Semánticas de punto fijo para programas anotados. Una realización por $\mho$-resolución

Carlos Ernesto Ramírez[***]

5 de abril de 2006

Resumen

En programación lógica clásica, una de las herramientas conceptuales más poderosas consiste en la interpretación semántica de un conjunto de fórmulas como funciones sobre conjuntos ordenados con ciertas propiedades estructurales. Tales funciones en combinación con teoremas de punto fijos como el de Knaster-Tarsky permite resolver ecuaciones de definición en la semántica declarativa de un programa lógico, dando origen a una semántica completa para el programa. En este trabajo, a través de un operador de consecuencia basado en una técnica de hiper-resolución anotada conocida como $\mho$-resolución, introducimos una semántica de punto fijo para programas lógicos anotados.

Introducción

Con el fin de construir una interpretación semántica completa de un programa lógico se introduce una definición ecuacional de los procedimientos e iteraciones que debe realizar un programa. Esto lleva a que la resolución de tales ecuaciones consiste en encontrar los puntos fijos para una ecuación de la forma $f(p) = p$ para alguna función que cumple ciertas propiedades

[*]Departamento de Matemáticas y Estadística. Universidad ICESI Cali.
[**]caramirez@correo.icesi.edu.co

especiales. Una vez hecho esto, se garantiza una interpretación para cada instrucción de un programa. En el caso de programas lógicos, las instrucciones se caracterizan por la introducción de operadores que representan el hecho de que una sentencia es consecuencia lógica de un conjunto de premisas. Sin embargo, si se introducen anotaciones ¿es posible obtener esta clase de operadores con propiedades adecuadas? más aún ¿permiten estos operadores utilizar teoremas de punto fijo que garanticen una semántica completa para programas anotados?. Para responde a tales preguntas presentamos en la primera sección, una rápida introducción a los fundamentos de la programación lógica junto con un refinamiento del principio de resolución conocido como hiper-resolución. La sección 2 se dedica a una presentación elemental de las semánticas de punto fijo. Puesto que nuestras anotaciones se realizan sobre retículos, en la sección 3 se presentan algunas definiciones sobre retículos particulares, necesarias para introducir el operador de $\mathfrak{U}$-resolución. En la sección 4 se demuestra que el operador de $\mathfrak{U}$-resolución es un operador de clausura y por lo tanto es posible definir un operador de consecuencia inmediata, con lo que finalmente se construye una semántica de punto fijo para el caso de programas lógicos anotados.

1. Resolución e Hiper-resolución

El principio de resolución es implementado sobre un conjunto especial de expresiones de L.P.O cuya obtención se realiza buscando lo que se conoce como *formas normales* que permiten obtener una expresión no necesariamente equivalente a un conjunto de términos $\mathfrak{T}$ pero cuya satisfacibilidad se cumple siempre que la forma normal $\mathcal{F}_n(\mathfrak{T})$ es satisfactible y viceversa. En últimas lo que se obtiene es una conjunción de disyunción de literales o FNC donde FNC $= \bigwedge_{i\leq n} \xi_i$ y para cada i, $\xi_i = \bigvee_{j\leq m_i} L_{ij}$ donde para todo i, j $L_{i,j}$ es un literal, es decir, una fórmula atómica o su negación. Las fórmulas $\xi_i = \bigvee_{j\leq m_i} L_{ij}$ se denominan cláusulas. En forma más general tenemos la siguiente definición.

Definición 1.1 *Diremos que una forma normal de Skolem es una **cláusula** si es de la forma*

$$\forall X_1 \dots X_n (\bigvee_{i=1}^{n} L_i)$$

en donde cada L_i es un literal y X_i son variables libres en la disyunción $\bigvee_{i=1}^{n} L_i$.

Así pues, el principio de resolución es una regla de inferencia aplicada a cláusulas, utilizando para ello la idea de par complementario, esto es, P es el complemento de $\neg P$ y viceversa. La disyunción de estas dos cláusulas produce la cláusula vacía $\square$ o de manera más general, realizando la unificación de términos de tal forma que sea posible encontrar una sustitución con la cual realizar la disyunción de pares complementarios. La estrategia de resolución para comprobar la validez de predicados en L.P.O consiste en negar la tesis y obtener la forma clausular del conjunto de predicados eliminando cuantificadores existenciales y universales (Skolemlización). Si la cláusula vacía $\square$ es obtenida por resolución se comprueba entonces la satisfacibilidad de la tesis original. Esto se conoce como razonamiento por refutación y es fundamental en programación lógica, pues también permite la obtención de respuestas y verificar la existencia de una instancia que responda a la pregunta. Existen algoritmos de complejidad lineal para obtener la forma FNC de un conjunto de predicados así como para obtener la forma de Skolem. Sin embargo, puesto que la complejidad del principio de Resolución clásico es exponencial, se han elaborado variaciones que resultan en complejidad polinomial de grado no mayor que 3.

Las formas normales de Skolem presentan la ventaja de ser formulas cerradas y cuantificadas universalmente. Tal ventaja se refleja en una estructura muy especial a saber:

Definición 1.2 (Estructura de Herbrand) *Sea Σ una conjunto de símbolos de función y predicado con al menos una constante. Una **estructura de Herbrand** es una Σ-interpretación $\mathfrak{U}$ tal que el universo de $\mathfrak{U}$ es el conjunto de términos cerrados de Σ, esto es $\mathfrak{U} = \{t \mid t \text{ es un } \Sigma\text{-término cerrado}\}$, llamado el **Universo de Herbrand**. La interpretación de cada símbolo constante $c \in \Sigma$ es $c^{\mathfrak{U}} = c$ y la interpretación de los símbolos funcionales es :*

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

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

Debe observarse que lo único que no está determinado son los símbolos de predicado y que los símbolos de constante y función se interpretan como ellos mismos. De hecho la interpretación en una estructura de Herbrand consiste en realizar sustituciones por términos cerrados y los términos cerrados son interpretados como ellos mismos.

