq_c \langle \mu, \nu \rangle$.
- ii) $\mathfrak{I}(F_{1\langle \mu_1, \nu_1 \rangle} \otimes \dots \otimes F_{p\langle \mu_p, \nu_p \rangle}) = \langle 1, 0 \rangle$ sí, y solo sí,
 $\mathfrak{I}(F_{1\langle \mu_1, \nu_1 \rangle} \oplus \dots \oplus F_{p\langle \mu_p, \nu_p \rangle}) = \langle 1, 0 \rangle$ sí, y solo sí,
 $\mathfrak{I}(F_{1\langle \mu_1, \nu_1 \rangle} \wedge \dots \wedge F_{p\langle \mu_p, \nu_p \rangle}) = \langle 1, 0 \rangle$ sí, y solo sí,
 $\mathfrak{I}(F_{i\langle \mu_i, \nu_i \rangle}) = \langle 1, 0 \rangle$ para todo $i \in \{1, \dots, p\}$.
- iii) Si $\mathfrak{I}(F_{1\langle \mu_1, \nu_1 \rangle} \odot \dots \odot F_{p\langle \mu_p, \nu_p \rangle}) = \langle 1, 0 \rangle$, entonces

$$\mathfrak{I} \left((F_1 \odot \dots \odot F_p)_{\langle \mu_1, \nu_1 \rangle \odot \dots \odot \langle \mu_p, \nu_p \rangle} \right) = \langle 1, 0 \rangle$$

donde $\odot$ es uno de los conectivos $\otimes, \oplus, \wedge$.

- iv) Para toda fórmula F , $\mathfrak{I}(F_{\langle 0, 0 \rangle}) = \langle 1, 0 \rangle$.

Definición 3.16. Sean $\mathfrak{I}$ una interpretación anotada en $\mathcal{B}$ para $\mathcal{L}$ y $\mathfrak{F}$ una fórmula anotada cerrada de $\mathcal{L}$. Entonces $\mathfrak{I}$ es un **modelo** para $\mathfrak{F}$ si $\mathfrak{I}(\mathfrak{F}) = \langle 1, 0 \rangle$. Se dice que $\mathfrak{I}$ es un modelo para un conjunto $\mathfrak{J}$ de fórmulas anotadas, si $\mathfrak{I}$ es un modelo para cada fórmula anotada de $\mathfrak{J}$.

Recordemos que si $\mathfrak{F}$ es “cerrada”, su valor de verdad no depende de la asignación $\mathfrak{A}$.

Definición 3.17. Sean $\mathfrak{Z}$ un conjunto de fórmulas anotadas cerradas y $\mathfrak{F}$ una fórmula anotada cerrada de $\mathfrak{L}$. Se dice que $\mathfrak{F}$ es una *consecuencia lógica* de $\mathfrak{Z}$ si, para cada interpretación $\mathfrak{T}$ de $\mathfrak{L}$, $\mathfrak{T}$ es un modelo para $\mathfrak{Z}$ implica que $\mathfrak{T}$ es un modelo para $\mathfrak{F}$.

3.5. Programas lógicos anotados sobre el birretículo $\mathcal{B}$

En esta sección definiremos los conceptos necesarios de la programación lógica tal como lo hicimos en la página 21 para el caso clásico, pero ahora apoyados en la lógica anotada sobre el birretículo $\mathcal{B}$.

Definición 3.18. Un *literal anotado* es un átomo anotado o la negación átomo anotado. Un literal anotado *positivo* es un átomo anotado. Un literal anotado *negativo* es la negación de un átomo anotado.

Definición 3.19. Un *programa lógico anotado en birretículos* $\mathfrak{P}$ consiste de un conjunto finito de *cláusulas de programa* (anotadas) de la forma:

$$A_{\langle\mu,\nu\rangle} \leftarrow L_{1\langle\mu_1,\nu_1\rangle}, \dots, L_{n\langle\mu_n,\nu_n\rangle}$$

donde $A_{\langle\mu,\nu\rangle}$ denota un átomo anotado llamado la **cabeza** de la cláusula y $L_{1\langle\mu_1,\nu_1\rangle}, \dots, L_{n\langle\mu_n,\nu_n\rangle}$ denota $L_{1\langle\mu_1,\nu_1\rangle} \otimes \dots \otimes L_{n\langle\mu_n,\nu_n\rangle}$ y es llamado el **cuerpo** de la cláusula; cada $L_{i\langle\mu_i,\nu_i\rangle}$ es un literal anotado llamado un *literal de cuerpo* de la cláusula.

Observación 46. Es necesario mencionar que cualquier programa lógico dado contiene solamente un conjunto finito de anotaciones constantes.

Procedemos a dar una caracterización semántica de los programas lógicos anotados en birretículos que hemos definido.

87

i) θ es un *umg* de $\mathcal{B}_k$ y $\mathcal{A}_j$, para algún $j = 1, \dots, m$, y una de las siguientes condiciones se cumple:

a) ó λ es un *umg* de $\langle \mu_k, \nu_k \rangle$ y $\langle \mu_j^*, \nu_j^* \rangle$;

b) ó $\lambda[\langle \mu_k, \nu_k \rangle]$ y $\lambda[\langle \mu_j^*, \nu_j^* \rangle]$ son instancias que reciben valores constantes tal que $\langle \mu_k, \nu_k \rangle \leq_c \langle \mu_j^*, \nu_j^* \rangle$;

ó

ii) θ es una *umg* de $\mathcal{B}_k$ y $\mathcal{A}_{j_1}, \dots, \mathcal{A}_{j_t}$, con $1 \leq j_1, \dots, j_t \leq m$ y,

a) ó λ es un *umg* de $\langle \mu_k, \nu_k \rangle$ y $\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle, \dots, \langle \mu_{j_t}^*, \nu_{j_t}^* \rangle$;

b) ó $\lambda[\langle \mu_k, \nu_k \rangle]$ y $\lambda[\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle], \dots, \lambda[\langle \mu_{j_t}^*, \nu_{j_t}^* \rangle]$ son instancias que reciben valores constantes tal que

$$\langle \mu_k, \nu_k \rangle \leq_c (\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle \oplus \dots \oplus \langle \mu_{j_t}^*, \nu_{j_t}^* \rangle) = \bigoplus (\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle, \dots, \langle \mu_{j_t}^*, \nu_{j_t}^* \rangle)$$

ó

iii) La anotación del átomo seleccionado $\mathcal{B}_k$ es nula, es decir, $\langle \mu_k, \nu_k \rangle = \langle 0, 0 \rangle$.

• En el caso (**i**), $\mathcal{O}^*$ es el objetivo

$$\theta\lambda \left[\leftarrow \mathcal{B}_{1\langle \mu_1, \nu_1 \rangle}, \dots, \mathcal{B}_{(k-1)\langle \mu_{(k-1)}, \nu_{(k-1)} \rangle}, \text{cuerpo}_j, \mathcal{B}_{(k+1)\langle \mu_{(k+1)}, \nu_{(k+1)} \rangle}, \dots, \mathcal{B}_{p\langle \mu_p, \nu_p \rangle} \right]$$

y se dice que $\mathcal{O}^*$ es *derivado* de $\mathcal{O}$ y $\mathcal{C}_j$ usando $\theta\lambda$.

• En el caso (**ii**), $\mathcal{O}^*$ es el objetivo

$$\theta\lambda \left[\leftarrow \mathcal{B}_{1\langle \mu_1, \nu_1 \rangle}, \dots, \mathcal{B}_{(k-1)\langle \mu_{(k-1)}, \nu_{(k-1)} \rangle}, \text{cuerpo}_{j_1}, \dots, \dots, \text{cuerpo}_{j_t}, \mathcal{B}_{(k+1)\langle \mu_{(k+1)}, \nu_{(k+1)} \rangle}, \dots, \mathcal{B}_{p\langle \mu_p, \nu_p \rangle} \right]$$

y se dice que $\mathcal{O}^*$ es *derivado* de $\mathcal{O}$ y $\mathcal{C}_{j_1}, \dots, \mathcal{C}_{j_t}$, con $1 \leq j_1, \dots, j_t \leq m$, usando $\theta\lambda$.

• En el caso (**iii**), remueva $\mathcal{B}_{k\langle 0, 0 \rangle}$ del objetivo y forme el siguiente objetivo $\mathcal{O}^*$ así,

$$\leftarrow \mathcal{B}_{1\langle \mu_1, \nu_1 \rangle}, \dots, \mathcal{B}_{(k-1)\langle \mu_{(k-1)}, \nu_{(k-1)} \rangle}, \mathcal{B}_{(k+1)\langle \mu_{(k+1)}, \nu_{(k+1)} \rangle}, \dots, \mathcal{B}_{p\langle \mu_p, \nu_p \rangle}$$

que es $\mathcal{O}$ excepto que este no contiene a $\mathcal{B}_{k\langle 0, 0 \rangle}$.

Observación 47. En el ítem (**iii**), la anotación nula $\langle 0, 0 \rangle$, se puede interpretar como el desconocimiento total.

3.7. Semántica operacional anotada

Definición 3.23. Una *derivación-SLD* de $\mathfrak{P} \cup \{\mathcal{O}\}$ esta formado por:

- una sucesión de objetivos $\mathcal{O}_0, \mathcal{O}_1, \mathcal{O}_2, \dots$,
- una sucesión de conjuntos finitos $\mathcal{S}_1, \mathcal{S}_2, \dots$ de cláusulas de $\mathfrak{P}$ y,
- una sucesión $\theta_1\lambda_1, \theta_2\lambda_2, \dots$ de *umg*'s,

tal que $\mathcal{O} = \mathcal{O}_0$ y cada $\mathcal{O}_{i+1}$, $i \geq 0$, se *deriva* de $\mathcal{O}_i$ y $\mathcal{S}_{i+1}$ utilizando $\theta_{i+1}\lambda_{i+1}$.

Observación 48. Los $\mathcal{S}_1, \mathcal{S}_2, \dots$ son una *secuencia de conjuntos finitos de cláusulas de $\mathfrak{P}$ que intervienen en cada uno de los pasos de derivación de los objetivos, a diferencia del caso clásico de la Resolución-SLD (ver Definición 2.6), que comprende solamente una secuencia de cláusulas de $\mathfrak{P}$.*

Al igual que para el caso clásico, una *derivación-SLD* se puede clasificar en varios tipos (ver Definición 2.7). Nosotros definiremos una *derivación-SLD finita* que finaliza en *éxito*.

Definición 3.24. Una *refutación-SLD* de $\mathfrak{P} \cup \{\mathcal{O}\}$ es una *derivación-SLD finita* de $\mathfrak{P} \cup \{\mathcal{O}\}$, la cual tiene la cláusula vacía $\square$ como el último objetivo de la derivación.

Observación 49. Si el último objetivo de la derivación-SLD finita es $\mathcal{O}_m = \square$, diremos que la refutación-SLD tiene longitud m .

Definición 3.25. El *conjunto de éxitos (Success Set)* de $\mathfrak{P}$, denotado por $\mathfrak{SS}_{\mathfrak{P}}$, es el conjunto de todos los átomos básicos tal que su negación junto al programa lógico tienen una refutación-SLD, es decir,

$$\mathfrak{SS}_{\mathfrak{P}} = \{\mathcal{R} \in \mathfrak{B}_{\mathfrak{P}} \mid \mathfrak{P} \cup \{\leftarrow \mathcal{R}\} \text{ tiene una refutación-SLD}\}$$

donde $\mathcal{R}$ denota un elemento típico de $\mathfrak{B}_{\mathfrak{P}}$.

Definición 3.26. Una respuesta $\theta\lambda$ es una *respuesta calculada* para $\mathfrak{P} \cup \{\mathcal{O}\}$ si es la sustitución que se obtiene restringiendo la composición $\theta_n\lambda_n \cdots \theta_1\lambda_1$ a las variables de $\mathcal{O}$, donde $\theta_1, \dots, \theta_n$ y $\lambda_1, \dots, \lambda_n$ es la secuencia de *umg*'s empleados en una refutación-SLD de $\mathfrak{P} \cup \{\mathcal{O}\}$ de longitud n .

3.8. Semántica declarativa anotada en $\mathcal{B}$

En esta sección, definimos la interpretación de Herbrand para programas lógicos anotados en birretículos y presentamos algunas de sus propiedades. En adelante, $\mathfrak{P}$ denota un programa lógico anotado sobre birretículos para $\mathcal{L}$, donde $\mathcal{L}$ está ligado al lenguaje de primer orden $\mathcal{L}$ y al birretículo $\mathcal{B}$.

Definición 3.27. Un *termino básico* es un término de $\mathcal{L}$ que no contiene variables. Un *término de anotación básico* es un término de anotación de $\mathcal{B}$ que no contiene variables. Un *átomo anotado básico* es un átomo anotado de $\mathcal{L}$, cuyos términos en $\mathcal{L}$ y términos de anotación en $\mathcal{B}$, son básicos.

