Un sistema de restricciones anotado

A annotated constraint system

Carlos Ernesto Ramírez Ovalle
Departamento de matemáticas y estadística Universidad ICESI, Colombia.
carlos.ramirez@correo.icesi.edu.co

Recibido para revisión 26 de marzo de 2009, aceptado 15 de junio de 2009, versión final 9 de julio de 2009

Resumen—En la programación concurrente con restricciones, el concepto básico es el de sistema de restricciones. Aquí presentamos una innovación sobre los sistemas de restricciones al introducir una lógica anotada como componente sintáctico y semántico junto con un motor de inferencia lógica en hiper-resolución anotada. Esto permite incrementar la capacidad expresiva de un modelo basado en satisfacción de restricciones pues es ahora posible incluir información adicional a las restricciones clásicas del modelo. Se extiende así el universo interpretativo de un problema basado en restricciones, al introducir información pertinente que permite la toma de decisiones aún en situaciones contradictorias.

Palabras Clave—Programación por Restricciones, Lógicas Anotadas, Programación Lógica, Sistemas de Restricciones

Abstract—In the concurrent programming with restrictions, the basic concept is precisely this one of restriction system. Here we present an innovation on the restriction system, introducing a n annotated logic as a syntactic and semantic component together with a motor of logical inference in annotated hyper-resolution. This allows to increase the expressive capacity of a model based on satisfaction of restrictions, therefore additional information to the classic restrictions of the model can be included. The interpretative universe of a problem based on restrictions extends in this way, when adding pertinent information giving rise to the decision making still in contradictory situations.

Keywords— Constraint Programming, Annotated Logic, Programming Logic, Constraints Systems.

I. INTRODUCCIÓN

La ntroducción de lógicas anotadas, desarrolladas inicialmente por Kifer y Subrahmanian [4] en el marco de la programación lógica, como mecanismo de razonamiento en un sistema de restricciones dota al sistema de la habilidad de manejar información contradictoria que el caso clásico conduce a la trivialización del sistema de información. Más aún, se está en condiciones de hacer uso de una estructura de orden apropiada que permite obtener resultados mucho más expresivos semánticamente en el sentido de que al poder agregar elementos informativos como tiempo o espacio a las restricciones usuales en la sintaxis de un sistema de restricciones clásico, esta información extra, actúa como elemento de decisión en la solución de un sistema de satisfacción de restricciones. Al asociar esta lógica con mecanismos automáticos de inferencia como resolución, más específicamente, la hiper-resolución anotada presentada por Lu en [5], se dota al sistema de un motor de inferencia lógica eficiente.

El trabajo se presenta en 5 secciones. En la primera sección se introduce el concepto de sistema de restricciones que se ajusta más a los propósitos de este trabajo, el cual es presentado por Fruwirth en [2].En la segunda sección se introducen algunas propiedades básicas y deseables en un sistema de restricciones. En la tercera sección se presenta de forma clara mas no exhaustiva la lógica Q_τ como lógica anotada a utilizar en el sistema de restricciones a construir. En la cuarta sección se desarrolla un sistema de restricciones anotado junto con su mecanismo de inferencia. Por último, la sección cinco se ha destinado a desarrollar un ejemplo factible en un contexto particular y simplificado de optimización combinatoria.

II. SISTEMAS DE RESTRICCIONES

Si bien, es posible presentar diversas definiciones de lo que es un sistema de restricciones, se ha optado por presentar la usada por Fruwirth en [2]. Esto en razón de que tal definición permite elaborar de manera esquemática cada uno de los componentes necesarios en la construcción de un sistema de restricciones.

Definición 2.1: Un sistema de restricciones es una tupla (Σ, D, CT, C) en donde

1. $\sum$ es un conjunto de símbolos de función y de predicados cada uno con su respectiva aridad y que además contiene los símbolos de restricción constantes **true** y **false**, junto con un símbolo de restricción binaria "=" que se usa para igualdad.
2. D es un dominio junto con una interpretación de las funciones y de los símbolos de restricción presentes en
3. CT es un teoría de restricciones que es no vacía y además consistente sobre .
4. C consiste de todas las restricciones permitidas, un conjunto de fórmulas que contiene a las respectivas restricciones **true,false**,"=" y que además es cerrado bajo conjunción y cuantificación existencial.

La inclusión de las constantes **true,false**, y "=" es particularmente importante puesto que en este cálculo lógico se está interesado en la conjunción de restricciones atómicas cuyas variables se encuentran existencialmente cuantificadas. Por ello, que el conjunto de fórmulas en C sea existencialmente cuantificado indica que no importan los nombres de las variables, es decir, C es cerrado bajo renombres de variables. Esto deja ver a C haciendo el papel de la sintaxis del sistema de restricciones. Pero, porque restricciones atómicas? Esto es deseable puesto que restricciones simples y sencillas en su estructura permite que sean resueltas fácilmente (con resolver queremos decir que existen valores para las variables en C que cumplen la restricción, valor que no tiene porque ser único).