La utilidad de la estructura de Herbrand, radica en que permite determinar un universo de interpretación en el cual es posible garantizar la insatisfaciblidad de un conjunto de cláusulas. En principio, tal objetivo implicaría determinar que dada cualquier interpretación v, más aun, para cualquier dominio, no es posible satisfacer un conjunto de cláusulas. Obviamente que tal tarea es de por más imposible. Sin embargo en el caso de la estructura de Herbrand, se puede demostrar que un conjunto de cláusulas es insatisfacible si y solo si las cláusulas resultan falsas para las interpretaciones sobre el universo de Herbrand. Por lo tanto, el problema de determinar si un conjunto de cláusulas es insatisfacible se reduce a explorar la interpretaciones sobre el universo de Herbrand, labor que se puede realizar de forma sistemática. En efecto:

Teorema 1.3 (Teorema de Herbrand) *Un conjunto de cláusulas S es insatisfacible si y solo si existe un número finito de sustituciones de las variables de las cláusulas de S sobre el Universo de Herbrand que son insatisfacibles.*

Obsérvese que la determinación de la insatisfacibilidad del conjunto citado en el teorema anterior se puede realizar sistemáticamente con métodos de la lógica proposicional transformando así un problema de L.P.O en una verificación sobre lógica proposicional.

Ahora, como es de esperarse por la indecidibilidad del problema de satisfacción para L.P.O, el método de resolución no comprueba la satisfacibilidad para cualquier conjunto de predicados. Pero si se puede asegurar lo siguiente:

Teorema 1.4 (Teorema de resolución para el cálculo de predicados) *Un conjunto de Cláusulas M es no satisfactible sii $\square \in R^*(M)$*[1].

Demostración

Puede consultar la demostración de este importante resultado de lógica computacional en [5]

[1] $R^*(M)$ es el conjunto de todas las resolventes obtenidas de M.

1.1. Hiper-resolución

La hiperresolución es una variante del método de resolución que permite obtener un resolvente de forma simultanea saltando varias ramas del árbol de derivación. Para ello ubica los componentes de la cláusula de una forma conveniente para realizar esta poda múltiple.

Definición 1.5 *Una cláusula es* ***positiva*** *si todas sus literales son átomos no negados. Es* ***negativa*** *si todas sus literales son átomos negados y es mixta en cualquier otro caso.*

Dado que una cláusula es cerrada puede ser simbolizada con la expresión $\forall \bigvee_{i=1}^{n} L_i$ o simplemente por $\bigvee_{i=1}^{n} L_i$ siempre que el contexto indique claramente que se trata de una cláusula. Por la conmutatividad del conector $\vee$ la expresión $\bigvee_{i=1}^{n} L_i$ puede ser reordenada de tal forma que aparezcan en primer lugar los literales positivos y luego los negativos, consiguiendo así que:

$$\bigvee_{i=1}^{n} L_i = (\bigvee_{i=1}^{k} A_i) \vee (\bigvee_{i=1}^{l} \neg B_i)$$

con $n = k + l$ y $A_i \dots A_k, B_i \dots B_l$ literales positivos. Ahora bien, como

$$\bigvee_{i=1}^{l} \neg B_i \equiv \neg \bigwedge_{i=1}^{l} B_i$$

una cláusula se puede escribir como $(\bigvee_{i=1}^{k} A_i) \vee \neg(\bigwedge_{i=1}^{l} B_i)$ lo que equivale lógicamente a la expresión

$$(\bigwedge_{i=1}^{l} B_i) \rightarrow (\bigvee_{i=1}^{k} A_)$$

lo que motivo la siguiente notación usual en programación lógica para representar una cláusula:

$$A_1, \dots, A_k \leftarrow B_1, \dots, B_l \text{ con } k, p \geq 0$$

notación en la que se escribe en primer lugar el consecuente y en segundo lugar el antecedente con el objetivo de enfatizar más la atención sobre lo que podemos derivar de lo que conocemos. Debe aclararse que en esta notación todos los literales son positivos y que la coma que separa los literales de la izquierda de la implicación representa el $\vee$ mientras que la coma que separa los de la derecha representa el $\wedge$. A la parte izquierda de $\leftarrow$ se le llama **cabeza** y la parte derecha **cuerpo** de la cláusula.

Una cláusula de la forma $A \leftarrow B_1, \ldots, B_l$ con $A, B_1, \ldots, B_l$ literales positivos se conoce como cláusula de **Horn** es decir, un conjunto de cláusulas con a la más un literal positivo. Así, un programa de **Horn** no es más que un conjunto finito de cláusulas de Horn.

El objetivo de la regla de hiperresolución es producir una cláusula positiva a partir de un conjunto de cláusulas, una de ellas negativa o mixta y las demás positivos.

En general, la estructura de la regla de hiper-resolución viene dada por:

Hiper-resolución:

$$\neg K_1 \ldots \neg K_n$$

$$\frac{L_1 \ldots L_n \neg L_{n+1} \ldots L_{n+m}}{\neg \sigma L_{n+1} \ldots \neg \sigma L_{n+m}}$$

donde σ es el u.m.g para los átomos $K_1, \ldots, K_n$ y $L_1, \ldots, L_n$

Puede demostrarse como en el teorema 1.4 la completitud correspondiente a la hiper-resolución con complejidad de tipo polinomial.