Definición 3.28. El *Universo de Herbrand* $\mathfrak{U}_{\mathfrak{P}}$ para $\mathfrak{P}$ es el conjunto de todos los *términos básicos* que pueden ser formados a partir de las constantes y símbolos de función de $\mathcal{L}$. En el caso que $\mathcal{L}$ no tenga constantes, se adiciona alguna constante, una dicha, para formar los términos básicos.

Observación 50. Como en el caso clásico, $\mathfrak{U}_{\mathfrak{P}}$ es finito sí, y sólo sí, no aparecen símbolos de función $\mathfrak{P}$.

Definición 3.29. Una *instancia básica* es una instancia en la cual el reemplazo simultaneo de cada ocurrencia de una variable, en una expresión dada, se hace por los elementos de $\mathfrak{U}_{\mathfrak{P}}$.

Definición 3.30. La *Base de Herbrand anotada* $\mathfrak{B}_{\mathfrak{P}}$ para $\mathfrak{P}$ es el conjunto de todos los *átomos anotados básicos* formados a partir de, las instancias básicas de los átomos de $\mathcal{L}$ y los términos de anotación básicos de $\mathcal{B}$. En el caso que $\mathcal{L}$ no tenga anotaciones constantes, se adiciona alguna constante de anotación, una dicha, para formar los términos de anotación básicos.

Observación 51. Igual que en el caso clásico, $\mathfrak{B}_{\mathfrak{P}}$ será infinito sí, y sólo sí, $\mathfrak{U}_{\mathfrak{P}}$ es infinito.

Definición 3.31. Una *estructura anotada en $\mathcal{B}$ de Herbrand* para $\mathfrak{P}$ es un par $\mathfrak{E}_{\mathfrak{P}} = \langle \mathfrak{U}_{\mathfrak{P}}, \mathfrak{T}_{\mathfrak{P}} \rangle$, donde $\mathfrak{U}_{\mathfrak{P}}$ es el *dominio de interpretación* de $\mathfrak{T}_{\mathfrak{P}}$, y $\mathfrak{T}_{\mathfrak{P}}$ es una *interpretación anotada en $\mathcal{B}$ de Herbrand* (o simplemente una *$\mathfrak{H}_{\mathcal{B}}$ -interpretación*) para $\mathfrak{P}$ que consiste de las siguientes dos funciones:

- La primer función $\mathfrak{I}_{\mathfrak{H}}$, que interpreta:
 - * a cada símbolo constante c y a cada símbolo de función n -ario f , por ellos mismos, es decir, $[c]^{\mathfrak{I}_{\mathfrak{H}}} = c$ y $[f]^{\mathfrak{I}_{\mathfrak{H}}} = f$, y
 - * a cada símbolo de relación n -ario R , una función $[R]^{\mathfrak{I}_{\mathfrak{H}}} : \mathcal{U}_{\mathfrak{P}}^n \longrightarrow \mathcal{B}$.
- La segunda función $\mathfrak{X}$ tal como se definió en 3.12.

Observación 52. *Note que las constantes y las funciones tomadas pueden ser tanto de $\mathcal{L}$ como de $\mathcal{B}$.*

Una $\mathfrak{H}_{\mathcal{B}}$ -interpretación fija el dominio y la interpretación para los símbolos constantes y símbolos de función de $\mathfrak{P}$, pero deja libre la interpretación de los símbolos de relación. Por lo tanto, hay tantas $\mathfrak{H}_{\mathcal{B}}$ -interpretaciones de $\mathfrak{P}$, como interpretaciones podamos definir de los símbolos de relación.

Por esta razón se adopta, convenientemente, definir una $\mathfrak{H}_{\mathcal{B}}$ -interpretación por la interpretación que se asigne a los símbolos de relación. De esta forma cada subconjunto de la base de Herbrand $\mathcal{B}_{\mathfrak{P}}$ nos definirá una $\mathfrak{H}_{\mathcal{B}}$ -interpretación.

Definición 3.32. Una $\mathfrak{H}_{\mathcal{B}}$ -interpretación para $\mathfrak{P}$, es una interpretación $\mathfrak{I}_{\mathfrak{H}} : \mathcal{B}_{\mathfrak{P}} \longrightarrow \mathcal{B}$, definida por:

$$\mathfrak{I}_{\mathfrak{H}} = \{R_{\langle\mu,\nu\rangle}(t_1, \dots, t_n) \in \mathcal{B}_{\mathfrak{P}} \mid \mathfrak{I}_{\mathfrak{H}}(R_{\langle\mu,\nu\rangle}(t_1, \dots, t_n)) = \langle 1, 0 \rangle\}$$

donde $R_{\langle\mu,\nu\rangle}(t_1, \dots, t_n)$ denota un elemento típico de $\mathcal{B}_{\mathfrak{P}}$.

Una $\mathfrak{H}_{\mathcal{B}}$ -interpretación asocia simultáneamente cada símbolo de relación n -ario en $\mathfrak{P}$ con un único símbolo de relación n -ario en $\mathcal{U}_{\mathfrak{P}}$.

Definición 3.33. Sea $\mathfrak{P}$ como antes. Un *modelo anotado en $\mathcal{B}$ de Herbrand* (o simplemente un $\mathfrak{H}_{\mathcal{B}}$ -modelo) para $\mathfrak{P}$ es una $\mathfrak{H}_{\mathcal{B}}$ -interpretación que es un *modelo* de $\mathfrak{P}$.

Observación 53. *Cada una de las $\mathfrak{H}_{\mathcal{B}}$ -interpretaciones constituyen un $\mathfrak{H}_{\mathcal{B}}$ -modelo para el subconjunto de $\mathcal{B}_{\mathfrak{P}}$ que representa.*

Observación 54. $\mathcal{B}_{\mathfrak{P}}$ siempre será un $\mathfrak{H}_{\mathcal{B}}$ -modelo de $\mathfrak{P}$, para todo $\mathfrak{P}$. $\emptyset$ será un $\mathfrak{H}_{\mathcal{B}}$ -modelo de $\mathfrak{P}$ sí, y solo sí, $\mathfrak{P}$ no posee hechos.

Dado que los $\mathfrak{H}_{\mathcal{B}}$ -interpretaciones son subconjuntos de $\mathfrak{B}_{\mathfrak{P}}$, se puede definir un orden parcial en $\mathfrak{H}_{\mathcal{B}}^{\mathfrak{P}}$, la relación de inclusión $\subseteq$. Entonces, $\mathfrak{H}_{\mathcal{B}}^{\mathfrak{P}}$ tiene estructura de retículo bajo el orden parcial $\subseteq$. Esto permite pensar en la existencia de un elemento mínimo. Para el desarrollo de nuestra teoría estamos interesados en el elemento mínimo de $\mathfrak{H}_{\mathcal{B}}^{\mathfrak{P}}$ que sea $\mathfrak{H}_{\mathcal{B}}$ -modelo.

Definición 3.34. Sea $\mathfrak{P}$ como antes. La intersección de todos los $\mathfrak{H}_{\mathcal{B}}$ -modelos para $\mathfrak{P}$ será el *modelo mínimo de Herbrand* y se denota por $\mathfrak{M}_{\mathfrak{P}}$.

3.9. Semántica de Punto Fijo anotada en $\mathcal{B}$

En esta sección, se define el operador semántico $\mathcal{T}_{\mathfrak{P}}$ y se muestra como caracterizar los *modelos mínimos* de $\mathfrak{P}$ usando $\mathcal{T}_{\mathfrak{P}}$.

Para las siguientes definiciones, sean $\mathcal{L}$ un retículo completo (aunque se puede definir para cualquier conjunto parcialmente ordenado) y $\mathcal{T} : \mathcal{L} \rightarrow \mathcal{L}$ una función.

Definición 3.35. $\mathcal{T}$ es una función *monótona* sí, y solo sí, para todo $a, b \in \mathcal{L}$ se tiene que $a \leq b$ implica $\mathcal{T}(a) \leq \mathcal{T}(b)$, donde $\leq$ denota el orden definido en $\mathcal{L}$.

Definición 3.36. $\mathcal{T}$ es una función *continua* sí, y solo sí, $\mathcal{T}$ es *monótona* y preserva supremos sobre cadenas, es decir, para todo cadena $a_1 \leq a_2 \leq \dots$ en $\mathcal{L}$ se cumple que

$$\mathcal{T} \left(\bigsqcup_{i \geq 0} a_i \right) = \bigsqcup_{i \geq 0} \mathcal{T}(a_i)$$

donde $\bigsqcup$ se refiere al supremo de los elementos.

Definición 3.37. Sea a un elemento de $\mathcal{L}$. Diremos que a es un *punto fijo* de $\mathcal{T}$ sí, y solo sí, $\mathcal{T}(a) = a$.

Definición 3.38. Sea a un elemento de $\mathcal{L}$. Diremos que a es el *menor punto fijo* (*mpf*) de $\mathcal{T}$ sí, y solo sí, para todo otro *punto fijo* b de $\mathcal{T}$, $a \leq b$.

A continuación, se define el operador semántico $\mathcal{T}_{\mathfrak{P}}$, se prueba su continuidad y se muestra que $\mathcal{T}_{\mathfrak{P}}$ calcula el *modelo mínimo de Herbrand*, $\mathfrak{M}_{\mathfrak{P}}$, es decir, calcula las *consecuencias lógicas* de un programa lógico anotado en birretículos $\mathfrak{P}$.

Sea $\mathfrak{P}$ un programa lógico anotado en el birretículo $\mathcal{B}$ y $\mathfrak{H}_{\mathcal{B}}^{\mathfrak{P}}$ el conjunto de todas las $\mathfrak{H}_{\mathcal{B}}$ -interpretaciones para $\mathfrak{P}$.

Definición 3.39. El *operador semántico* $\mathcal{T}_{\mathfrak{P}}$ para $\mathfrak{P}$ es una función

$$\mathcal{T}_{\mathfrak{P}} : \mathfrak{H}_{\mathcal{B}}^{\mathfrak{P}} \longrightarrow \mathfrak{H}_{\mathcal{B}}^{\mathfrak{P}}$$

definida por:

$$\mathcal{T}_{\mathfrak{P}}(\mathfrak{H}_{\mathcal{B}}) = \{ R_{\langle \mu, \nu \rangle} \in \mathfrak{B}_{\mathfrak{P}} \text{ tal que se puede dar}$$

- (i) $R_{\langle \mu, \nu \rangle} \longleftarrow L_{1\langle \mu_1, \nu_1 \rangle}, \dots, L_{n\langle \mu_n, \nu_n \rangle}$ es una instancia básica de una cláusula en $\mathfrak{P}$, tal que existen $\langle \mu'_1, \nu'_1 \rangle, \dots, \langle \mu'_n, \nu'_n \rangle$ anotaciones que satisfacen $\{ L_{1\langle \mu'_1, \nu'_1 \rangle}, \dots, L_{n\langle \mu'_n, \nu'_n \rangle} \} \subseteq \mathfrak{H}_{\mathcal{B}}$, y para cada $\langle \mu'_i, \nu'_i \rangle$, se tiene que $\langle \mu_i, \nu_i \rangle \leq_c \langle \mu'_i, \nu'_i \rangle$,
ó

- (ii) existen átomos anotados básicos $R_{\langle \mu_1^*, \nu_1^* \rangle}, \dots, R_{\langle \mu_k^*, \nu_k^* \rangle} \in \mathfrak{H}_{\mathcal{B}}$ tal que $\langle \mu, \nu \rangle \leq_c (\langle \mu_1^*, \nu_1^* \rangle \oplus \dots \oplus \langle \mu_k^*, \nu_k^* \rangle)$.

El ítem (i) refleja la propiedad indicada en la proposición 3.2 ítem (i). El ítem (ii) refleja las propiedades indicadas en la proposición 3.2 ítem (ii) y (iii). Además, el ítem (iv) de la proposición 3.2 asegura que para cada fórmula $\mathcal{F}$, $\mathcal{F}_{\langle 0, 0 \rangle} \in \mathfrak{H}_{\mathcal{B}}$, y se asume esta propiedad de $\mathfrak{H}_{\mathcal{B}}$ para el resto de este trabajo. Por lo tanto, se asume que cada $\mathfrak{H}_{\mathcal{B}}$ -interpretación contiene todos los átomos anotados básicos de la forma $\mathcal{F}_{\langle 0, 0 \rangle}$ antes de cualquier aplicación del operador semántico.