III. PROPIEDADES DE UN SISTEMAS DE RESTRICCIONES

A continuación se describen una serie de propiedades deseables en un sistema de restricciones

Notación: Se utilizará el símbolo CT ^% C para indicar la relación de satisfacción lógica, en donde las variables de las restricciones se encuentran cuantificadas existencialmente de forma implícita. Así que a manera de ejemplo ^% debe leerse como: "de la teoría de restricciones se deduce que no existen variables que satisfagan C" indicando con esto que la restricción C no puede satisfacerse. Análogamente ^%indicaría que del sistema de restricciones es posible deducir todas las restricciones impuestas dentro de C, es decir, encontrar valores tales que las condiciones en C se satisfagan para todas las variables en la restricción.

De la definición presentada por Saraswat en [10] extraemos la conveniente propiedad de que CT sea decidible para todas las restricciones permitidas, es decir

Definición 2.2: Sea un sistema de restricciones

1) Una teoría de restricciones CT es ***completa*** si para toda restricción se tiene que ^%ó ^%[1]
2) Una teoría de restricciones CT es de **satisfacción-completa** si para toda restricción se tiene que ^% ó ^%.

La completitud es por lo general una condición bastante fuerte. La parte b) de la definición 2.2.1 permite que un sistema de restricciones sea capaz de operar con todas las restricciones permitidas. Esta propiedad determina la satisfacibilidad de las restricciones válidas en el sistema.

Existen otras propiedades usuales en una sistema de restricciones como lo son la independencia de restricciones negadas o de sistema fuertemente compacto. Para mayor detalle, ver [2].

IV. LA LÓGICA ANOTADA Q_τ

Las lógicas anotadas están dentro del marco de las lógicas no clásicas y surgen dentro del ambiente de la programación lógica. Se puede mostrar que estas lógicas son adecuadas para proveer un esqueleto formal a la programación lógica cuando se presentan inconsistencias sin que esto trivialice el sistema como ocurre en el caso clásico. Si se quiere, aquí está la principal aplicación de las lógicas anotadas, y su principal motivación. La esencia está en representar el conocimiento a través de *átomos anotados*, donde las anotaciones constituyen objetos tanto sintácticos como semánticos.

Se han publicado desarrollos recientes en lógicas anotadas que permiten implementaciones eficientes de técnicas de demostración automática como la hiper-resolución anotada.

Particularmente la lógica anotada Q_τ es una lógica de primer orden definida a partir de un retículo completo y distributivo τ, junto con un operador ~ que es un automorfismo sobre que pretende representar la negación sobre los átomos, y un lenguaje de primer orden estándar sin igualdad.

Los *términos* de esta lógica L son introducidos de la forma usual. Un *átomo* (ordinario) es una expresión de la forma $P(t_1, t_2, ..., t_n)$

donde P es un símbolo de predicado de aridad n. Si $\lambda \in \tau$ entonces un *predicado anotado* es un par $\langle P, \lambda \rangle$ el cual se denotará simplemente por P_λ. Dado un predicado anotado

1 Se insiste en que ^% se usa para simbolizar una relación de satisfacción lógica y no debe confundirse con el operador de vinculamiento %

P_λ de aridad n y n términos $t_1, t_2, ..., t_n$, un *átomo anotado* es una expresión de la forma .

$$P_\lambda(t_1, t_2, ..., t_n) \qquad (1)$$

Las interpretaciones para el lenguaje L se definen de la forma usual. Sin embargo, se distingue a las expresiones de la forma

$$\underbrace{\neg...\neg}_{k}(A_\mu) \qquad (2)$$

donde A es un átomo ordinario, con el nombre de *hiperliterales de orden k,* $k \geq 0$ y se usara la abreviación $\neg^k(A_\mu)$

Si I es una interpretación tendremos que

I^% $P(t_1, t_2, ..., t_n)_\mu$ sii $P^I(t_1^I, t_2^I, ..., t_n^I) \geq \mu$

I^% $\neg A_\mu$ sii I^% $A_{\sim\mu}$ (3)

Los otros casos son estándar.