2. Introducción a semánticas de punto fijo

Dado que los modelos de Herbrand son subconjuntos de la base de Herbrand, la relación de inclusión conjuntista $\subseteq$ establece un orden en el conjunto de los modelos de Herbrand. Existe entonces un modelo de Herbrand minimal al que llamaremos modelo mínimo de Herbrand del programa definido.

Teorema 2.1 *Sea $F = \{I_1, I_2, \ldots\}$ una familia no vacía de modelos de Herbrand de un programa de Horn P. Entonces*

$$I = \bigcap_{I_i \in F} I_i$$

es un modelo de Herbrand de P.

Demostración
Supongamos que I no es un modelo de P. Luego, debe existir una instancia básica de una cláusula de P que es falsa bajo I, es decir:

$$\not\models_I (A \leftarrow B_i, \ldots, B_l)$$

Lo que implica que $I \supseteq \{B_1, \ldots, B_l\}$ pero $I \not\supseteq A$. Por lo tanto, para todo $J \in F$ se cumple que $J \supset \{B_1, \ldots, B_l\}$ y que existe algún $J_0 \in F$ tal que A no es un elemento de J_0. Así que

$$\not\models_{J_0} (A \leftarrow B_i, \ldots, B_l)$$

Lo cual es una contradicción pues J_0 es un modelo de P ■

Este resultado da pie para la siguiente definición de modelo mínimo de Herbrand de un programa de Horn.

Definición 2.2 *Sea P un programa de Horn. El modelo mínimo de Herbrand de P ($mmH(P)$) se define como la intersección de todos los modelos de Herbrand de P.*

Así, $mmH(P)$ es la interpretación que se corresponde con el "modelo deseado". De hecho, podemos demostrar el siguiente teorema.

Teorema 2.3 *El modelo mínimo de Herbrand de un programa de Horn es el conjunto de todos los átomos básicos que son consecuencia lógica del programa. Es decir,*

$$mmH(P) = \{A \in B_H(P) \mid P \models A\}$$

Demostración
[$\supseteq$] Sea $A \in B_H(P) \mid P \models A$. Por definición de consecuencia lógica, A es verdad bajo todos los modelos de P, y por lo tanto, A es verdad bajo $mmH(P)$. Por lo que $A \in mmH(P)$.

[$\subseteq$] Sea $B \in mmH(P)$. Como $mmH(P)$ es la intersección de todos los modelos de Herbrand de P, por lo que B es verdad bajo todo modelo de Herbrand de P, o sea, $B \in M$ para todo M que sea un modelo de Herbrand de P. Supongamos que $\mathcal{I}$ es un modelo de P. La interpretación $\widetilde{\mathcal{I}} = \{A \in B_H(P) / \models_{\mathcal{I}} A\}$ asociada a $\mathcal{I}$ es un modelo de Herbrand de P. Por lo tanto $B \in \widetilde{\mathcal{I}}$ y así $\models_{\mathcal{I}} B$, concluyéndose que $\mathcal{I}$ es un modelo de B. Luego, todo modelo de P es modelo de B, esto es, $P \models B$, implicando que $B \in \{A \in B_H(P) \mid P \models A\}$ ■

2.1. Aproximación de punto fijo a la semántica de programas de Horn

Podemos ahora construir un modelo mínimo de Herbrand a través de un proceso iterativo:

1. Las instancias básicas de los hechos [2] deben estar en todo modelo de Herbrand, y por lo tanto también en el modelo mínimo. Llamemos a este conjunto M_0, el cual será un subconjunto de $mmH(P)$

2. Dada la siguiente instancia básica de una regla $A \leftarrow B_1, \ldots, B_l$ con $l \geq 1$, si $B_1, \ldots, B_l$ pertenecen a $mmH(P)$, entonces A también tiene que pertenecer. Luego, si $B_i \in M_k$ con $1 \leq i \leq l$ definimos

$$M_{k+1} = \{A \mid A \leftarrow B_1, \ldots, B_l \text{ es una instancia básica de } P \text{ y} \\ B_i \in M_k 1 \leq i \leq l\} \cup M_k$$

y se tendrá que M_{k+1} es un subconjunto de $mmH(P)$.

Con base en esta idea intuitiva daremos una definición formal del anterior proceso a través del operador de consecuencia inmediata.

Definición 2.4 *Dado un programa de Horn P, sea $Basicas(P)$ el conjunto de todas las instancias básicas de cláusulas de P. Definimos el operador de consecuencia inmediata de P representado por T_P a la aplicación de $\mathcal{P}(B_H(P))$ a $\mathcal{P}(B_H(P))$ de las partes de la base de Herbrand definida así:*

$$T_P(I) = \{A \mid (A \leftarrow B_1, \ldots, B_l) \in Basicas(P) \text{ y } \{B_1, \ldots, B_l\} \subseteq I\}$$

Con la definición anterior $(\mathcal{P}(B_H(P)), \subseteq)$ tiene estructura de retículo y sobre ella se define la siguiente sucesión creciente

$$T_P \uparrow 0 = \varnothing$$
$$T_P \uparrow (i+1) = T_P(T_P \uparrow i)$$
$$T_P \uparrow \omega = \bigcup_{i=0}^{\infty} T_P \uparrow i$$

[2] cláusulas con el cuerpo vacío

siendo $T_P \uparrow 0 \subseteq T_P \uparrow 1 \subseteq T_P \uparrow 2 \subseteq \ldots \subseteq T_P \uparrow \omega$. Obsérvese que cuando para algún i se cumple que $T_P \uparrow i = T_P \uparrow (i-1)$, entonces $T_P \uparrow i$ es un punto fijo de T_P y se cumple que $T_P \uparrow \omega = T_P \uparrow i$.