Consideremos la siguiente sucesión (transfinita) *creciente* de $\mathfrak{H}_{\mathcal{B}}$ -interpretaciones

$$\begin{aligned} \mathcal{T}_{\mathfrak{P}} \uparrow 0 &= \emptyset \\ \mathcal{T}_{\mathfrak{P}} \uparrow 1 &= \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow 0) = \mathcal{T}_{\mathfrak{P}}(\emptyset) \\ \mathcal{T}_{\mathfrak{P}} \uparrow 2 &= \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow 1) = \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}}(\emptyset)) \\ \mathcal{T}_{\mathfrak{P}} \uparrow 3 &= \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow 2) = \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}}(\emptyset))) \\ &\vdots \\ \mathcal{T}_{\mathfrak{P}} \uparrow i &= \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow (i-1)) \\ &\vdots \\ \mathcal{T}_{\mathfrak{P}} \uparrow \omega &= \bigsqcup_{i=0}^{\infty} \mathcal{T}_{\mathfrak{P}} \uparrow i = \sup \{ \mathcal{T}_{\mathfrak{P}}^k(\emptyset) / k \in \mathbb{N} \} \end{aligned}$$

siendo

$$\mathcal{T}_{\mathfrak{P}} \uparrow 0 \subseteq \mathcal{T}_{\mathfrak{P}} \uparrow 1 \subseteq \mathcal{T}_{\mathfrak{P}} \uparrow 2 \subseteq \cdots \subseteq \mathcal{T}_{\mathfrak{P}} \uparrow i \subseteq \cdots \subseteq \mathcal{T}_{\mathfrak{P}} \uparrow \omega$$

o de manera equivalente,

$$\emptyset \subseteq \mathcal{T}_{\mathfrak{P}}(\emptyset) \subseteq \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}}(\emptyset)) \subseteq \cdots \subseteq \mathcal{T}_{\mathfrak{P}}^k(\emptyset) \subseteq \cdots \subseteq \sup \{ \mathcal{T}_{\mathfrak{P}}^k(\emptyset) / k \in \mathbb{N} \}$$

Se observa que cuando para algún i se cumple que $\mathcal{T}_{\mathfrak{P}} \uparrow i = \mathcal{T}_{\mathfrak{P}}(\mathcal{T}_{\mathfrak{P}} \uparrow (i - 1))$, entonces $\mathcal{T}_{\mathfrak{P}} \uparrow i$ es un *punto fijo* de $\mathcal{T}_{\mathfrak{P}}$ y se cumple que $\mathcal{T}_{\mathfrak{P}} \uparrow \omega = \mathcal{T}_{\mathfrak{P}} \uparrow i$

Validez y Completez de la Resolución-SLD anotada en $\mathcal{B}$

Hemos presentado la Resolución-SLD en la Sección 2.1 y se realizaron las pruebas de validez y completez en la Sección 2.5, todo esto sobre lógicas de Primer Orden; luego, describimos una nueva lógica, llamada *lógica anotada*, y desarrollamos su teoría en el Capítulo 3, para lo cual decidimos emplear en el conjunto de anotaciones, la estructura de birretículo. En esta capítulo, por ser consecuencia directa de los capítulos anteriores, emplearemos la misma notación usada hasta el momento.

Lo que haremos será demostrar que el Principio de Resolución-SLD en lógicas anotadas sobre birretículos, es *válido* y *completo*; con base en la Definición 3.22, que describe la manera de operación del algoritmo de Resolución-SLD y el desarrollo presentado en el caso clásico del Capítulo 2. Todo esto presentado como una reconstrucción del trabajo desarrollado por Komendanskaya[20, pág. 153–157].

4.1. Validez de la resolución-SLD para programas lógicos anotados en birretículos

Tal como hemos indicado a lo largo de este trabajo, nuestro interés se enfoca en el desarrollo teórico de un método deductivo sobre programas lógicos definidos y por ende, sobre cláusulas definidas; además, nuestro conjunto de anotación es un conjunto con estructura de birretículo, tal como se expuso en la Sección 3.2.

El propósito en esta sección, es demostrar que el Principio de Resolución-SLD anotado sobre birretículos es también, como el caso clásico (ver Teorema 11), un método deductivo *válido*. El siguiente teorema muestra que los cálculos ejecutados por el algoritmo de resolución-SLD anotado son correctos.

Teorema 13 (Validez de la Resolución-SLD anotada).

Cada *respuesta calculada* para $\mathfrak{P} \cup \{\mathcal{O}\}$ es una *respuesta correcta* para $\mathfrak{P} \cup \{\mathcal{O}\}$.

Demostración.

Sean $\mathcal{O} : \quad \leftarrow \mathcal{B}_{1\langle\mu_1, \nu_1\rangle}, \dots, \mathcal{B}_{p\langle\mu_p, \nu_p\rangle}$ el objetivo y $\theta\lambda$ una respuesta calculada para $\mathfrak{P} \cup \{\mathcal{O}\}$, donde $\theta_1, \dots, \theta_n$ y $\lambda_1, \dots, \lambda_n$ es la secuencia de *umg*'s empleados en una refutación-SLD de $\mathfrak{P} \cup \{\mathcal{O}\}$ de longitud n .

Tenemos que probar que

$$\theta\lambda \left[\bigotimes (\mathcal{B}_{1\langle\mu_1, \nu_1\rangle}, \dots, \mathcal{B}_{p\langle\mu_p, \nu_p\rangle}) \right] = \theta\lambda [\mathcal{B}_{1\langle\mu_1, \nu_1\rangle} \otimes \dots \otimes \mathcal{B}_{p\langle\mu_p, \nu_p\rangle}]$$

es una consecuencia lógica de $\mathfrak{P}$. Lo haremos aplicando inducción sobre la longitud n de la refutación-SLD.

Paso básico

Supongamos que $n = 1$, esto significa que el objetivo es de la forma:

$$\mathcal{O} : \quad \leftarrow \mathcal{B}_{\langle\mu, \nu\rangle}$$

y una de las siguientes condiciones se cumple:

i) $\mathfrak{P}$ tiene un hecho, de la forma $\mathcal{A}_{j\langle\mu_j^*, \nu_j^*\rangle} \leftarrow$, para algún $1 \leq j \leq m$, tal que $(\theta\lambda = \theta_1\lambda_1) \ \theta[\mathcal{A}_j] = \theta[\mathcal{B}]$ y,

a) ó $\lambda[\langle\mu, \nu\rangle] = \lambda[\langle\mu_j^*, \nu_j^*\rangle]$

b) ó $\lambda[\langle\mu, \nu\rangle]$ y $\lambda[\langle\mu_j^*, \nu_j^*\rangle]$ son instancias que reciben valores constantes tal que $\langle\mu, \nu\rangle \leq_c \langle\mu_j^*, \nu_j^*\rangle$.

ó

ii) $\mathfrak{P}$ tiene cláusulas de la forma:

$$\begin{aligned} \mathcal{A}_{j_1\langle\mu_{j_1}^*, \nu_{j_1}^*\rangle} &\leftarrow \\ \mathcal{A}_{j_2\langle\mu_{j_2}^*, \nu_{j_2}^*\rangle} &\leftarrow \\ \vdots & \\ \mathcal{A}_{j_t\langle\mu_{j_t}^*, \nu_{j_t}^*\rangle} &\leftarrow \end{aligned}$$

con $1 \leq j_1, \dots, j_t \leq m$, tal que $\theta[\mathcal{B}] = \theta[\mathcal{A}_{j_1}] = \dots = \theta[\mathcal{A}_{j_t}]$ y,

- a)** ó $\lambda[\langle \mu, \nu \rangle] = \lambda[\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle] = \dots = \lambda[\langle \mu_{j_t}^*, \nu_{j_t}^* \rangle]$
b) ó $\lambda[\langle \mu_1, \nu_1 \rangle]$ y $\lambda[\langle \mu_1^*, \nu_1^* \rangle], \dots, \lambda[\langle \mu_t^*, \nu_t^* \rangle]$ son instancias que reciben valores constantes tal que $\langle \mu_1, \nu_1 \rangle \leq_c \bigoplus (\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle, \dots, \langle \mu_{j_t}^*, \nu_{j_t}^* \rangle)$.

ó

iii) La anotación de $\mathcal{B}$ es $\langle \mu, \nu \rangle = \langle 0, 0 \rangle$, es decir, el objetivo es de la forma

$$\mathcal{O} : \quad \longleftarrow \mathcal{B}_{\langle 0, 0 \rangle}$$

Supongamos que se cumple (**i.a**). Dado que $\theta\lambda[\mathcal{A}_{j\langle \mu_j^*, \nu_j^* \rangle}]$ es la instancia de una cláusula $\mathcal{C}_j$, $1 \leq j \leq m$, de $\mathfrak{P}$, entonces $\theta\lambda[\mathcal{A}_{j\langle \mu_j^*, \nu_j^* \rangle}]$ es una consecuencia lógica de $\mathfrak{P}$; por consiguiente, $\theta\lambda[\mathcal{B}_{\langle \mu, \nu \rangle}]$ es una consecuencia lógica de $\mathfrak{P}$.

Supongamos que se cumple (**i.b**). Dado que $\theta\lambda[\mathcal{A}_{j\langle \mu_j^*, \nu_j^* \rangle}]$ es la instancia de una cláusula $\mathcal{C}_j$, $1 \leq j \leq m$, de $\mathfrak{P}$, que $\theta[\mathcal{A}_j] = \theta[\mathcal{B}]$ y que $\langle \mu, \nu \rangle \leq_c \langle \mu_j^*, \nu_j^* \rangle$, entonces por el ítem (**i**) de la Proposición 3.2, concluimos que $\theta\lambda[\mathcal{B}_{\langle \mu, \nu \rangle}]$ es una consecuencia lógica de $\mathfrak{P}$.

Supongamos que se cumple (**ii**). El caso (**ii.a**) se prueba análogamente al punto (**i.a**). Para el caso (**ii.b**), dado que todas

$$\begin{array}{c} \theta\lambda[\mathcal{A}_{j_1\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle} \longleftarrow \quad] \\ \theta\lambda[\mathcal{A}_{j_2\langle \mu_{j_2}^*, \nu_{j_2}^* \rangle} \longleftarrow \quad] \\ \vdots \\ \theta\lambda[\mathcal{A}_{j_t\langle \mu_{j_t}^*, \nu_{j_t}^* \rangle} \longleftarrow \quad] \end{array}$$

son instancias de hechos de $\mathfrak{P}$, cada una de ellas son consecuencias lógicas de $\mathfrak{P}$. Aplicando la Proposición 3.2 ítem (**ii**) tenemos que $(\mathcal{A}_{j_1\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle} \oplus \dots \oplus \mathcal{A}_{j_t\langle \mu_{j_t}^*, \nu_{j_t}^* \rangle})$ es una consecuencia lógica de $\mathfrak{P}$. Ahora, usando primero el ítem (**iii**), con el hecho que $\theta[\mathcal{B}] = \theta[\mathcal{A}_{j_1}] = \dots = \theta[\mathcal{A}_{j_t}]$; y luego el ítem (**i**), con el hecho que $\langle \mu_1, \nu_1 \rangle \leq_c (\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle \oplus \dots \oplus \langle \mu_{j_t}^*, \nu_{j_t}^* \rangle)$, concluimos que $\lambda\theta[\mathcal{B}_{\langle \mu, \nu \rangle}]$ es una consecuencia lógica de $\mathfrak{P}$.

Finalmente, en el caso (**iii**) aplicamos el ítem (**iv**) de la Proposición 3.2 y concluimos que $\lambda\theta[\mathcal{B}_{\langle \mu, \nu \rangle}]$ es una consecuencia lógica de $\mathfrak{P}$.

Paso inductivo

Supongamos que el resultado se cumple para respuestas calculadas obtenidas de refutaciones-SLD de longitud $(n - 1)$.

Supongamos que $\theta_1 \lambda_1, \dots, \theta_n \lambda_n$ es la secuencia de *umg*'s usados en una refutación-SLD de $\mathfrak{P} \cup \{\mathcal{O}\}$ de longitud n .