Se dice además que α es una consecuencia semántica de un conjunto de fórmulas Γ sii para toda interpretación I tal que I^% β para todo $\beta \in \Gamma$, se cumple que I^%. Por otro lado, si , son fórmulas del lenguaje L entonces $\alpha \leftrightarrow \beta =_{def} (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$[2]; para toda fórmula α se define la negación fuerte por $\neg_f \alpha =_{def} (\alpha \rightarrow (\alpha \rightarrow \alpha) \wedge \neg(\alpha \rightarrow \alpha)))$. (4)

El sistema Q_τ consta de la lógica intuicionista y positiva, y por tanto del cálculo proposicional clásico. Los axiomas que competen a la negación y los cuantificadores se presentan en la figura 1

$(\neg_1)$ $(\varphi \rightarrow \psi) \rightarrow ((\varphi \rightarrow \neg\psi) \rightarrow \neg\varphi))$ $(\neg_2)$ $\varphi \rightarrow (\neg\varphi \rightarrow \alpha)$ $(\neg_3)$ $\varphi \vee \neg\varphi$

$(\exists_1)$ $\alpha(t) \rightarrow \exists x\alpha(x)$ $(\exists_2)$ $\frac{\alpha(x)\rightarrow\beta}{\exists x\alpha(x)\rightarrow\beta}$ $(\forall_1)$ $\forall x\alpha(x) \rightarrow \alpha(t)$ $(\forall_2)$ $\frac{\alpha\rightarrow\beta(x)}{\alpha\rightarrow\forall x\beta(x)}$

(τ_1) $(\theta_\perp) \wedge \neg^k(\theta_\mu) \leftrightarrow \neg^{k-1}(\theta_{\sim\mu})$ (τ_2) $(\theta_{\sim\mu}) \rightarrow (\theta_{\sim\lambda})$ donde $\lambda \leq \mu$

(τ_3) Si $\alpha \rightarrow (\theta_{\mu_j}) \forall j \in J$, entonces $\alpha \rightarrow (\theta_\mu)$ donde $\mu = sup\{\mu_j : j \in J\}$

Figura 1. Axiomas para la negación y cuantificación en Q_τ

Dado que τ es un retículo completo, el supremo en (τ_3) esta bien definido. Más aún, cuando es finito, la regla puede ser reemplazada por

donde $\mu = \sup\{u_j : 1 \leq j \leq n\}$.

Las lógicas anotadas Q_τ resultan ser paraconsistentes. Esto es, el lenguaje asociado L es inconsistente y no trivial en relación al operador de negación $\neg$, lo que significa que existe una interpretación I para L tal que I sea al mismo tiempo inconsistente y no trivial. Una interpretación I se dice inconsistente si se satisface A y $\neg A$ para alguna fórmula A de L. Por otro lado I es no trivial si y solo si existe una fórmula A de L tal que A no es satisfecha por I. Puede demostrarse también el teorema de validez para .

La negación fuerte satisface todas las propiedades de la negación clásica, así, la lógica clásica de primer orden esta incluida dentro de Q_τ

V. UN SISTEMA DE RESTRICICONES ANOTADO S.R.A

Un sistema de restricciones establece de manera formal la sintaxis y la semántica con la que se podrán escribir e interpretar expresiones que se convertirán en las restricciones que queremos indicar. Así que un sistema de restricciones permite establecer que restricciones son permitidas, cómo son interpretadas y cuales restricciones son apropiadas para verificar la validez de razonamientos.

El sistema de restricciones que se presenta a continuación esta basado en el sistema de restricciones tipo Herbrand presentado por Fruwirth en [2], pero ahora este se ha extendido al introducir lógicas anotadas. Se referencia este sistema de restricciones como un *sistema de restricciones tipor Herbrand anotado s.r.a.*

A. Símbolos de función y predicado Σ de un s.r.a

Se adoptará a Σ como el conjunto de símbolos de función, predicado y conectivos, tal y como se presentan en [2], junto con los símbolos de restricción $\perp$,,% (elemento mínimo y supremo de un retículo respectivamente),**false, true** , $\doteq$ como igualdad sintáctica y binaria y los elementos de un de un bi-retículo distributivo y completo. Para detalles sobre bi-retículos consulte [6]

2 Se usará $=_{def}$ para indicar igualdad por definición y no por operación

B. Dominio D de un s.r.a

En un sistema de restricciones tipo Herbrand clásico, el dominio está caracterizado por el por el universo de Herbrand como se define usualmente en lógica de primer orden. Sin embargo, dado que el lenguaje base es Q_τ se presenta la siguiente definición

Definición 5.1: Sea Σ una estructura de Herbrand. Una **estructura de Herbrand anotada** es una Σ-interpretación *I* tal que el universo de *I* es el conjunto de términos cerrados anotados de Σ, es decir *I*={*t*%*t* es un término Σ-cerrado y anotado}, llamado **universo de Herbrand anotado.**

Las anotaciones se encuentran ubicadas en un bi-retículo *B* distributivo y completo.