A partir de la definición de T_P, es evidente que dada una secuencia infinita de interpretaciones $I_0 \subseteq I_1 \subseteq I_2 \ldots$ entonces

$$T_P(\bigsqcup_{i=1}^{\omega} I_i) \subseteq \bigsqcup_{i=1}^{\omega} T_P(I_i)$$

donde $\bigsqcup_{i=1}^{\omega}$ se refiere al supremo sobre $\mathcal{P}(B_H(P))$. Tal propiedad de conservación de la mínima cota superior sobre cadenas es conocida en la literatura informática como continuidad y monotonía. En forma más general damos la siguiente definición.

Definición 2.5

- *Una función $f : D \to E$ entre conjuntos parcialmente ordenados es* ***monótona*** *sii $\forall d, d' \in D$ se tiene que $d \leq d' \Rightarrow f(d) \leq f(d')$*

- *Si D y E son posets, f es* ***continua*** *sii es monótona y preserva supremos sobre cadenas, es decir, para toda cadena $d_0 \leq d_1 \leq \ldots$ en D se cumple que*
$$f(\bigsqcup_{i \geq 0} d_i) = \bigsqcup_{i \geq 0} f(d_i)$$

Proposición 2.1 *Si f es un operador continuo, entonces es monótono.*

Demostración
Sea $f : D \to E$ un operador continuo y sean $a_1, a_2 \in D$ tal que $a_1 \leq a_2$. Entonces $\{a_1, a_2\}$ es una cadena. Luego

$$\begin{aligned} f(\bigsqcup\{a_1, a_2\}) &= f(a_2) \\ &= \bigsqcup\{f(a_1), f(a_2)\} \end{aligned}$$

Es decir, $f(a_1) \leq f(a_2)$ $\blacksquare$

Igualmente, el siguiente lema resulta inmediato:

Lema 2.6 *Dado un programa de Horn P, sea $S \subseteq \mathcal{P}(B_H(P))$. Entonces, $\{B_1,\ldots,B_l\} \subseteq \sup(S)$ si y solo si existe $I \in S$ tal que $\{B_1,\ldots,B_l\} \subseteq I$*

Lema 2.7 *Dado un programa de Horn P, para toda $I \in \mathcal{P}(B_H(P))$ se cumple que $\models_I P$ si y solo si $T_P(I) \subseteq I$*

Demostración
[$\Rightarrow$] Sea $A \in T_P(I)$. Por definición de T_P existe una instancia básica de una cláusula de P que tiene como cabeza a A tal que $\{B_1,\ldots,B_l\} \subseteq I$. Dado que I es un modelo de P, se tiene que $A \in I$ y por lo tanto $T_P(I) \subseteq I$.

[$\Leftarrow$] Supongamos que I no es un modelo de P. Entonces debe existir una instancia básica de alguna cláusula de P que es falsa bajo I. Digamos que esta cláusula es $A \leftarrow B_1,\ldots,B_l$, así, $\{B_1,\ldots,B_l\} \subseteq I$ y $A \notin I$. Se obtiene entonces que $A \in T_P(I)$, $T_P(I) \subseteq I$ y a su vez $A \notin I$ lo que es un contradicción, por lo tanto I debe ser un modelo de P ■

Si $T_P(I) \subseteq I$ diremos que I es un **pre-punto fijo** de T_P. Así, los modelos de Herbrand de un programa de Horn son los pre-puntos fijos del operador de consecuencia inmediata asociado. Podemos ahora mostrar que el modelo mínimo de Herbrand de un programa de Horn P es justamente un punto fijo especial del operador T_P. Para ello enunciamos un teorema clásico de la teoría de operadores monótonos:

Teorema 2.8 (Teorema de Punto fijo de Knaster y Tarsky) *Un operador continuo T tiene un mínimo punto fijo ($mpf(T)$) que coincide con un pre-punto fijo mínimo.*

Demostración
En esencia debemos mostrar que:

1. T posee un mínimo pre-punto fijo dado por:

$$mpf(T) = \bigsqcup_{i \geq 0} T^i(\varnothing)$$

2. $mpf(T)$ es un punto fijo de T, es decir, satisface: $T(mpf(T)) = mpf(T)$ y por lo tanto es el mínimo punto fijo de T.

Debe aclararse que la notación $T^i(\varnothing)$ usada en el teorema está definida por

$$(1)\ \begin{cases} T^0(\varnothing) & =_{def} \varnothing \\ T^{i+1}(\varnothing) & =_{def} T(T^i(\varnothing)). \end{cases}$$

Observe que como para todo $d \in \mathcal{P}(D)$ se cumple que $\varnothing \subseteq d$, uno tiene que $T^0(\varnothing) = \varnothing \subseteq T^1(\varnothing)$;y por la monotonía de T se tiene que :

$$T^i(\varnothing) \subseteq T^{i+1}(\varnothing) \Rightarrow T^{i+1}(\varnothing) = T(T^i(\varnothing)) \subseteq T(T^{i+1}(\varnothing)) = T^{i+2}(\varnothing).$$

Por lo tanto, por inducción sobre $i \in \mathbb{N}$ se concluye que $\forall i \in \mathbb{N}$, $T^i(\varnothing) \subseteq T^{i+1}(\varnothing)$.