- i) Supongamos además que el primer paso en la refutación-SLD de longitud n fue hecho de la misma forma expuesta en la Definición 3.22 ítem (i), y sea $\mathcal{A}_{1\langle\mu_1^*, \nu_1^*\rangle} \leftarrow \text{cuerpo}_1$ la primer cláusula de entrada, tal que

$$\theta_n \cdots \theta_2 \theta_1 [\mathcal{A}_{1\langle\mu_1^*, \nu_1^*\rangle}] = \theta_n \cdots \theta_2 \theta_1 [\mathcal{B}_{k\langle\mu_k, \nu_k\rangle}]$$

para algún $\mathcal{B}_{k\langle\mu_k, \nu_k\rangle}$ en $\mathcal{O}$. Por la hipótesis inductiva,

$$\theta_n \lambda_n \cdots \theta_1 \lambda_1 \left[\bigotimes (\mathcal{B}_{1\langle\mu_1, \nu_1\rangle}, \dots, \mathcal{B}_{(k-1)\langle\mu_{(k-1)}, \nu_{(k-1)}\rangle}, \text{cuerpo}_1, \right. \\ \left. \mathcal{B}_{(k+1)\langle\mu_{(k+1)}, \nu_{(k+1)}\rangle}, \dots, \mathcal{B}_{p\langle\mu_p, \nu_p\rangle}) \right]$$

es una consecuencia lógica de $\mathfrak{P}$, debido a que solamente son necesarios $(n-1)$ pasos para obtener una refutación-SLD en esta fórmula. Entonces, por la Proposición 3.2 ítem (ii), $\theta_n \lambda_n \cdots \theta_1 \lambda_1 [\text{cuerpo}_1]$ es una consecuencia lógica de $\mathfrak{P}$. Por consiguiente, $\theta_n \lambda_n \cdots \theta_1 \lambda_1 [\mathcal{A}_{1\langle\mu_1^*, \nu_1^*\rangle}]$ es una consecuencia lógica de $\mathfrak{P}$.

Ahora, por el ítem (i) de la Definición 3.22, puede ser que:

$$\text{ó } \lambda_n \cdots \lambda_2 \lambda_1 [\langle\mu_k, \nu_k\rangle] = \lambda_n \cdots \lambda_2 \lambda_1 [\langle\mu_1^*, \nu_1^*\rangle]$$

y en este caso, evidentemente $\theta_n \lambda_n \cdots \theta_1 \lambda_1 [\mathcal{B}_{k\langle\mu_k, \nu_k\rangle}]$ es una consecuencia lógica de $\mathfrak{P}$;

ó $\lambda_n \cdots \lambda_2 \lambda_1 [\langle\mu_k, \nu_k\rangle]$ y $\lambda_n \cdots \lambda_2 \lambda_1 [\langle\mu_1^*, \nu_1^*\rangle]$ son instancias que reciben valores constantes tal que $\langle\mu_k, \nu_k\rangle \leq_c \langle\mu_1^*, \nu_1^*\rangle$; y por la Proposición 3.2 ítem (i), se tiene que $\theta_n \lambda_n \cdots \theta_1 \lambda_1 [\mathcal{B}_{k\langle\mu_k, \nu_k\rangle}]$ es una consecuencia lógica de $\mathfrak{P}$.

- ii) Supongamos que el primer paso en la refutación-SLD de longitud n fue hecho de la misma forma expuesta en el ítem (ii) de la Definición 3.22, y sean

$$\begin{aligned} \mathcal{A}_{j_1\langle\mu_{j_1}^*, \nu_{j_1}^*\rangle} &\leftarrow \text{cuerpo}_{j_1} \\ \mathcal{A}_{j_2\langle\mu_{j_2}^*, \nu_{j_2}^*\rangle} &\leftarrow \text{cuerpo}_{j_2} \\ &\vdots \\ \mathcal{A}_{j_t\langle\mu_{j_t}^*, \nu_{j_t}^*\rangle} &\leftarrow \text{cuerpo}_{j_t} \end{aligned}$$

con $1 \leq j_1, \dots, j_t \leq m$, las cláusulas de entrada. Seleccionamos $\mathcal{B}_{k\langle\mu_k, \nu_k\rangle}$ de $\mathcal{O}$ para este paso de la refutación-SLD. Por la hipótesis inductiva,

$$\theta_n \lambda_n \cdots \theta_1 \lambda_1 \left[\bigotimes (\mathcal{B}_{1\langle\mu_1, \nu_1\rangle}, \dots, \mathcal{B}_{(k-1)\langle\mu_{(k-1)}, \nu_{(k-1)}\rangle}, \text{cuerpo}_{j_1}, \dots, \right. \\ \left. \dots, \text{cuerpo}_{j_t}, \mathcal{B}_{(k+1)\langle\mu_{(k+1)}, \nu_{(k+1)}\rangle}, \dots, \mathcal{B}_{p\langle\mu_p, \nu_p\rangle}) \right]$$

es una consecuencia lógica de $\mathfrak{P}$, debido a que solamente son necesarios $(n-1)$ pasos para obtener una refutación-SLD en esta fórmula. Por consiguiente, $\theta_n \lambda_n \cdots \theta_1 \lambda_1 [\text{cuerpo}_{j_1} \otimes \cdots \otimes \text{cuerpo}_{j_t}]$ es una consecuencia lógica de $\mathfrak{P}$. Lo cual implica que

$$\theta_n \lambda_n \cdots \theta_1 \lambda_1 \left[\mathcal{A}_{j_1 \langle \mu_{j_1}^*, \nu_{j_1}^* \rangle} \right], \dots, \theta_n \lambda_n \cdots \theta_1 \lambda_1 \left[\mathcal{A}_{j_t \langle \mu_{j_t}^*, \nu_{j_t}^* \rangle} \right]$$

son consecuencias lógicas de $\mathfrak{P}$. Luego, por el item (ii) de la Proposición 3.2,

$$\theta_n \lambda_n \cdots \theta_1 \lambda_1 \left[\mathcal{A}_{j_1 \langle \mu_{j_1}^*, \nu_{j_1}^* \rangle} \oplus \cdots \oplus \mathcal{A}_{j_t \langle \mu_{j_t}^*, \nu_{j_t}^* \rangle} \right]$$

también es una consecuencia lógica de $\mathfrak{P}$; y dado que

$$\theta_n \cdots \theta_2 \theta_1 [\mathcal{A}_{j_1}] = \cdots = \theta_n \cdots \theta_2 \theta_1 [\mathcal{A}_{j_t}] = \theta_n \cdots \theta_2 \theta_1 [\mathcal{B}_k]$$

y aplicando el item (iii) de la Proposición 3.2, se tiene que

$$\theta_n \lambda_n \cdots \theta_1 \lambda_1 \left[\mathcal{A}_{j_1 \langle \mu_{j_1}^*, \nu_{j_1}^* \rangle \oplus \cdots \oplus \langle \mu_{j_t}^*, \nu_{j_t}^* \rangle} \right]$$

es una consecuencia lógica de $\mathfrak{P}$.

Ahora, por el item (ii) de la Definición 3.22, puede ser que:

$$\text{ó } \lambda_n \cdots \lambda_2 \lambda_1 [\langle \mu_k, \nu_k \rangle] = \lambda_n \cdots \lambda_2 \lambda_1 [\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle] = \cdots = \lambda_n \cdots \lambda_2 \lambda_1 [\langle \mu_{j_t}^*, \nu_{j_t}^* \rangle]$$

y en este caso, evidentemente $\theta_n \lambda_n \cdots \theta_1 \lambda_1 [\mathcal{B}_{k \langle \mu_k, \nu_k \rangle}]$ es una consecuencia lógica de $\mathfrak{P}$;

ó $\lambda_n \cdots \lambda_2 \lambda_1 [\langle \mu_k, \nu_k \rangle]$ y $\lambda_n \cdots \lambda_2 \lambda_1 [\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle], \dots, \lambda_n \cdots \lambda_2 \lambda_1 [\langle \mu_{j_t}^*, \nu_{j_t}^* \rangle]$ son instancias que reciben valores constantes tal que

$$\langle \mu_k, \nu_k \rangle \leq_c (\langle \mu_{j_1}^*, \nu_{j_1}^* \rangle \oplus \cdots \oplus \langle \mu_{j_t}^*, \nu_{j_t}^* \rangle)$$

y por el item (i) de la Proposición 3.2, se tiene que $\theta_n \lambda_n \cdots \theta_1 \lambda_1 [\mathcal{B}_{k \langle \mu_k, \nu_k \rangle}]$ es una consecuencia lógica de $\mathfrak{P}$.

Finalmente, podemos aplicar el mismo argumento para el resto de los átomos $\mathcal{B}_{1 \langle \mu_1, \nu_1 \rangle}, \dots, \mathcal{B}_{(k-1) \langle \mu_{(k-1)}, \nu_{(k-1)} \rangle}, \mathcal{B}_{(k+1) \langle \mu_{(k+1)}, \nu_{(k+1)} \rangle}, \dots, \mathcal{B}_{p \langle \mu_p, \nu_p \rangle}$ de $\mathcal{O}$.

Por lo tanto,

$$\theta \lambda \left[\bigotimes (\mathcal{B}_{1 \langle \mu_1, \nu_1 \rangle}, \dots, \mathcal{B}_{p \langle \mu_p, \nu_p \rangle}) \right] = \theta_n \lambda_n \cdots \theta_1 \lambda_1 \left[\bigotimes (\mathcal{B}_{1 \langle \mu_1, \nu_1 \rangle}, \dots, \mathcal{B}_{p \langle \mu_p, \nu_p \rangle}) \right]$$

es una consecuencia lógica de $\mathfrak{P}$.

■

Tal como se demostró en el Corolario 11.1, parte de la relación existente entre el *conjunto de éxitos* y el *modelo mínimo de Herbrand* de un programa lógico definido; haremos lo propio en el caso anotado.

Corolario 13.1.

Para un programa anotado $\mathfrak{P}$, el *conjunto de éxitos* de $\mathfrak{P}$ está contenido en el *modelo mínimo de Herbrand* de $\mathfrak{P}$, es decir, $\mathfrak{ES}_{\mathfrak{P}} \subseteq \mathbf{mmh}_{\mathfrak{P}}$.

Demostración.

Sea $\mathcal{R}_{\langle\mu,\nu\rangle} \in \mathfrak{B}_{\mathfrak{P}}$ y suponga que $\mathfrak{P} \cup \{\leftarrow \mathcal{R}_{\langle\mu,\nu\rangle}\}$ tiene una refutación-SLD.

Por consiguiente, existe una *respuesta calculada* $\theta\lambda$ para $\mathfrak{P} \cup \{\leftarrow \mathcal{R}_{\langle\mu,\nu\rangle}\}$. Por el Teorema 13, $\theta\lambda$ es una *respuesta correcta* para $\mathfrak{P} \cup \{\leftarrow \mathcal{R}_{\langle\mu,\nu\rangle}\}$; luego, $\theta\lambda[\mathcal{R}_{\langle\mu,\nu\rangle}]$ es una consecuencia lógica de $\mathfrak{P}$. Pero $\mathcal{R}_{\langle\mu,\nu\rangle}$ es una instancia básica de una cláusula en $\mathfrak{P}$, debido a que $\mathcal{R}_{\langle\mu,\nu\rangle} \in \mathfrak{B}_{\mathfrak{P}}$, entonces $\theta\lambda[\mathcal{R}_{\langle\mu,\nu\rangle}] = \mathcal{R}_{\langle\mu,\nu\rangle}$, es decir, $\theta\lambda = \epsilon$ (sustitución identidad) y así, $\mathcal{R}_{\langle\mu,\nu\rangle}$ es una consecuencia lógica de $\mathfrak{P}$, es decir, $\mathfrak{P} \models \mathcal{R}_{\langle\mu,\nu\rangle}$. ■

Hasta aquí, hemos probado que el algoritmo de resolución-SLD anotado para programas lógicos definidos anotados en birretículos es **válido**. La siguiente sección finaliza la discusión formal de los programas lógicos anotados en birretículos mostrando que el algoritmo de resolución-SLD para estos programas es **completo**.

4.2. Completez de la resolución-SLD para programas lógicos anotados en birretículos

Tal como lo desarrollamos en la Sección 2.5, nuestro siguiente propósito es demostrar que el Principio de Resolución-SLD anotado sobre birretículos es también un método deductivo completo; para esto haremos uso del Lema 3 de la página 74.

El siguiente teorema de completez extiende el Teorema 12 correspondiente al caso clásico.

Teorema 14 (Completez de la Resolución-SLD anotada).

Sea $\mathfrak{P}$ un programa anotado. El *conjunto de éxitos* de $\mathfrak{P}$ es igual al *modelo mínimo de Herbrand* de $\mathfrak{P}$, es decir, $\mathfrak{ES}_{\mathfrak{P}} = \mathbf{mmh}_{\mathfrak{P}}$.

Demostración.

Por el Corolario 13.1, es suficiente probar que $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$ está contenido en $\mathfrak{SS}_{\mathfrak{P}}$.

Supongamos que $\mathcal{R}_{\langle\mu,\nu\rangle}$ está en $\mathbf{mm}\mathfrak{H}_{\mathfrak{P}}$. Por el Teorema 10, $\mathcal{R}_{\langle\mu,\nu\rangle} \in \mathcal{T}_{\mathfrak{P}} \uparrow n$, para algún $n \in \omega$.

Afirmamos que $\mathcal{R}_{\langle\mu,\nu\rangle} \in \mathcal{T}_{\mathfrak{P}} \uparrow n$ implica que $\mathfrak{P} \cup \{\leftarrow \mathcal{R}_{\langle\mu,\nu\rangle}\}$ tiene una refutación-SLD y, por tanto, que $\mathcal{R}_{\langle\mu,\nu\rangle}$ está en el *conjunto de éxitos*.