Esta definición es más general que la presentada en [1] pues permita la anotación de átomos cerrados en Σ. La ventaja de esta generalización se reflejará en el desarrollo de *CT* para nuestro ***s.r.a***

C. Teoría de restricciones CT en un s.r.a

CT es tomado como la teoría de restricciones de un sistema tipo Herbrand clásico sobre árboles racionales. A esta teoría se agrega el soporte deductivo para la evaluación de restricciones con lógica anotada basada en la regla de hiper-resolución anotada presentada en [5]. Allí se detalla un mecanismo de deducción automática válido y completo basado en hiper-resolución clásica pero que extiende su operación sobre proposiciones en lógica anotada. Esto permite dar cuenta de una semántica declarativa completa vía semánticas de punto fijo. La demostración de este resultado puede encontrarse en [9].

En un bi-retículo existen dos órdenes (así como dos álgebras con sus respectivas operaciones). Esto permite interpretar uno de estos órdenes $\leq_v$ como un retículo de valores de verdad (o grado de creencia) y por otro lado un retículo de valores de conocimiento $\leq_c$. De esta forma, usando unificación sobre el retículo de conocimiento y el retículo de valores de verdad se obtiene la capacidad de usar hiper-resolución con anotaciones en ambos órdenes de bí-retículo. Como se están usando bi-retículos distributivos y completos, se hará referencia a ambos órdenes en el bí-retículo simplemente como $\leq$

Definición 5.2: Sea *B* un bi-retículo distributivo y completo y sea $A \subseteq B$ se define

$$\uparrow A=\{\rho \in B\% (\exists \mu \in A)\mu \leq \rho\} \quad (5)$$

Definición 4.3.2: Sean *B* y *A* como en la definición 4.3.1. Se dice que *A* es regular si para algún $\delta \in B$ se tiene que $A=\uparrow \delta$ o $A=(\uparrow \delta)^c$ (aquí, c, denota la operación complemento sobre el conjunto).

Aunque las lógicas anotadas son paraconsistentes, la teoría de restricciones *CT* para ***s.r.a*** continúa siendo consistente. En efecto, considere una interpretación *I* sobre Σ que mapea literales en *B*. Basta con definir una interpretación -consistente I_c así:

$$I_c(A_\lambda)=\textbf{true} \text{ si } I(A) \in \uparrow \lambda \quad (6)$$

$$I_c()=\textbf{false} \text{ si } I(A)\ (\uparrow)^c$$

El objetivo ahora consiste en introducir una especie de negación que dé cuenta de la idea de negación clásica pero dentro del contexto de las lógicas anotadas. Tal operación de negación debe estar caracterizada sobre bi-retículos por un operador inyectivo que respete la doble negación, leyes de DeMorgan y complementariedad[3], es decir, una involución. Más aún, en algunos casos es posible definir sobre un bi-retículo apropiado una operación similar a la negación pero sobre el retículo de conocimiento. Este operador unario sobre *B* invierte el orden $\leq_v$ en que se encuentran los elementos,

es decir, si $x \leq_v y$ entonces $\sim y \leq_v \sim x$. De la misma forma, dado que un bi-retículo consta de dos órdenes, es posible establecer otro operador unario, digamos -, sobre el orden restante $\leq_c$ en *B*, es decir, si $x \leq_c y$ entonces $-y \leq_c -x$. Además, se exige que $\sim\sim x=x$ y también que $--x=x$.

Se fijará ahora un bi-retículo distributivo y completo que permita ilustrar la discusión presentada. Considérese a *Nine* como el bi-retículo de anotaciones a utilizar en ***s.r.a.***. El diagrama de Hasse para *Nine* se presenta en la figura 2:

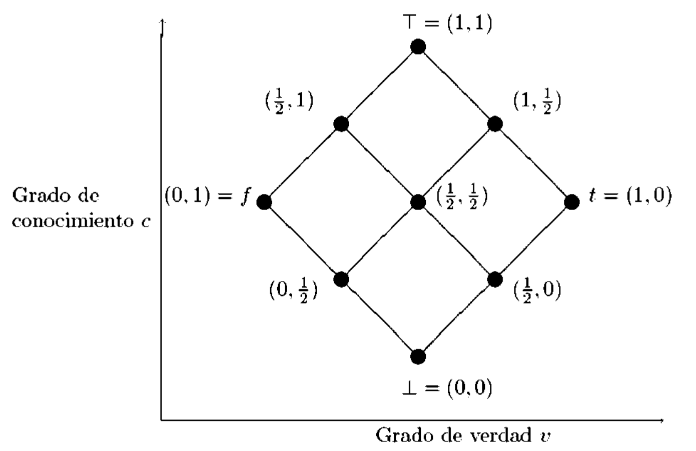

Figura 2. Diagrama de Hasse para el bi-retículo *Nine*

3 Si $a \leq b$ entonces

El bi-retículo *Nine* está formado por el conjunto $\{0, \frac{1}{2}, 1\} \times \{0, 1\}$ junto con los órdenes definidos por:

$$\langle \mu, \rho \rangle \leq_v \langle \beta, \gamma \rangle \text{ sii } \beta \leq \mu \text{ y } \rho \leq \gamma \qquad (7)$$

$$\langle \mu, \rho \rangle \leq_c \langle \beta, \gamma \rangle \text{ sii } \mu \leq \beta \text{ y } \rho \leq \gamma$$