En otras palabras, los elementos $T^i(\varnothing)$ forman una cadena en D y como D es un poset, el supremo usado para definir $mpf(T)$ tiene sentido.

En primer lugar note que

$$\begin{aligned} T(mpf(T)) &= T(\bigsqcup_{i\geq 0} T^i(\varnothing)) \\ &= \bigsqcup_{i\geq 0} T(T^i(\varnothing)) && \text{Por continuidad de } T \\ &= \bigsqcup_{i\geq 0} T^{i+1}(\varnothing) && \text{Por (1)} \\ &= \bigsqcup_{i\geq 0} T^i(\varnothing) && \text{Pues } \bigsqcup_{i\geq 0} d_i = \bigsqcup_{i\geq 0} d_{I+i} \text{ para cualquier } I \in \mathbb{N} \\ &= mpf(T) \end{aligned}$$

Así $mpf(T)$ es realmente un punto fijo de T y por lo tanto $T(mpf(T)) \subseteq mpf(T)$. Para verificar la segunda condición necesaria para un mínimo prepunto fijo, supongamos que $d \in D$ satisface $T(d) \subseteq d$. Como $\varnothing$ es el menor elemento en D entonces:

$$T^0(\varnothing) = \varnothing \subseteq d$$

y

$$\begin{aligned} T^i(\varnothing) \subseteq d \Rightarrow T^{i+1}(\varnothing) = T(T^i(\varnothing)) &\subseteq T(d) && \text{Por la monotonía de T} \\ &\subseteq d && \text{Hipótesis sobre } d \end{aligned}$$

Ahora, por inducción sobre $i \in \mathbb{N}$ se tiene que $\forall i \in \mathbb{N}, T^i(\varnothing) \subseteq d$. Luego d es una cota superior para la cadena y por lo tanto cubre al menor elemento de la cadena, es decir:

$$mpf(T) = \bigsqcup_{i\geq 0} T^i(\varnothing) \subseteq d$$

lo que demuestra la segunda parte del teorema $\blacksquare$

Este teorema establece un resultado clave para permitir una semántica denotacional de programas que involucran formas recursivas.

Teorema 2.9 *Dado un programa de Horn,* $mmH(P) = mpf(P) = T_P \uparrow \omega$

Demostración

$mmH(P)$	Por el teorema 2.3
$= \inf\{I \mid I \quad es \quad modelo \quad de \quad Herbrand \quad de \quad P\}$	Por el lema 2.7
$= \inf\{I \mid T_P(I) \subseteq I\}$	Teorema 2.8
$= mpf(T_P)$	puesto que T_P es continua
$= T_P \uparrow \omega \quad \blacksquare$	

Se logra con esto, construir una semántica denotacional completa para un programa de Horn.

3. ℧-resolución y retículos ordinarios o co-Brower)

Suponemos que el lector esta familiarizado con la noción de retículo completo.Sin embargo puede consultar una detallada introducción en [1].

Definición 3.1 *Sea $\mathcal{B}$ un retículo distributivo y completo y sea $A \subseteq \mathcal{B}$, definimos*

$$\uparrow A = \{\rho \in \mathcal{B} \mid (\exists \mu \in A)\mu \preceq \rho\}$$

De la anterior definición es inmediato que si $\zeta = Sup\{\mu, \rho\}$ entonces $(\uparrow \mu) \cap (\uparrow \rho) = \uparrow \zeta$.

Definición 3.2 *Sean $\mathcal{B}$ y A como en la definición anterior. Entonces A es* ***regular*** *si para algún $\delta \in \mathcal{B}$*[3] *se tiene que $A = \uparrow \delta$ ó $A = (\uparrow \delta)'$. Diremos que δ es el* ***elemento definidor*** *del conjunto A.*

[3] Obsérvese que aquí δ es un elemento de $\mathcal{B}$, no un subconjunto del mismo.

Aunque las lógicas anotadas son paraconsistentes, es posible adecuarlas de tal forma que resulten consistentes.En efecto: tomemos una interpretación I sobre Σ que mapea literales en $\mathcal{B}$. Basta con definir una interpretación Σ-consistente I_c así: $I_c(A_\lambda) = true$ si $I(A) \in \uparrow \lambda$ y $I_c(A_\lambda) = false$ si $I(A) \in (\uparrow \lambda)'$.

Nuestro interés ahora consiste en introducir una especie de negación que de cuenta de la idea de negación clásica pero dentro del contexto de las lógicas anotadas. Más precisamente:

Definición 3.3 *Sea* $\neg : \mathcal{B} \to \mathcal{B}$ *una función inyectiva. Entonces* $\neg$ *es una* ***involución*** *si satisface:*

1 Si $x \preceq_v y$ *entonces* $\neg y \preceq_v \neg x$

2 Si $x \preceq_c y$ *entonces* $\neg x \preceq_c \neg y$

3 $\neg\neg x = x$

En [2] se demuestra un lema muy interesante en el sentido de aclarar la forma como una cláusula con negación transfiere esta negación a una operación de negación sobre un retículo. Presentamos su enunciado:

Lema 3.4 *Sea* I *un interpretación. Entonces* $I \vDash (\exists) \sim A_\lambda$[4] *sii* $I \vDash (\exists) A_{\neg\lambda}$