Probaremos esta afirmación aplicando inducción sobre la longitud n de la refutación-SLD.

Paso básico

Supongamos que $n = 1$. Entonces $\mathcal{R}_{\langle\mu,\nu\rangle} \in \mathcal{T}_{\mathfrak{P}} \uparrow 1$, significa que ó $\mathcal{R}_{\langle\mu,\nu\rangle} \leftarrow$ es una instancia básica de un hecho en $\mathfrak{P}$ ó $\langle\mu,\nu\rangle = \langle 0, 0 \rangle$; y en ambos casos, por los items (i) y (iii) de la Definición 3.22, $\mathfrak{P} \cup \{\leftarrow \mathcal{R}_{\langle\mu,\nu\rangle}\}$ tiene una refutación-SLD.

Paso inductivo

Supongamos que la afirmación se cumple para $(n - 1)$. Así, $\mathcal{R}_{\langle\mu,\nu\rangle} \in \mathcal{T}_{\mathfrak{P}} \uparrow (n - 1)$ implica que $\mathfrak{P} \cup \{\leftarrow \mathcal{R}_{\langle\mu,\nu\rangle}\}$ tiene una refutación-SLD.

Sea $\mathcal{R}_{\langle\mu,\nu\rangle} \in \mathcal{T}_{\mathfrak{P}} \uparrow n$. Por la definición de $\mathcal{T}_{\mathfrak{P}}$, se cumple uno de los siguientes casos:

- i) Existe una instancia básica de una cláusula $\mathcal{B}_{\langle\mu^*,\nu^*\rangle} \leftarrow \mathcal{B}_{1\langle\mu_1,\nu_1\rangle}, \dots, \mathcal{B}_{p\langle\mu_p,\nu_p\rangle}$ de $\mathfrak{P}$ tal que $\mathcal{R} = \theta[\mathcal{B}]$, $\langle\mu,\nu\rangle = \lambda[\langle\mu^*,\nu^*\rangle]$ y

$$\theta[\mathcal{B}_{1\langle\mu_1^*,\nu_1^*\rangle}], \dots, \theta[\mathcal{B}_{p\langle\mu_p^*,\nu_p^*\rangle}] \in \mathcal{T}_{\mathfrak{P}} \uparrow (n - 1)$$

con $\langle\mu_1^*,\nu_1^*\rangle, \dots, \langle\mu_p^*,\nu_p^*\rangle$ tal que:

$$\lambda[\langle\mu_j,\nu_j\rangle] \leq_c \langle\mu_j^*,\nu_j^*\rangle, \text{ para } j \in \{1, \dots, p\}.$$

- ii) Existen átomos anotados básicos $\mathcal{R}_{\langle\mu_1,\nu_1\rangle}, \dots, \mathcal{R}_{\langle\mu_q,\nu_q\rangle} \in \mathcal{T}_{\mathfrak{P}} \uparrow (n - 1)$ tal que $\langle\mu,\nu\rangle \leq_c (\langle\mu_1,\nu_1\rangle \oplus \dots \oplus \langle\mu_q,\nu_q\rangle)$.

Supongamos que se cumple (i). Por la hipótesis de inducción, cada $\mathfrak{P} \cup \{\leftarrow \theta[\mathcal{B}_{j\langle\mu_j^*,\nu_j^*\rangle}]\}$ para $j \in \{1, \dots, p\}$, tiene una refutación-SLD.

Queremos probar entonces que $\mathfrak{P} \cup \{\leftarrow \theta\lambda[\mathcal{B}_{j\langle\mu_j, \nu_j\rangle}]\}$ tiene una *refutación-SLD*, para $j \in \{1, \dots, p\}$.

Consideremos la refutación-SLD del objetivo $\mathcal{O} : \leftarrow \theta[\mathcal{B}_{j\langle\mu_j^*, \nu_j^*\rangle}]$. De acuerdo a la Definición 3.22, existen dos formas en las cuales $\mathcal{O}$ puede ser derivado.

Caso 1

Existe una cláusula $\mathcal{D}_{\langle\mu', \nu'\rangle} \leftarrow \text{cuerpo}$ en $\mathfrak{P}$ tal que $\theta[\mathcal{B}_j] = \theta[\mathcal{D}]$ y $\langle\mu_j^*, \nu_j^*\rangle \leq_c \lambda[\langle\mu', \nu'\rangle]$. Teniendo en cuenta que $\lambda[\langle\mu_j, \nu_j\rangle] \leq_c \langle\mu_j^*, \nu_j^*\rangle$, por transitividad de $\leq_c$, concluimos que $\lambda[\langle\mu_j, \nu_j\rangle] \leq_c \lambda[\langle\mu', \nu'\rangle]$. Pero entonces, por el item (i) de la Definición 3.22, el objetivo $\leftarrow \theta\lambda[\mathcal{B}_{j\langle\mu_j, \nu_j\rangle}]$ recibirá una refutación-SLD.

Caso 2

Existen cláusulas $\mathcal{D}_{1\langle\mu'_1, \nu'_1\rangle} \leftarrow \text{cuerpo}_1, \dots, \mathcal{D}_{m\langle\mu'_m, \nu'_m\rangle} \leftarrow \text{cuerpo}_m$ en $\mathfrak{P}$ tal que $\theta[\mathcal{B}_j] = \theta[\mathcal{D}_1] = \dots = \theta[\mathcal{D}_m]$ y $\langle\mu_j^*, \nu_j^*\rangle \leq_c (\lambda[\langle\mu'_1, \nu'_1\rangle] \oplus \dots \oplus \lambda[\langle\mu'_m, \nu'_m\rangle])$. Dado que $\lambda[\langle\mu_j, \nu_j\rangle] \leq_c \langle\mu_j^*, \nu_j^*\rangle$, tenemos que

$$\lambda[\langle\mu_j, \nu_j\rangle] \leq_c \left(\lambda[\langle\mu'_1, \nu'_1\rangle] \oplus \dots \oplus \lambda[\langle\mu'_m, \nu'_m\rangle] \right)$$

Así, por el item (ii) de la Definición 3.22, el objetivo $\leftarrow \theta\lambda[\mathcal{B}_{j\langle\mu_j, \nu_j\rangle}]$ recibirá una refutación-SLD.

Supongamos que se cumple (ii). Usando la hipótesis inductiva, cada uno de los $\mathfrak{P} \cup \{\leftarrow \mathcal{R}_{\langle\mu_1, \nu_1\rangle}\}, \dots, \mathfrak{P} \cup \{\leftarrow \mathcal{R}_{\langle\mu_p, \nu_p\rangle}\}$ tiene una refutación-SLD. Esto significa, de acuerdo a los items (i) y (ii) de la Definición 3.22, que existen cláusulas (solo las necesarias) $\mathcal{R}_{\langle\mu_1^*, \nu_1^*\rangle} \leftarrow \text{cuerpo}_1^*, \dots, \mathcal{R}_{\langle\mu_q^*, \nu_q^*\rangle} \leftarrow \text{cuerpo}_q^*$, tal que para cada $\mathcal{R}_{\langle\mu_i, \nu_i\rangle}$ se cumple uno de los siguientes casos:

- $\lambda[\langle\mu_i, \nu_i\rangle] \leq_c \lambda[\langle\mu_j^*, \nu_j^*\rangle]$. En este caso, $\mathfrak{P} \cup \{\leftarrow \text{cuerpo}_j^*\}$ tiene una refutación-SLD.
- $\lambda[\langle\mu_i, \nu_i\rangle] \leq_c \lambda[\langle\mu_j^*, \nu_j^*\rangle \oplus \dots \oplus \langle\mu_t^*, \nu_t^*\rangle]$. Entonces

$$\mathfrak{P} \cup \{\leftarrow \text{cuerpo}_j^*, \dots, \text{cuerpo}_t^*\}$$

tiene una refutación-SLD, para algún $j, t \in \{1, \dots, q\}$.

Combinando todas aquellas refutaciones-SLD para $i \in \{1, \dots, p\}$, obtenemos una refutación-SLD de $\mathfrak{P} \cup \{\leftarrow \theta\lambda[\text{cuerpo}_1^*, \dots, \text{cuerpo}_q^*]\}$. Pero entonces, de acuerdo a la Definición 3.22, items (i) y (ii), existe una refutación-SLD para $\mathfrak{P} \cup \{\leftarrow \theta\lambda[\mathcal{R}_{\langle\mu', \nu'\rangle}]\}$, donde $\lambda[\langle\mu', \nu'\rangle]$ es alguna anotación tal que

$$\lambda[\langle\mu', \nu'\rangle] \leq_c (\lambda[\langle\mu_1^*, \nu_1^*\rangle] \oplus \dots \oplus \lambda[\langle\mu_q^*, \nu_q^*\rangle]) \quad (*)$$

donde, en particular,

$$\begin{aligned}\lambda [\langle \mu', \nu' \rangle] &\leq_c \lambda [\langle \mu_j^*, \nu_j^* \rangle] && \text{(en el caso } \bullet \text{) } \text{ó} \\ \lambda [\langle \mu', \nu' \rangle] &\leq_c \lambda [\langle \mu_j^*, \nu_j^* \rangle \oplus \cdots \oplus \langle \mu_t^*, \nu_t^* \rangle] && \text{(en el caso } \bullet\bullet \text{)}\end{aligned}$$

para $j, t \in \{1, \dots, q\}$.

Ahora, teniendo que:

$$\begin{aligned}\text{ó cada } \lambda [\langle \mu_i, \nu_i \rangle] &\leq_c \lambda [\langle \mu_j^*, \nu_j^* \rangle], \\ \text{ó cada } \lambda [\langle \mu_i, \nu_i \rangle] &\leq_c \lambda [\langle \mu_j^*, \nu_j^* \rangle \oplus \cdots \oplus \langle \mu_t^*, \nu_t^* \rangle],\end{aligned}$$

para $i \in \{1, \dots, p\}$ y $j, t \in \{1, \dots, q\}$; se concluye, por la monotonía de $\oplus$, que

$$(\lambda [\langle \mu_1, \nu_1 \rangle] \oplus \cdots \oplus \lambda [\langle \mu_p, \nu_p \rangle]) \leq_c (\lambda [\langle \mu_1^*, \nu_1^* \rangle] \oplus \cdots \oplus \lambda [\langle \mu_q^*, \nu_q^* \rangle]) \quad (**)$$

Pero entonces, por la Definición 3.22, ítem (i), la condición (*) implica que, para cada anotación constante $\langle \mu^\diamond, \nu^\diamond \rangle$ tal que

$$\langle \mu^\diamond, \nu^\diamond \rangle = \lambda [\langle \mu^\diamond, \nu^\diamond \rangle] \leq_c (\lambda [\langle \mu_1^*, \nu_1^* \rangle] \oplus \cdots \oplus \lambda [\langle \mu_q^*, \nu_q^* \rangle])$$

existe una *refutación-SLD* para $\mathfrak{P} \cup \{\leftarrow \theta \lambda [\mathcal{R}_{\langle \mu^\diamond, \nu^\diamond \rangle}]\}$; de (**), vemos que en particular se cumple, para todo $\langle \mu^\diamond, \nu^\diamond \rangle$, que

$$\langle \mu^\diamond, \nu^\diamond \rangle \leq_c (\lambda [\langle \mu_1, \nu_1 \rangle] \oplus \cdots \oplus \lambda [\langle \mu_p, \nu_p \rangle]).$$

De acuerdo al ítem (ii) supuesto, entre tales $\langle \mu^\diamond, \nu^\diamond \rangle$ estará $\langle \mu^\diamond, \nu^\diamond \rangle = \lambda [\langle \mu, \nu \rangle]$. Por lo tanto, concluimos que $\mathfrak{P} \cup \{\leftarrow \theta \lambda [\mathcal{R}_{\langle \mu, \nu \rangle}]\}$ tiene una *refutación-SLD sin restricciones*. Finalmente, aplicamos el Lema 3 para obtener una *refutación-SLD* para $\mathfrak{P} \cup \{\leftarrow \mathcal{R}_{\langle \mu, \nu \rangle}\}$. ■

4.3. Conclusiones

Como observamos a través de todo este trabajo, el Principio de Resolución-SLD es una estrategia muy refinada de deducción, soportada por la lógica de Primer Orden, goza de facilidades para su implementación y permite adecuadamente extender todas sus propiedades a las lógicas Anotadas. Además, empleando la estructura de birretículos se logró enriquecer aun más la estrategia de deducción

respecto a la presentada por Kifer y Subrahmanian en *Theory of Generalized Annotated Logic Programming and its Applications*[16].