Además, los operadores de involución y conflación en *Nine* se definen respectivamente por

$$\sim \langle \mu, \rho \rangle = \langle \rho, \mu \rangle \qquad (8)$$

$$\text{y } -\langle \mu, \rho \rangle \qquad (9)$$

construídos a partir de la figura 3

De esta forma una proposición *P*, es más verdadera que una proposición *Q* si y solo si la evidencia negativa para *P* es menor que la evidencia negativa para *Q* y la evidencia positiva para *P* es más grande que la evidencia positiva para *Q*. Análogamente, *P* tiene más información que *Q* si la evidencia tanto a favor como en contra de *P* es más grande que la correspondiente evidencia para *Q*.

Las operaciones algebráicas sobre *Nine*, están definidas por :

$$\langle \mu, \rho \rangle \wedge \langle \beta, \gamma \rangle = \langle \mu \bullet \beta, \rho + \gamma \rangle \qquad (10)$$

$$\langle \mu, \rho \rangle \vee \langle \beta, \gamma \rangle = \langle \mu + \beta, \rho \bullet \gamma \rangle \qquad (11)$$

$$\langle \mu, \rho \rangle \otimes \langle \beta, \gamma \rangle = \langle \mu \bullet \beta, \rho \bullet \gamma \rangle \qquad (12)$$

$$\langle \mu, \rho \rangle \oplus \langle \beta, \gamma \rangle = \langle \mu + \beta, \rho + \gamma \rangle \qquad (13)$$

donde $+$ y $\cdot$ se refieren al sup, e inf entre cada par de elementos. Con este, se hace explícita también la presencia de dos estructuras algebráicas inmersas en *Nine*. Esto hace directa la justificación de la distributividad de *Nine*.

Ahora bien, es posible percibir también como la negación de una cláusula transfiere esta negación a una operación de involución o conflación sobre un bi-retículo. La demostración de este resultado puede consultarse en [1].

μ	$(0,1)$	$(\frac{1}{2},1)$	$(1,1)$	$(1,\frac{1}{2})$	$(1,0)$	$(\frac{1}{2},\frac{1}{2})$	$(\frac{1}{2},0)$	$(0,0)$	$(0,\frac{1}{2})$
$-\mu$	$(0,1)$	$(0,\frac{1}{2})$	$(0,0)$	$(\frac{1}{2},0)$	$(1,0)$	$(\frac{1}{2},\frac{1}{2})$	$(1,\frac{1}{2})$	$(1,1)$	$(\frac{1}{2},1)$

Figura 3 Operadores de involución y conflación.

Lema 5.1: Sea *I* una interpretación. Entonces I% $\exists\neg A_\lambda$ si y solo si I% $\exists A_{\sim\lambda}$ donde $\exists$ se refiere a la clausura existencial de las variables libres sobre *A*. Así que dado un átomo anotado A_λ se tiene que

$$\neg A_\lambda \equiv A_{\sim\lambda} \qquad (14)$$

De esta forma, queda claro que dado un conjunto de cláusulas anotadas, la operación de negación puede definirse sobre un bi-retículo distributivo y completo. Con esto en mente, puede introducirse el operador de hiper-resolución que ser constituirá en la herramienta deductiva del ***s.r.a***.

Sin embargo, debe tenerse presente que es el anotador quien decide como se interpreta la información presente en el bi-retículo, es decir, en vez de considerar el orden de veracidad y el orden de conocimiento, el anotador podría tomar estos órdenes como órdenes de espacio y tiempo sobre *Nine*.. Para ello, dado un par de elementos en *Nine* digamos $\langle e_1, e_2 \rangle, \langle t_1, t_2 \rangle$, en donde $\langle e_1, e_2 \rangle$ representará un desplazamiento del punto e_1 al punto e_2 en el orden $\leq_e$ de espacio, mientras que la anotación de la forma $\langle t_1, t_2 \rangle$ sobre el orden $\leq_t$ indicará una variación positiva de tiempo desde t_1 a t_2 si $t_1 \leq_e t_2$, mientras que si $t_2 \leq_e t_1$ entonces se registra una variación negativa de tiempo, es decir , si un objeto se desplaza de un punto *A* a un punto *B* en un tiempo que va de t_2 a t_1 es porque la ruta recorrida por el objeto es más larga que la tomó para llegar en un tiempo de t_1 a t_2. Aunque se ha modificado la forma como interpretamos la información provista por las anotaciones, la estructura interna de las operaciones no ha cambiado. Así $\vee$, será tomado como el supremo sobre el orden de espacio $\leq_e$, mientras que $\oplus$ se tomará como supremo en el orden de tiempo $\leq_t$.