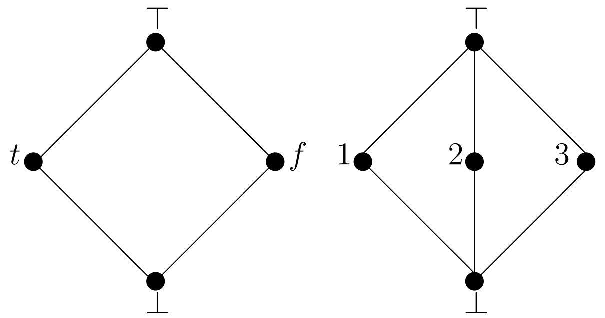
Así que dado un átomo A_λ tenemos que $\sim A_\lambda \equiv A_{\neg\lambda}$

De esta forma queda claro que dado un conjunto de cláusulas anotadas la operación de negación puede proveerse sobre un retículo distributivo y completo. Con esto en mente podemos entonces introducir el operador de hiper- resolución.

Definición 3.5 *Sea* $\mathcal{M}(\mu, \rho) = \{\gamma \in \mathcal{B} \mid Sup\{\gamma, \rho\} \succeq \mu\}$. *El* $\mho$*-operador es definido por* $\mho(\mu, \rho) = Inf\ \mathcal{M}(\mu, \rho)$

Definición 3.6 *Un retículo* $\mathcal{B}$ *es ordinario si* $\mu, \rho \in \mathcal{B}$ *implica que* $\mho(\mu, \rho) \in \mathcal{M}(\mu, \rho)$

[4] $\exists$ se refiere a la clausura existencial de variables libres sobre una fórmula A



$Four$ $\qquad\qquad$ M_5

Ejemplo

Tomemos el retículo $Four$. Si $\mu = \top$ y $\rho = t$, entonces $\mathcal{M}(\mu, \rho) = \{f, \top\}$ Evidentemente $Four$ es ordinario a diferencia de M_5, pues $\mathcal{M}(2, 1) = \{2, 3, \top\}$, pero $\inf\{2, 3, \top\} = \bot$ y $\bot \notin \mathcal{M}(2, 1)$

Es necesario aclarar en este punto el porqué de estas dos definiciones. En primer lugar, para definir la resolvente entre las cláusulas $\sim A_\mu \vee E_1$ y $A_\rho \vee E_2$ (se dice que $\sim A_\mu$ y A_ρ son complementarios) en el caso de que $\mu \leq \rho$ la resolvente está dada por $E_1 \vee E_2$. Así, dos cláusulas son resolubles solo si la anotación del átomo negado es menor o igual que la anotación del átomo no negado. Esta regla se conoce como *resolución anotada*. Para el caso en que se tienen las cláusulas $A_{\mu_1} \vee D_1$ y $A_{\mu_2} \vee D_2$ cuando μ_1 y μ_2 son incomparables, la resolvente está definida por $A_{\sup\{\mu_1, \mu_2\}} \vee D_1 \vee D_2$. Tal regla se conoce como la *regla de deducción*. Sin embargo, sería deseable tener una regla para anotaciones regulares que no dependa de la reducción y en la que se pueda aplicar de alguna manera la regla de resolución anotada. Tal intento presenta dificultades cuando quiere aplicarse a anotaciones regulares. Por ejemplo, Sean $\sim A_\mu$ y A_ρ,y suponga que $\mu \not\leq \rho$. Así. la regla de resolución no es aplicable en este caso. Por ello, para mantener la reducción regular es necesario encontrar una especie de anotación minimal $\mathcal{S}_0$ que contenga a $(\uparrow \mu)' \cap \uparrow \rho$ y así el resolvente de las dos cláusulas tenga a $\mathcal{S}_0$ como su anotación. En [4] se demuestra que tal conjunto minimal sobre las anotaciones esta dado por $(\uparrow \mho(\mu, \rho))'$ por lo que es indispensable que el retículo sea ordinario. Por otra parte tal definición elimina la necesidad de hacer reducción sobre cláusulas anotadas cuando μ y ρ son incomparables ya que por la minimalidad de $(\uparrow \mho(\mu,\rho))'$ la regularidad sobre las anotaciones está garantizada.

En [4] Lu demuestra que los retículos ordinarios son equivalentes a retículos distributivos. Si el retículo es completo y finito la implementación computacional es favorable. Por ello nuestra exigencia de retículos de estas características. Sin embargo tal resultado solo es cierto en el caso de retículos finitos. En la tesis de Muñoz[3] se presenta un contraejemplo que verifica la necesidad de contar con retículos finitos para que la equivalencia se mantenga. Más aún, se presenta una clase de retículos en los cuales es posible obtener una estructura adecuada para lograr reducción sobre cláusulas anotadas con elementos incomparables. Estos retículos son conocidos como retículos co-Brower, que equivalen a retículos ordinarios para el caso finito, por lo que optaremos por usar retículos de esta categoría para realizar las anotaciones de las cláusulas.

Definición 3.7 ($\mho$-resolución) *Sean las cláusulas anotadas* $(\sim A_\mu \vee D_1)$ *y* $(A_\rho \vee D_2)$ *entonces la* $\mho$-***resolvente*** *de las dos cláusulas sobre los literales anotados* $\sim A_\mu$ *y* A_ρ *es* $\sim A_{\mho(\mu,\rho)} \vee D_1 \vee D_2$

Es decir, la $\mho$-resolvente se obtiene para el par complementario de cláusulas anotadas al calcular $(\uparrow \mho(\mu,\rho))'$

La validez y completitud como prueba por refutación del operador de $\mho$-resolución están garantizadas por los siguientes resultados cuya demostración puede consultarse en [4].