Se presentó detalladamente el algoritmo de Resolución-SLD anotado sobre birretículos, se contempló el manejo de variables en las anotaciones y se generalizó la selección de las cláusulas elegidas para cada paso de la resolución, a diferencia de lo expuesto por Komendantkaya[20], en la cual se asume un orden pre-establecido.

Al igual que para el caso clásico se define y se formalizan las semánticas Operacional, Declarativa y de Punto Fijo, enriqueciéndola así,

- Se definió una aplicación $\mathfrak{X}$ que interprete los términos del conjunto de anotación, empleando primero lo presentado para el caso clásico sobre la asignación de valores para las fórmulas de Primer Orden sobre elementos del birretículo y luego, con los valores obtenidos para las fórmulas, interpretarlos por medio de la nueva aplicación con la anotación de las fórmulas en la lógica Anotada; garantizando que se cumpla un mínimo de información, visualizado como valores en el birretículo, para que la fórmula anotada sea interpretada como verdadera.
- La inclusion de una propiedad en la definición del operador de consecuencia inmediata $\mathcal{T}_{\mathfrak{P}}$, que permite para el caso anotado incluir todas las consecuencias lógicas resultantes de un conjunto definido, situación que no alcanza a cubrir el operador clásico y deja átomos por fuera de las posibilidades de interpretación. Este hecho resulta ser muy relevante, ya que por medio de este “nuevo” operador es posible, como vimos en la prueba del Teorema 4.2, demostrar la completez del Principio de Resolución-SLD anotado.

Se enunció como referencia el lenguaje de programación PROLOG, que implementa la estrategia de Resolución-SLD y nos orienta en la percepción del gran alcance que tiene este estrategia en el campo computacional y por ende en la ingeniería, tal como lo exhibe en su Tesis de Doctorado Komendantskaya[17] sobre redes neuronales.

Finalmente, con base en la teoría de birretículos expuesta en el Capítulo 3, la interpretación de los valores de anotación se realizan en general empleando el orden de conocimiento $\leq_c$, en el cual se toma siempre los valores que sean mayores simultáneamente que otros; acordado así por Komendantskaya para quien resulta más significativo tener más información representada en los valores de la anotación. Sin embargo, considero que es más adecuado emplear el orden de verdad $\leq_v$ en el cual se elige el valor mayor en el primer término de la anotación y el menor en el segundo término de la anotación, siendo consecuente con el criterio interpretativo asignado a los valores de las anotaciones, donde se dice que el primer termino hace referencia al valor “a favor” y el segundo término al valor “en contra” de la posibilidad de que ocurra el evento representado por el átomo de Primer Orden.

Ejemplo Final

El propósito de este capítulo es ilustrar por medio de un ejemplo la manera en que se utiliza el algoritmo de Resolución-SLD para programas lógicos anotados sobre birretículos, aprovechando el hecho, expuesto en el Capítulo 4, que este método deductivo es válido y completo.

Komendantskaya[pág. 146][20], uso un ejemplo sencillo referente a la situación presentada en un aeropuerto en el cual se quiere implementar un programa que pueda tomar decisiones, sin supervisión humana, sobre si se retrasa el aeropuerto o se cancela un vuelo. Este programa trabaja con base a los datos que se obtienen a través de la acumulación de información de las previsiones meteorológicas tomadas de diferentes fuentes. Estas fuentes pueden dar información inconsistente o incompleta, que se recoge en la base de datos del programa.

Para nuestro ejemplo, emplearemos la situación presente en las salas de sistemas de la Universidad del Valle, para lo cual deseamos implementar un programa que decida, sin supervisión humana, cual es la posibilidad que tiene un usuario de hacer uso de un computador. Este programa trabaja con base a los datos que se obtienen a través de la acumulación de información de los registros de las visitas de usuarios durante un periodo establecido.

A.1. Planteamiento

La Universidad del Valle está interesada en optimizar el servicio de los computadores institucionales. Para esto ha decidido emplear una estrategia de Resolución-SLD sobre lógicas anotadas en el birretículo $\mathcal{B}_{25}$ (ver Gráfica 3.1), que permita agilizar y priorizar el servicio dependiendo de las condiciones académicas del estudiante y de las condiciones de servicio de la sala.

Las decisiones realizadas por el programa son calculadas con base en el registro histórico de utilización de los computadores en sus respectivas salas en los dos últimos semestres.

Las anotaciones $\langle \mu, \nu \rangle$, tomadas del birretículo $\mathcal{B}_{25}$, se interpretan como: μ es el valor a favor y ν es el valor en contra, de que suceda el evento.

Con este objetivo en mente, emplearemos el siguiente lenguaje:

<u>VARIABLES</u>	<u>CONSTANTES</u>	<u>DESCRIPCIÓN</u>
$[E]$ Estudiante	$[E_H]$ Est. de Humanidades $[E_C]$ Est. de Ciencias $[E_I]$ Est. de Ingeniería	Caracteriza al estudiante según la facultad que pertenece. Se destacan 3 facultades: Ciencias, Ingeniería y Humanidades.
$[J]$ Jornada	$[J_D]$ Jornada Diurna $[J_N]$ Jornada Nocturna	La jornada académica en la cual el estudiante de cierta facultad está matriculado. Las jornadas en estudio son: * Ciencias: Diurna. * Ingeniería: Diurna. * Humanidades: Nocturna.
$[S]$ Sala	$[S_B]$ Sala de Biblioteca $[S_C]$ Sala de Ciencias $[S_I]$ Sala de Ingeniería	Caracteriza las salas de sistemas en las cuales se prestará el servicio. Se destacan 3 salas: Biblioteca, Ciencias e Ingeniería.
$[H]$ Horario	$[H_M]$ Horario de la Mañana $[H_T]$ Horario de la Tarde $[H_N]$ Horario de la Noche	Horarios establecidos en los que presta el servicio las salas de sistemas; a saber: * Ciencias: Mañana. * Biblioteca: Tarde y Noche. * Ingeniería: Tarde.

<u>PREDICADOS</u>	<u>DESCRIPCIÓN</u>
[S] Servicio (E, J, S, H)	El <u>estudiante</u> de cierta <u>jornada</u> académica, utiliza el servicio de una <u>sala</u> de sistemas en cierto <u>horario</u> .
[EJ] EstudianteJornada (E, J)	El <u>estudiante</u> está matriculado en cierta <u>jornada</u> académica.
[JHO] JornadaHorario (J, S, H)	Para determinada <u>jornada</u> académica se presta el servicio de una <u>sala</u> en un <u>horario</u> establecido.
[HS] HorarioSala (H, S)	En un <u>horario</u> se puede acceder al servicio de una <u>sala</u> .
[DHO] DisponibleHorario (E, J, H)	El <u>estudiante</u> matriculado en una <u>jornada</u> académica, tiene acceso a un computador en cierto <u>horario</u> .

A.2. Programa

En esta sección haremos la presentación formal de nuestro programa lógico definido anotado $\mathfrak{P}$ sobre el birretículo $\mathcal{B}_{25}$.

Debemos destacar tres condiciones relevantes en la interpretación de las respuestas calculadas por el siguiente programa,

- Primero, existen tres situaciones (que las llamaremos *perfiles*) que describen la situación ideal (entiéndase por ideal lo más adecuado para el cumplimiento del objetivo del programa), para cada uno de los sujetos en estudio. A estos perfiles les asignamos la anotación $\langle 1, 0 \rangle$ que será interpretada como *todo* a favor y *nada* en contra.
- Segundo, las anotaciones nulas, a saber $\langle 0, 0 \rangle$, se deben interpretar como eventos que no satisfacen ninguna de las posibilidades del programa, es decir, situaciones que no existen bajo las condiciones inducidas por la base de datos.
- Por último, la regla de computación consiste en elegir el literal más a la izquierda del objetivo a resolver.

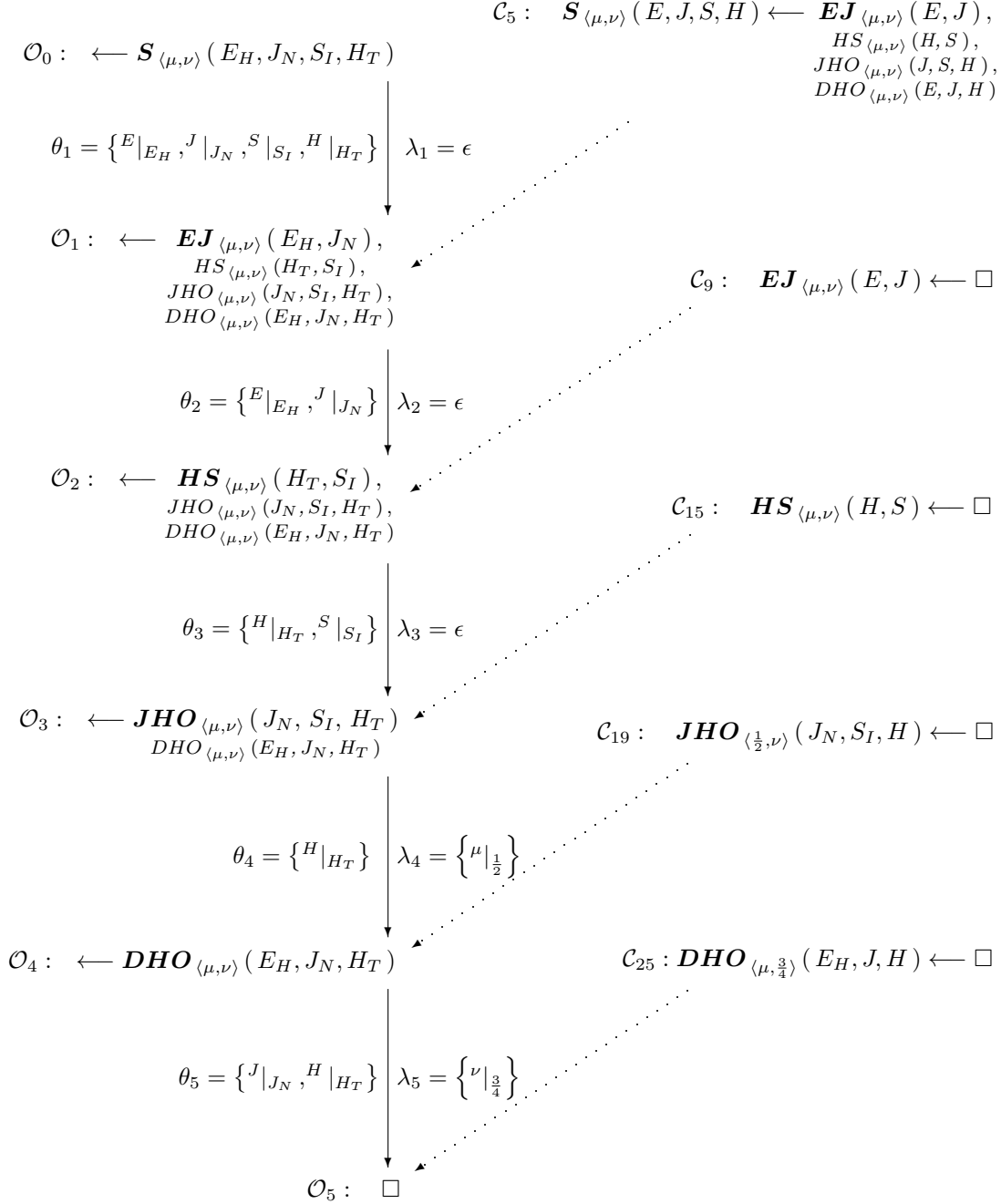
$$\mathfrak{P} = \left\{ \begin{array}{ll} \mathcal{C}_1 : \mathbf{S}_{\langle 1,0 \rangle} (E_C, J_D, S_C, H_M) & \leftarrow \square \\ \mathcal{C}_2 : \mathbf{S}_{\langle 1,0 \rangle} (E_I, J_D, S_I, H_T) & \leftarrow \square \\ \mathcal{C}_3 : \mathbf{S}_{\langle 0,1 \rangle} (E_H, J_N, S_C, H_M) & \leftarrow \square \\ \mathcal{C}_4 : \mathbf{S}_{\langle 1,0 \rangle} (E_H, J_N, S_B, H_N) & \leftarrow \square \\ \mathcal{C}_5 : \mathbf{S}_{\langle \mu, \nu \rangle} (E, J, S, H) & \leftarrow \mathbf{EJ}_{\langle \mu, \nu \rangle} (E, J), \mathbf{HS}_{\langle \mu, \nu \rangle} (H, S), \dots \\ & \mathbf{JHO}_{\langle \mu, \nu \rangle} (J, S, H), \mathbf{DHO}_{\langle \mu, \nu \rangle} (E, J, H) \\ \mathcal{C}_6 : \mathbf{EJ}_{\langle 0,0 \rangle} (E_C, J_N) & \leftarrow \square \\ \mathcal{C}_7 : \mathbf{EJ}_{\langle 0,0 \rangle} (E_I, J_N) & \leftarrow \square \\ \mathcal{C}_8 : \mathbf{EJ}_{\langle 0,0 \rangle} (E_H, J_D) & \leftarrow \square \\ \mathcal{C}_9 : \mathbf{EJ}_{\langle \mu, \nu \rangle} (E, J) & \leftarrow \square \\ \mathcal{C}_{10} : \mathbf{HS}_{\langle 0,0 \rangle} (H_M, S_B) & \leftarrow \square \\ \mathcal{C}_{11} : \mathbf{HS}_{\langle 0,0 \rangle} (H_M, S_I) & \leftarrow \square \\ \mathcal{C}_{12} : \mathbf{HS}_{\langle 0,0 \rangle} (H_N, S_C) & \leftarrow \square \\ \mathcal{C}_{13} : \mathbf{HS}_{\langle 0,0 \rangle} (H_N, S_I) & \leftarrow \square \\ \mathcal{C}_{14} : \mathbf{HS}_{\langle 0,0 \rangle} (H_T, S_C) & \leftarrow \square \\ \mathcal{C}_{15} : \mathbf{HS}_{\langle \mu, \nu \rangle} (H, S) & \leftarrow \square \\ \mathcal{C}_{16} : \mathbf{JHO}_{\langle \frac{3}{4}, \nu \rangle} (J, S, H_M) & \leftarrow \square \\ \mathcal{C}_{17} : \mathbf{JHO}_{\langle \frac{3}{4}, \nu \rangle} (J_D, S, H_T) & \leftarrow \square \\ \mathcal{C}_{18} : \mathbf{JHO}_{\langle \frac{1}{2}, \nu \rangle} (J, S, H_N) & \leftarrow \square \\ \mathcal{C}_{19} : \mathbf{JHO}_{\langle \frac{1}{2}, \nu \rangle} (J_N, S_I, H) & \leftarrow \square \\ \mathcal{C}_{20} : \mathbf{JHO}_{\langle \frac{3}{4}, \nu \rangle} (J_N, S_B, H) & \leftarrow \square \\ \mathcal{C}_{21} : \mathbf{DHO}_{\langle \mu, \frac{1}{2} \rangle} (E_C, J_D, H_T) & \leftarrow \square \\ \mathcal{C}_{22} : \mathbf{DHO}_{\langle \mu, 1 \rangle} (E, J_D, H_N) & \leftarrow \square \\ \mathcal{C}_{23} : \mathbf{DHO}_{\langle \mu, \frac{1}{2} \rangle} (E_I, J_D, H_M) & \leftarrow \square \\ \mathcal{C}_{24} : \mathbf{DHO}_{\langle \mu, \frac{1}{4} \rangle} (E_I, J_D, H_T) & \leftarrow \square \\ \mathcal{C}_{25} : \mathbf{DHO}_{\langle \mu, \frac{3}{4} \rangle} (E_H, J, H) & \leftarrow \square \end{array} \right.$$