De la misma forma entonces, una anotación de espacio sobre una fórmula *P* es más informativa que una fórmula *Q* si la información contenida en *P* es más abundante que aquella contenida en *Q*. Una interpretación semejante se tiene en el caso de anotaciones de tiempo. Obviamente, las operaciones de involución y de conflación operan de la misma manera.

Observe también el sentido que se le da a la operación negación sobre el espacio y tiempo. Por la forma en cómo estas se definen, tienen una proximidad afín a la negación clásica, esto es, en un sentido epistemológico, indicando con cada operación, ya sea esta de involución o de conflación, que se tiene información negativa respecto del objeto que se trate, ya sea que se trate como en el caso propuesto, de tiempo o espacio

Definición 5.3: Sea

$$M(\mu,\rho) = \{\gamma \in B / \sup\{\gamma,\rho\} \geq \mu\}. \quad (15)$$

El $\searrow$-operador está definido por

$$\searrow(\mu,\rho) = \inf M(\mu,\rho) \quad (16)$$

Es necesario aclarar en este punto el porqué de estas dos definiciones. En primer lugar, para definir la resolvente entre las cláusulas $\sim \mathrm{A}_\mu \vee E_1$ y $\mathrm{A}_\rho \vee E_2$ en el caso en que $\mu \leq \rho$ la resolvente está dada por $E_1 \vee E_2$. Así, dos cláslulas 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 tengan dos 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 *regla de la deducción.* Sin embargo, sería deseable contar con una regla para anotaciones regulares que no depende de la reducción y en la que se pueda aplicar de alguna forma la regla de resolución anotada. Tal intento presenta dificultades cuando quiere aplicarse a anotaciones regulares. Por ejemplo, sean $\sim \mathrm{A}_\mu$ y A_ρ literales anotados y suponga que Dado que no son comparables, no es posible aplicar la regla de resolución. Por ello, para mantener la reducción regular es necesario encontrar una especie de anotación minimal S_0 que contenga a $(\uparrow\mu)^c \cap \uparrow\rho$ y asi el resolvente de las dos cláusulas tenga a S_0 como su anotación. En [5] se demuestra que tal conjunto minimal sobre las anotaciones esta dado por

$(\uparrow\searrow(\mu,\rho))^c$ por lo que es indispensable que el retículo sea ordinario. Por otra parte, tal definición elimina la necesidad de hacer reducción sobre las cláusulas anotadas cuando μ y ρ son incomparables ya que por la minimalidad de $(\uparrow\searrow(\mu,\rho))^c$ la regularidad sobre las anotaciones está garantizada.

En [5] Lu demuestra la equivalencia entre retículos ordinarios y distributivos. Si el retículo es completo y finito, la implementación computacional es eficiente. Por ello la exigencia en este trabajo de retículos (y por ende bi-retículos con estas características). Con este esquema se logra extender la *CT* presentada por Fruwirht en [3] pues ahora toda cláusula (incluyendo las de Horn) puede estar anotada con valores en un bi-retículo distributivo y completo con un procedimiento de inferencia válido y completo.

De hecho, este esquema también permite involucrar en la sintaxis presentada, tantas anotaciones sobre un cláusula como se requieran. Claro está, es el anotador quien interpreta estas anotaciones dependiendo del contexto en que se encuentre. Lo importante es que anotando sobre bí-retículos siempre se tiene una noción de grado de creencia sobre la información presentada en las anotaciones. De esta forma se generaliza el esquema de programación lógica con restricciones espacio-temporales presentada en [8] tal y como se podrá ver en un ejemplo posterior.

Así, en el sistema de restricciones introducido, un literal anotado será de la forma $A: \lambda_1, \lambda_2$ donde λ_1 es información sobre el espacio y λ_2 es información sobre tiempo.