Teorema 3.8 (Validez) *La regla de inferencia* $\mho$*-resolución es válida para lógicas anotadas.*

Teorema 3.9 (Completez) *Si* $\mathcal{S}$ *es un conjunto insatisfacible de cláusulas anotadas entonces existe una refutación de* $\mathcal{S}$ *a través del operador de* $\mho$*-resolución*

En [4] se construye un procedimiento de macroinferencia llamado *Hiper-resolución anotada* basado en el operador de $\mho$-resolución el cual puede verse como una regla única que ejecuta de forma simultanea varios pasos de resolución anotada, es decir, conserva el sentido usual de la regla de hiper-resolución en lógica clásica. Los detalles de esta operación son similares a la idea clásica pero con unos cuantos detalles técnicos que no son presentados aquí pues nuestra intención no es mejorar esta técnica, sino más bien usarla como un mecanismo de inferencia lógica para un programa lógico anotado.Así, un literal anotado será de la forma $A : \lambda$. Ver detalles en [4].

Por el lema 3.4, nuestra noción de negación se convertirá en una involución sobre la anotación de la cláusula. Esto es

$$\sim A : \lambda \equiv A : \neg\lambda$$

4. Semántica de punto fijo para programas anotados

podemos caracterizar la regla de $\mho$-resolución como un operador de consecuencia o vinculamiento de tal forma que resulte ser un operador de clausura.

Definición 4.1 *Sea A un conjunto dado, un mapeo C:$\mathcal{P}(A) \to \mathcal{P}(A)$ se dice un **operador de clausura** sobre A si para $X, Y \subseteq A$ satisface:*

$$\begin{array}{ll} C_1 : X \subseteq C(X) & \textit{(Extensivo)} \\ C_2 : C(C(X)) = C(X) & \textit{(Idempotente)} \\ C_3 : X \subseteq Y \quad \textit{implica} \quad C(X) \subseteq C(Y) & \textit{(Monótono)} \end{array}$$

Definición 4.2 *Sea Γ el conjunto de cláusulas anotadas sobre el lenguaje L de Q_τ. $C : \mathcal{P}(\Gamma) \to \mathcal{P}(\Gamma)$ es definido como el operador de $\mho$-resolución, es decir, dado un subconjunto X de Γ, tal que X contenga al menos dos literales complementarios(digamos para A), $C(X) = A_{(\uparrow\mho(\mu,\rho))'} \vee X_0$ donde X_0 es el conjunto que se obtiene al eliminar de X el par complementario para A.*

En general tomaremos los átomos sobre la base de Herbrand, pues basta con realizar instancias básicas sobre las fórmulas,es decir realizar sustituciones cuyo dominio es la base de Herbrand, y el operador funciona igual.

Afirmación 4.1 *Sea C el operador de la definición 4.2. Entonces C es un operador de clausura sobre Γ.*

Demostración
Sean $X, Y \subseteq \Gamma$. Verifiquemos las condiciones de la definición 4.1.

- C_1.

  Por el teorema 3.9 $\mho$-resolución es una regla válida, por lo que $C(X)$ contiene todas las $\mho$-resolventes que se pueden obtener de X. Pero X también es una $\mho$-resolvente cuando $X_0 = \varnothing$ por lo que $X \subseteq C(X)$.

- C_2.

  Basta tomar una interpretación $\Sigma - consistente$ I_c en cuyo caso nos referimos a resolución clásica con lo que la comprobación es inmediata.

- C_3.

  Sean $X \subseteq Y$ y $\widehat{x} \in C(X)$. Luego $\widehat{x}$ es de la forma $\widehat{x} = A_{(\uparrow\mho(\mu,\rho))'} \vee X_0$ con $X_0 \subseteq X$. Puesto que $X \subseteq Y$ entonces existe un $Y_0 \subseteq Y$ tal que $\widehat{x} = A_{(\uparrow\mho(\mu,\rho))'} \vee Y_0$ con $X_0 \subseteq Y_0$. Pero esto implica que $\widehat{x} \in C(Y)$ por lo que $C(X) \subseteq C(Y)$.

En conclusión C es un operador de clausura sobre Γ ■

Definamos ahora una relación de consecuencia $\vdash$ sobre $\mathcal{P}(\Gamma) \times \mathcal{P}(\Gamma)$ determinada así:

$$\Gamma \vdash \alpha \text{ si y solo si } \alpha \in C(\Gamma)$$

En otras palabras Γ vincula α si α es una consecuencia de Γ

Afirmación 4.2 *Sea $\vdash$ como se definió en el párrafo anterior. Entonces $\vdash$ satisface*

1 Si $\alpha \in \Gamma$ entonces $\Gamma \vdash \alpha$

2 Si $\Gamma \subseteq \Delta$ entonces $\Gamma \vdash \alpha$ implica que $\Delta \vdash \alpha$

3 Si $\Gamma \vdash \alpha$ y $\Gamma \cup \{\alpha\} \vdash \beta$ entonces $\Gamma \vdash \beta$

Demostración

1 Inmediato por la definición de $\vdash$.

2 Es una consecuencia la monotonía del operador de clausura C.

3 Por hipótesis $\alpha \in C(\Gamma)$ y $\beta \in C(\Gamma \cup \{\alpha)\}$. Claramente $C(\Gamma \cup \{\alpha\}) = C(\Gamma) \cup C(\{\alpha\})$. Como $\alpha \subseteq C(\{\alpha\}) \in C(\Gamma)$, se tiene que $\beta \in C(\Gamma)$, es decir, $\Gamma \vdash \beta$ ■

Estamos ahora interesados en definir una semántica de punto fijo para nuestro modelo de cómputo. Para ello adoptaremos la siguiente convención: si $B_{0_\mu}, B_{1_{\rho_1}}, \ldots, B_{n_{\rho_l}}$ es un conjunto de cláusulas anotadas, y $A_{\mho(\mu,\rho_1,\ldots,\rho_l)=\gamma\in\mathcal{B}}$[5] es una resolvente obtenida de ese conjunto entonces $A_\gamma \leftarrow B_{0_\mu}, B_{1_{\rho_1}}, \ldots, B_{n_{\rho_l}}$.