A.3. Resolución

Objetivo 1

$$\mathcal{O}_0 : \quad \leftarrow \quad \mathbf{S}_{\langle \mu, \nu \rangle} (\mathbf{E}_H, \mathbf{J}_N, \mathbf{S}_I, \mathbf{H}_T)$$

¿Cual es la posibilidad de que un estudiante de la facultad de humanidades de la jornada nocturna, utilice el servicio de la sala de sistemas de ingeniería en el horario de la tarde?



Gráfica A.1. Resolución-SLD Objetivo 1 (Ejemplo Final)

Respuesta

Después de aplicar la estrategia de Resolución-SLD al programa $\mathfrak{P}$ junto con nuestro objetivo $\mathcal{O}_0$: $\leftarrow S_{\langle\mu,\nu\rangle}(E_H, J_N, S_I, H_T)$, obtenemos que la respuesta calculada $\theta\lambda = \epsilon \left\{ \mu|_{\frac{1}{2}}, \nu|_{\frac{3}{4}} \right\}$ es una respuesta correcta, de la cual podemos decir que la posibilidad de que el estudiante de humanidades de la jornada nocturna, utilice en la tarde un equipo de la sala de sistemas de ingeniería es de $\frac{1}{2}$ a favor y $\frac{3}{4}$ en contra; y esto lo interpretamos como que este usuario tendrá poco prioridad en la asignación de un equipo frente a los otros usuarios.

Objetivo 2

$$\mathcal{O}_0 : \quad \leftarrow \quad S_{\langle\mu,\nu\rangle}(E_I, J_D, S_I, H_T)$$

¿Cual es la posibilidad de que un estudiante de la facultad de ingeniería de la jornada diurna, utilice el servicio de la sala de sistemas de ingeniería en el horario de la tarde?

$$\begin{array}{ccc} \mathcal{O}_0 : \quad \leftarrow S_{\langle\mu,\nu\rangle}(E_I, J_D, S_I, H_T) & & \mathcal{C}_2 : \quad S_{\langle 1,0 \rangle}(E_I, J_D, S_I, H_T) \leftarrow \square \\ & \downarrow \lambda_1 = \{\mu|_1, \nu|_0\} \quad \theta_1 = \epsilon & \nearrow \text{dotted arrow} \\ & \mathcal{O}_1 : \quad \square & \end{array}$$

Gráfica A.2. Resolución-SLD Objetivo 2 (Ejemplo Final)

Respuesta

Para este ejemplo, al aplicar la estrategia de Resolución-SLD al programa $\mathfrak{P}$ junto con nuestro objetivo $\mathcal{O}_0$: $\leftarrow S_{\langle\mu,\nu\rangle}(E_I, J_D, S_I, H_T)$, obtenemos de manera trivial la respuesta calculada $\theta\lambda = \epsilon \left\{ \mu|_1, \nu|_0 \right\}$, ya que en este caso el objetivo corresponde con uno de los *perfiles* enunciados en la página 107. Luego, $\theta\lambda$ es una respuesta correcta, de la cual podemos decir que la posibilidad de que el estudiante de ingeniería de la jornada diurna, utilice en la tarde un equipo de la sala de sistemas de ingeniería es de 1 a favor y 0 en contra; y esto lo interpretamos como que este usuario tendrá toda la prioridad en la asignación de un equipo frente a los otros usuarios, ya que esta acudiendo a la sala de su facultad y en el horario correspondiente a su programa académico.

Referencias bibliográficas

- [1] M. Ben-Ari, *Mathematical Logic for Computer Science*, Prentice Hall International (UK), 1993.
- [2] M. Celma, L. Mota, and J. C. Casamayor, *Lógica y Base de Datos*, Memorias del XVII Congreso Nacional de Matemáticas (Medellín, Colombia), Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia (España), 2007, pp. 3–28.
- [3] A. Church, *An unsolvable problem of elementary number theory*, American Journal of Mathematics **58** (1936), 354–363.
- [4] M. Fitting, *Bilattices in logic programming*, 20th International Symposium on Multiple-Valued Logic, IEEE CS Press, 1990, pp. 238–246.
- [5] M. Fitting, *First-Order Logic and Automated Theorem Proving*, second ed., Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.
- [6] M. Fitting, *Fixpoint Semantics for Logic Programming - A Survey*, Theoretical Computer Science **278** (2002), no. 1–2, 25–51.
- [7] G. Frege, *Begriffsschrift, eine der arithmetischen nachgebildete formelsprache des reinen denkens*, Halle, Berlin, 1879.
- [8] H. Geuvers, *Herbrand Models and SLD resolution*, Tech. report, Institute for Computing and Information Science (ICIS), Radboud University Nijmegen, Netherlands, 14 December 2008.
- [9] P. A. Gilmore, *Structuring of Parallel Algorithms*, Journal of the Association for Computing Machinery **15** (1968), no. 2, 176–192.
- [10] M. L. Ginsberg, *Multivalued logics: a uniform approach to reasoning in artificial intelligence*, Computational Intelligence **4** (1988), 265–316.

- [11] K. Gödel, *The completeness of the axioms of the functional calculus*, From Frege to Gödel: A Source Book in Mathematical Logic (J. Van Heijenoort, ed.), 1967, pp. 582–591.
- [12] C. Green, *Application of Theorem Proving to Problem Solving*, Proceedings of the First International Joint Conference on Artificial Intelligence (Washington, DC) (Donald E. Walker and Lewis M. Morton, ed.), International Joint Conference on Artificial Intelligence, 1969, pp. 219–239.
- [13] J. Herbrand, *Recherches sur la théorie de la démonstration*, Ph.D. thesis, Université de Paris, France, 1930.
- [14] A. Horn, *On sentences which are true of direct unions of algebras*, Journal of Symbolic Logic **16** (1951), 14–21.
- [15] M. Kifer and E. L. Lozinskii, *RI: A Logic for Reasoning with Inconsistency*, Proceedings of the 4.th Annual Symposium on Logic in Computer Science (Asilomar Conference Center, Pacific Grove, California), IEEE Computer Society Press, 5–8 June 1989, pp. 253–262.
- [16] M. Kifer and V. S. Subrahmanian, *Theory of Generalized Annotated Logic Programming and its Applications*, Journal of Logic Programming (1992).
- [17] E. Komendantskaya, *Learning and Deduction in Neural Networks and Logic*, Ph.D. thesis, Department of Mathematics, University College Cork, Ireland, 2007.
- [18] E. Komendantskaya, *A Sequent Calculus for Bilattice-Based Logic and its Many-Sorted Representation*, Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX’07 (Nicola Olivetti, ed.), Lecture Notes in Computer Science, vol. 4548, Springer, 3–6 July 2007, pp. 165–182.
- [19] E. Komendantskaya and A. K. Seda, *Declarative and Operational Semantics for Bilattice-based Annotated Logic Programs*, Proceedings of the 4th Irish Conference on the Mathematical Foundations of Computer Science and Information Technology (MFCSIT) (Cork, Ireland), 2006, pp. 229–233.
- [20] E. Komendantskaya and A. K. Seda, *Sound and Complete SLD-Resolution for Bilattice-Based Annotated Logic Programs*, Electronic Notes in Theoretical Computer Science **225** (2009), 141–159.
- [21] E. Komendantskaya, A. K. Seda, and V. Komendantsky, *On approximation of the semantic operators determined by bilattice-based logic programs*, Proceedings of the 7.th International Workshop on First-Order Theorem Proving (Koblenz, Germany), 15–17 September 2005, pp. 112–130.

- [22] R. Kowalski and D. Kuehner, *Linear resolution with selection function*, Artificial Intelligence **2** (1971), no. 3–4, 227–260.
- [23] J. W. Lloyd, *Foundations of logic programming (2nd extended ed.)*, Springer-Verlag New York, Inc., USA, 1987.
- [24] D. W. Loveland, *A linear format for resolution*, Proceedings of the IRIA Symposium on Automatic Demonstration (New York), Springer-Verlag, 1970, pp. 147–162.
- [25] L. Löwenheim, *Über Möglichkeiten im Relativkalkül*, Mathematische Annalen **76** (1915), 447–470.
- [26] D. Luckham, *Refinements in resolution theory*, Proceedings of the IRIA Symposium on Automatic Demonstration (New York), Springer-Verlag, 1970, pp. 163–190.
- [27] M. Davis and H. Putnam, *A computing procedure for quantification theory*, Journal of the Association for Computing Machinery **7** (1960), no. 3, 201–215.
- [28] J. Muñoz, *Resolución en lógicas anotadas en retículos ordinarios*, Departamento de Matemáticas, Universidad del Valle (2005).
- [29] G. Ortiz, *Introducción a la Lógica Difusa*, Tech. report, Departamento de Matemáticas, Universidad del Valle, Cali, Colombia, 2006.
- [30] G. Ortiz, *Fundamentos de Computación Inteligente (FCI) Lógica Clásica y Lógicas Anotadas*, Tech. report, Departamento de Matemáticas, Universidad del Valle, Cali, Colombia, 26 Marzo 2008.
- [31] A. Polleres, *Lógica y Métodos Avanzados de Razonamiento*, Tech. Report Lecture 8, Digital Enterprise Research Institute (DERI), National University of Ireland, Galway, December 2008.
- [32] Dag Prawitz, *An improved proof procedure*, Theoria **26** (1960), 102–139.
- [33] C. E. Ramírez, *Semántica de Punto Fijo para programas anotados. Una realización por $\mathcal{U}$ -resolución*, Matemáticas: Enseñanza Universitaria, Escuela Regional de Matemáticas (Universidad del Valle - Colombia) **XVI** (2008), no. 2, 87–102.
- [34] J. A. Robinson, *Logic and logic programming*, Communications of the Association for Computing Machinery, Special Section on Logic Programming **35** (1992), no. 3, 40.
- [35] A. Rodríguez, *Un tratamiento a los birretículos entrelazados*, Departamento de Matemáticas, Universidad del Valle (2006).