De esta forma introducimos pues una negación explícita - $\neg$, la cual se determina en términos de una involución y una conflación sobre la anotación del espacio y tiempo respectivamente, es decir

$$\neg A: \lambda_1, \lambda_2 \equiv A: \sim \lambda_1, -\lambda_2 \quad (17)$$

VI. EJEMPLO (RESTRICCIONES SOBRE)

El objetivo es indicar la forma como pueden introducirse restricciones sobre ***s.r.a*** de tal forma que por vía hiper-î-resolución puedan encontrarse los valores que satisfacen las restricciones impuestas, o en su defecto la imposibilidad de cumplirlas. Tales restricciones se imponen se imponen sobre las anotaciones. Estas anotaciones hacen referencia a información de tipo espacial y temporal.

Imagine que desea representar una colección de salones de clase los cuales están identificados con un número y en los cuales se dictan clases en diferentes horarios. Suponga que se está interesado en encontrar una tabla de horarios en la cual los salones no se crucen uno al otro con algún curso y que además respeten los horarios disponibles de los profesores. Problemas de este tipo pueden modelarse de una manera convencional usando un paradigma de programación concurrente (es decir, ejecución múltiple de procesos relacionados entre sí) con restricciones y de hecho ha sido un problema de investigación exhaustiva desde la década de los 70. Una propuesta de solución ya implementada se conoce como *PATHOS* [7].

Se desea hacer un modelo basado en lógicas anotadas que permita indicar la clase que se dicta, el profesor encargado del curso y el horario (días de la semana y hora en que se dicta) del cuso a asignar. Se debe incluir también una cláusula que registre los horarios no disponibles del profesor, es decir, días y horas en los cuales los profesores no están disponibles para dictar un curso. Se agrega una cláusula que indique los cursos que están por asignar junto con su horario propuesto. Vía $\searrow$-resolución deseamos obtener una asignación de cursos donde se satisfagan las restricciones previas planteadas por los profesores. Por supuesto, este es un ejemplo simplificado en extremo pero que permite hacer un uso simple del poder expresivo del sistema de restricciones construido.

Suponga que Dictaclase(x,y,z): $\langle s, e\rangle, \langle t_1, t_2\rangle$ indica que el profesor x dicta la materia y , grupo z en el salón s del edificio e en el horario de t1 a t2[4]. Debe destacarse la flexibilidad interpretativa de la lógica anotada, pues el anotador tiene la libertad de ajustar las anotaciones como sean más convenientes. En este caso se ha pasado de valores de creencia e información, a valores de espacio y tiempo, los cuales a su vez pueden adecuarse a salones y franjas horarias. Incluso, se podría incluir un tercer par anotador que indique los días en los cuales se dicta el curso. Tal par anotador puede ser un elemento de *Nine* renombrado con los días de la semana y cuya negación funciona como la involución en *Nine*. Por ejemplo:

Dictaclase(Diego,Geometría,A): $\langle 1,0\rangle, \left\langle 1, \frac{1}{2}\right\rangle, \langle \textit{Martes, Jueves}\rangle$

significa que el profesor Diego dicta el curso de geometría grupo A en el salón 1 del edificio 0 entre la franja de 1 a $\frac{1}{2}$ día los días martes y jueves. De esta forma, expresar que dos sesiones de un mismo grupo se deben ubicar en días diferentes puede indicarse así:

Si

$$\mathit{Dictaclase}(x,y,z) : \langle s,e\rangle, \langle t_1,t_2\rangle, \langle d_1,d_2\rangle \doteq$$

$$\mathit{Dictaclase}(x,y,z) : \langle s,e\rangle, \langle t_1,t_2\rangle, \langle o_1,o_2\rangle$$

entonces $\langle o_1, o_2\rangle = \langle \widehat{d_1}, \widehat{d_2}\rangle$ con $\langle \widehat{d_1}, \widehat{d_2}\rangle \neq \langle d_1, d_2\rangle$ y $\langle \widehat{d_1}, \widehat{d_2}\rangle \in \mathit{Nine}$

De la misma manera pueden imponerse restricciones sobre un profesor que no puede asumir dos cursos al mismo tiempo, es decir, si

$$\mathit{Dictaclase}(x,y_1,z) : \langle s,e\rangle, \langle t_a,t_b\rangle, \langle d_a,d_b\rangle$$

hace parte del almacén de restricciones, entonces

$$\mathit{Dictaclase}(x,y_2,z) : \langle s,e\rangle, \langle t_1,t_2\rangle, \langle d_1,d_2\rangle$$

Debe implicar que

$$y_1 = y_2 \mid \langle t_a,t_b\rangle \neq \langle t_1,t_2\rangle \mid \langle d_a,d_b\rangle \neq \langle d_1,d_2\rangle$$

En caso de que se presente algún tipo de inconsistencia, al estar ubicado en una lógica paraconsistente tal inconsistencia no implica la trivialización de la información presente en el almacén de restricciones, lo que preserva la estabilidad del sistema, pues es simplemente una indicación de información insuficiente o incorrecta, que puede ser corregida por el anotador, o brindar una salida optativa conveniente.