En otras palabras,

$$A_\gamma \in C(B_{0_\mu}, B_{1_{\rho_1}}, \ldots, B_{n_{\rho_l}})$$

es decir, según la afirmación 4.2

$$B_{0_\mu}, B_{1_{\rho_1}}, \ldots, B_{n_{\rho_l}} \vdash A_\gamma$$

Con base en lo anterior, podemos definir de manera semejante a como se hizo en la sección 2 un operador de consecuencia inmediata. Es decir:

$$T_P(I)(A) = \sup\{\gamma \mid A_\gamma \leftarrow B_{0_\mu}, B_{1_{\rho_1}}, \ldots, B_{n_{\rho_l}} \text{ es una instancia cerrada de una cláusula en un programa } P \text{ e } I \models B_{0_\mu}, B_{1_{\rho_1}}, \ldots, B_{n_{\rho_l}}\}$$

Para construir iterativamente T_P hacemos uso de una interpretación especial Δ que asigna a todo átomo en $B_H(P)$ el elemento $\perp$.

$$\begin{aligned} T_P \uparrow 0 &= \Delta \\ T_P \uparrow (i+1) &= T_P(T_P \uparrow i) \\ T_P \uparrow \omega &= \sqcup_{i<n} T_P \uparrow i \end{aligned}$$

Por nuestro interés práctico, involucramos solamente los naturales finitos y su límite ordinal ω, sin preocuparnos por los ordinales sucesores. De nuevo, usando nuestra definición Σ- consistente, la continuidad y monotonía de T_P se comprueban de forma directa.

Teorema 4.3 *$T_P \uparrow \omega$ coincide con un mpf de T_P*

[5]Esto denota la operación $\mho(\mho(\ldots\mho(\mho(\mu,\rho_1),\rho_2)\ldots),\rho_l)$.

Demostración
Por la monotonía de T_P se tiene que $T_P \uparrow \omega \subseteq mpf(T_P)$. Así, basta mostrar que $mpf(T_P) \subseteq T_P \uparrow \omega$. Por el lema 2.6 debemos mostrar que $T_P \uparrow \omega$ es un modelo de P. En efecto:

Sea $A_\gamma \leftarrow B_{0_\mu}, B_{1_{\rho_1}}, \ldots, B_{n_{\rho_l}}$ una instancia cerrada de una cláusula en P.

Supongamos que $T_P \uparrow \omega \models B_{0_\mu}, B_{1_{\rho_1}}, \ldots, B_{n_{\rho_l}}$. Entonces, para cualquier átomo A en $(B_H(P))$, como T_P es monotónico tenemos que:

$$T_P \uparrow (0)(A) \subseteq T_P \uparrow (1)(A) \subseteq \ldots$$

Ahora, de la definición de T_P, cada $T_P \uparrow i(A)$ puede ser:

- El menor elemento del retículo ó
- El supremo de un conjunto finito de valores en el retículo referente a las anotaciones en las cabezas de ciertas cláusulas en P.

Obviamente el número de tales conjuntos es finito. De esta forma, la cadena anterior esta compuesta por un número finito de valores y por tanto, contiene su propio supremo.

Así mismo, debe existir un número natural $i < \omega$ tal que

$$T_P \uparrow i \models B_{0_\mu}, B_{1_{\rho_1}}, \ldots, B_{n_{\rho_l}}$$

Por la definición de T_P, se tiene que $T_P \uparrow (i+1)(A) \supseteq \gamma$ y por lo tanto, $T_P \uparrow (i+1) \models A_\gamma$. Ahora, por la definición de $T_P \uparrow \omega$, se tiene que $T_P \uparrow \omega \supseteq T_P \uparrow (i+1)$. Pero $T_P \uparrow (i+1)(A) \supseteq \gamma$ por lo que $T_P \uparrow \omega \supseteq \gamma$, y por lo tanto $T_P \uparrow \omega \models A_\gamma$.

Así, $T_P \uparrow \omega$ es un modelo de T_P y $mpf(T_P) \subseteq T_P \uparrow \omega$ ■

Referencias

[1] Birkhoff Garret. *Lattice Theory.*American Mathematical Society, Vol 25,Rodhe Island, 1940.

[2] Carbogim Daniela Vasconcelos. *Programacao em lógica anotada: Teoría y aplicaciones*, tesis de maestría en matemática aplicada, Sau Paulo, 1996

[3] Muñoz Posso Jhovanny. *Resolución en lógicas anotadas con retículos ordinarios* Tesis de pregrado Matemáticas. Cali, Universidad Del Valle, 2005.

[4] Lu J. *Deduction in multiple-value Logics.* Department of mathematics and computer science Emory University. Atlanta USA, 2003.

[5] Papadimitriu C. *Elements of the Theory of Computation*, Prentice-Hall International Editions, USA, 1981.