- [36] G. R. Simari, *Logic for Computer Science*, Tech. Report Lectures 6,7,8, Department of Computer Science & Engineering, Universidad Nacional del Sur, Bahía Blanca, Argentina, Primer cuatrimestre 2008.
- [37] T. Skolem, *Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Sätze nebst einem Theoreme über die dichte Mengen*, Videnskapsselskapets skrifter, I. Matematisk-naturvidenskabelig klasse **4** (1920).
- [38] S. Staab and S. Schenk, *Advanced Data Modeling*, Tech. Report Slides 4, 7, 8, 9, Institut für Informatik, University of Koblenz, Landau, Germany, 2008.
- [39] V. S. Subrahmanian, *On the Semantics of Quantitative Logic Programs*, Proceedings of the 4.th IEEE Symposium on Logic Programming (Washington DC), Computer Society Press, 1987, pp. 173–182.
- [40] A. Tarski, *A decision method for elementary algebra and geometry*, University of California Press, Berkeley, 1951.
- [41] A. M. Turing, *On computable numbers, with an application to the Entscheidungs problem*, Proceedings of the London Mathematical Society **42** (1936), no. 2, 230–265.
- [42] M. H. van Emden, *Quantitative Deduction and its Fixpoint Theory*, Journal of Logic Programming **3** (1986), no. 1, 37–54.
- [43] M. H. van Emden and R. A. Kowalski, *The Semantics of Predicate Logic as a Programming Language*, Journal of the Association of Computing Machinery **23** (1976), no. 4, 733–742.

Lista de símbolos

$\mathcal{A}$, alfabeto de Primer Orden, 1 anotado, 77	$\mathcal{F}$, conjunto de símbolos de función, 2
ξ, μ, ν , anotación, 78	ζ , función de anotación, 81
$\mathfrak{A}, \mathfrak{B}$, función de asignación, 6, 8	$Id_{\mathcal{B}}$, función identidad de $\mathcal{B}$, 78
$\mathfrak{B}_{\mathfrak{P}}$, base de Herbrand, 52	$\sqcap$, ínf de un retículo, 79
$\mathcal{B}$, birretículo, 78	ínf , ínfimo de un conjunto, 56
$\wedge, \vee, \otimes, \oplus$, operaciones infinitas entre birretículos, 79	$\mathfrak{I}_{\mathfrak{H}}$, interpretación de Herbrand, 53
$\wedge, \vee, \otimes, \oplus$, operaciones entre birretículos, 79	$\mathfrak{I}$, interpretación anotada para predicados, 82
$\mathcal{S} \models \mathcal{F}$, $\mathcal{F}$ es consecuencia lógica de $\mathcal{S}$, 11	$\mathfrak{I}$, interpretación de Primer Orden, 5
$\mathcal{C}$, conjunto de constantes, 2	$\mathfrak{X}$, interpretación anotada para birretículos, 82
$\mathcal{D}$, dominio de Primer Orden, 5	$\mathfrak{L}$, lenguaje anotado en birretículos, 82
$\equiv$, equivalencia lógica, 11	$\mathcal{L}(\mathcal{C}, \mathcal{R}, \mathcal{F})$, lenguaje de primer orden anotado, 77 de primer orden, 2
$\mathfrak{E}$, estructura de Primer Orden, 5	Q_{τ} , lógica de Primer Orden anotada, 77
$\mathfrak{E}_{\mathfrak{H}}$, estructura de Herbrand, 53, 54	P_{τ} , lógica Proposicional anotada, 76
$\mathfrak{ES}_{\mathfrak{P}}$, conjunto de éxitos, 53, 89	$\text{mpf}_{\mathcal{T}}$, menor punto fijo de $\mathcal{T}$, 64
$\mathcal{E}$, expresión, 23	$\mathfrak{M}$, modelo anotado, 82
■ , fin de una demostración	$\text{mm}\mathfrak{H}_{\mathfrak{P}}$, modelo mínimo de Herbrand de $\mathfrak{P}$, 61, 62
◆ , fin de un ejemplo	$\neg$, negación de un birretículo, 79
$\mathcal{F}, \mathcal{G}$, fórmulas de Primer Orden, 3	$\mathcal{S} \not\models \mathcal{F}$, $\mathcal{F}$ no es consecuencia lógica de $\mathcal{S}$, 42
$\mathfrak{F}$, fórmula anotada, 82	
$\mathcal{S}$, conjunto de fórmulas de primer orden, 10 estándar de cláusulas, 18	

$\mathcal{T}_{\mathfrak{P}}$, operador de consecuencia inmediata, 64

$\leq_c$, orden de conocimiento, 79

$\leq_v$, orden de verdad, 79

$\wp(\)$, conjunto de partes o conjunto potencia, 56

$\mathfrak{P}$, programa lógico anotado en birretículos, 86

$\mathfrak{P}$, programa lógico
de Horn, 22
de Primer Orden, 21, 22
definido, 22
indefinido, 22

$\mathcal{R}$, símbolo de relación, 2

$\mathcal{R}$, conjunto de símbolos de relación,
2

$\mathcal{R}_{\langle \lambda \rangle}$, relación anotada, 78

θ , respuesta, 41
correcta, 41

τ , retículo, 76

$\cup$, sup de un retículo, 79

$\sup$, supremo de un conjunto, 56

$\epsilon, \{ \}$, sustitución identidad, 23

σ, ρ, γ , sustitución, 23

$\mathcal{T}$, conjunto de términos, 23

umg , unificador más general, 28

$\mathfrak{U}_{\mathfrak{P}}$, universo de Herbrand, 50

$\square$, cláusula vacía, 21, 89

$\models \mathcal{F}$, fórmula válida $\mathcal{F}$, 9

$\mathcal{V}$, conjunto de variables, 2

Índice alfabético

A

- Alfabeto
 - de Primer Orden, 1
 - anotado, 77
- Algoritmo
 - de Skolem, *see* Skolemización
 - FNC, 12
 - FNCP, 14
 - SLD, 43
- Asignación, 6
 - v – variantes, 8
 - de satisfacción, 9
- Átomo, 3
 - anotado, 78, 82
 - elegido, 43

B

- Base de Herbrand, 52
- Birretículo, 78
 - operaciones, 79

C

- Cláusula, 18
 - cabeza de, 20
 - central, 34
 - cuerpo de, 20, 87
 - definida, 21
 - hecho, 40
 - objetivo, 40
 - regla, 40
 - Horn, 21
 - indefinida, 21
 - lateral, 34
 - mixta, 21
 - negativa, 20
 - padre, 29, 31
 - positiva, 20
 - unitaria, 20
 - positiva, 20
 - vacía, 21, 40, 89

Completez

- Resolución de Primer Orden, 32
- Resolución lineal, 36
- Resolución-SLD, 75, 100
 - anotada, 100

Composición, *see* Sustitución compo- sición

Conjunto

- de éxitos, 47
- de éxitos de $\mathfrak{P}$, 53
 - anotado, 89
- de partes, 56
- estándar de cláusulas, 18
- potencia, 56

Conjunto de éxitos, 73, 100

Consecuencia lógica, 11

Constante, 2

D

- Derivación, 32
 - lineal, 34
 - SLD, 43
 - algoritmo de, 43
 - anotada, 89
 - finita, 44
 - infinita, 44
 - longitud de, 44

Dominio, 5, 53

E

- Equivalencia lógica, 11
- Estandarización, 30
- Estructura
 - de Herbrand, 53, 54
 - de Primer Orden, 5
- Éxito, 44, 47, 89
- Éxitos
 - Conjunto de, 47, 53, 73, 89, 100
- Expresión, 23
 - unificables, 27

F

Factor, 31
 FNC, *see* Forma normal conjuntiva
 FNCP, *see* Forma normal conjuntiva
 prenexa

Forma

 clausal, *see* normal conjuntiva
 normal
 conjuntiva, 11
 conjuntiva prenexa, 13
 de Skolem, 16
 prenexa, 13

Fórmula, 3

 anotada, 78, 82
 atómica, 3
 básica, 3, 8
 bien formada, 3
 cerrada, 4
 contradictoria, *see* insatisfacible
 equivalentes lógicamente, 11
 falsificable, *see* no-válida
 insatisfacible, 10
 no-válida, 10
 satisfacible, 9
 universalmente valida, *see* válida
 válida, 9

Fracaso, 44

Función

 continua, 63
 de asignación, 6
 de interpretación, 5
 de Skolem, 17
 monótona, 63

H

$\mathfrak{H}$ -estructura, 53, 54

$\mathfrak{H}$ -interpretación, 53

$\mathfrak{H}$ -modelo, 57

 Hecho, 20, 40

 Herbrand, 49

 Base de, 52

 Estructura de, 53, 54

 Interpretación de, 53

 Modelo de, 50, 57

 Modelo Mínimo de, 61, 62, 73, 100

 Teoría de, 49

 Teorema de, 59

 Universo de, 50

Horn

 Cláusula de, 21

 Programa de, 22

I

 Instancia, 24

 básica, 24

Interpretación

 anotada

 birretículos, 82

 predicados, 82

Interpretación

 de Herbrand, 53

 de Primer Orden, 5

K

Kleene

 Teorema de, 69

L

Lenguaje

 anotado, 77

 en birretículos, 82

 de primer orden, 2

 anotado, 77

Literal, 11

 complementario, 11

 negativo, 11

 positivo, 11

 resuelto, 29

Longitud

 derivación-SLD, 44

M

 Matriz, 13

 Menor punto fijo, 64

Modelo

 de Herbrand, 50, 57

 de Primer Orden, 9

- Mínimo de Herbrand, [61](#), [62](#), [73](#), [100](#)
- N**
- Negación
 Birretículos, [79](#)
- O**
- Objetivo, [20](#), [40](#)
- Operador de consecuencia inmediata, [64](#)
- Orden
 de conocimiento, [79](#)
 de verdad, [79](#)
- P**
- Predicado, *see* Símbolos de relación
- Prefijo, [13](#)
- Principio de Resolución, *see* Resolución
- Programa lógico
 de Primer Orden, [21](#), [22](#)
 definido, [22](#)
 Horn, [22](#)
 indefinido, [22](#)
- Programa lógico anotado
 en birretículos, [86](#)
- Punto fijo, [64](#)
 Menor, [64](#)
- R**
- Refutación, [32](#)
 lineal, [34](#)
 SLD, [47](#)
 anotada, [89](#)
 sin restricciones, [73](#)
- Regla, [40](#)
 de Factorización, [31](#)
 de búsqueda, [37](#), [41](#)
 de computación, [37](#), [41](#)
 de Resolución, [31](#)
 de Resolución Binaria, [29](#)
- Relación, [2](#)
 anotada, [78](#)
- Resolución, [28](#)
 de Primer Orden
 completez, [32](#)
 validez, [32](#)
 lineal, [33](#)
 completez, [36](#)
 derivación, [34](#)
 Regla de, [31](#)
 SLD, [38](#), [40](#), [42](#)
 anotada en $\mathcal{B}$, [87](#), [95](#)
 completez, [75](#)
 completez anotada, [100](#)
 validez, [72](#), [96](#)
- Resolvente, [31](#)
 binaria, [29](#), [42](#)
- Respuesta, [41](#), [87](#)
 anotada, [87](#)
 calculada, [48](#), [72](#), [96](#)
 anotada, [89](#), [96](#)
 correcta, [41](#), [48](#), [72](#), [87](#), [96](#)
 anotada, [96](#)
- Retículo completo, [56](#)
- S**
- Sentencia, [4](#)
- Símbolos
 de función, [2](#)
 de relación, [2](#)
- Skolem
 Algoritmo de, [16](#)
 Forma normal de, [16](#)
 Función de, [17](#)
 Teorema de, [18](#)
- Skolemización, [16](#)
- SLD, *see* Resolución-SLD
- Soporte, [23](#)
- Sustitución, [23](#)
 composición, [24](#)
 identidad, [23](#), [26](#)
 más general, [26](#)
 respuesta, [41](#), [87](#)
 calculada, [48](#)
 calculada anotada, [89](#)

correcta, [41](#), [87](#)
 Soporte, [23](#)
 unificador, [27](#)

T

Teorema

de Herbrand, [59](#)
 de Kleene, [69](#)

Término, [2](#), [23](#)

básico, [2](#), [6](#)
 de anotación, [81](#)

U

umg, *see* unificador más general

Unificador, [27](#)

más general, [28](#)

Universo, *see* Dominio

Universo de Herbrand, [50](#)

V

Validez

Resolución de Primer Orden, [32](#)

Resolución-SLD, [72](#), [96](#)

anotada, [96](#)

Variable, [2](#), [23](#)

libre, [4](#)

ligada, [4](#)

v – variantes, *see* Asignación