Sea *limitaciones(x)*: $\langle d_1, d_2\rangle, \langle t_1, t_2\rangle$ la restricción que indica los días y la franja horaria en que el profesor x no puede recibir carga horaria (es un poco exagerado que cada profesor tenga la misma franja restringida para días distintos pero esto es por mantener la sencillez de la presentación).

Tomemos un caso de asignación de salones sujeto solamente a las tres restricciones anteriores para evitar complejidades innecesarias.

En este caso tenemos un conjunto de variables $x_1, x_2, x_3, \ldots, x_n$ cuyo dominio es el universo de Herbrand anotado junto con tres restricciones, esto es:

- $\mathit{Dictaclase}(x,y,z) : \langle s,e\rangle, \langle t_1,t_2\rangle, \langle d_1,d_2\rangle \doteq$

  $\mathit{Dictaclase}(x,y,z) : \langle s,e\rangle, \langle t_1,t_2\rangle, \langle o_1,o_2\rangle$ entonces $\langle o_1,o_2\rangle = \langle \widehat{d_1}, \widehat{d_2}\rangle$ con $\langle \widehat{d_1}, \widehat{d_2}\rangle \neq \langle d_1,d_2\rangle$ y $\langle \widehat{d_1}, \widehat{d_2}\rangle \in \mathit{Nine}$
- $y_1 = y_2 \mid \langle t_a,t_b\rangle \neq \langle t_1,t_2\rangle \mid \langle d_a,d_b\rangle \neq \langle d_1,d_2\rangle$.
- $\mathit{limitaciones}(x) : \langle d_1,d_2\rangle, \langle t_1,t_2\rangle$

`Dictaclase(Diego,Geometría,A)`: $\langle 1,0\rangle, \langle 1,\frac{1}{2}\rangle, \langle \mathit{Martes}, \mathit{Jueves}\rangle$
`Dictaclase(Guillermo,Lógica,C)`: $\langle \frac{1}{2},1\rangle, \langle 0,1\rangle, \langle \mathit{Miercoles}, \mathit{Viernes}\rangle$
`Dictaclase(Doris,Cálculo,B)`: $\langle 1,0\rangle, \langle \frac{1}{2},1\rangle, \langle \mathit{Martes}, \mathit{Jueves}\rangle$
`limitaciones(Guillermo)`: $\langle \mathit{Martes}, \mathit{Jueves}\rangle, \langle 0,1\rangle$
`limitaciones(Diego)`: $\langle \mathit{Lunes}, \mathit{Viernes}\rangle, \langle 0,1\rangle$
`limitaciones(Doris)`: $\langle \mathit{Lunes}, \mathit{Miercoles}\rangle, \langle 0,\frac{1}{2}\rangle$
`cursopendiente(Algebra,B)`: $\langle 0,1\rangle, \langle 0,1\rangle, \langle \mathit{Lunes}, \mathit{Viernes}\rangle$
`cursopendiente(Ecuaciones,A)`: $\langle \frac{1}{2},\frac{1}{2}\rangle, \langle 1,\frac{1}{2}\rangle, \langle \mathit{Miercoles}, \mathit{Viernes}\rangle$
`cursopendiente(Análisis,A)`: $\langle \frac{1}{2},1\rangle, \langle 0,1\rangle, \langle \mathit{Martes}, \mathit{Jueves}\rangle$

Usando ahora como mecanismo de inferencia î-resolución obtenernos una respuesta plausible para los cursos pendientes por asignar, es decir:

4 Es indiferente si $t_2 \geq t_1$, el punto es que esa franja horaria es ocupada por el profesor x con la materia y.

$$\texttt{Dictaclase(Guillermo,Algebra,B)}:\langle 0,1\rangle,\langle 0,1\rangle,\langle Lunes, Viernes\rangle$$
$$\texttt{Dictaclase(Diego,Ecuaciones,A)}:\langle \tfrac{1}{2},\tfrac{1}{2}\rangle,\langle 1,\tfrac{1}{2}\rangle,\langle Miercoles, Viernes\rangle$$
$$\texttt{Dictaclase(Doris,Análisis,A)}:\langle \tfrac{1}{2},1\rangle,\langle 0,1\rangle,\langle Martes, Jueves\rangle$$

En caso de presentarse inconsistencias, como cruces de horarios u horas no disponibles para un profesor, al ser un sistema paraconsistente, la estabilidad no se ve afectada. Es cuestión de brindar más información al sistema o que el anotador decida la prioridad dependiendo de las circunstancias establecidas alrededor del sistema.

REFERENCIAS

[1]Carbogim Daniela Vasconcelos.1996. Programacao em logica anotada.Teoría y aplicaciones. Tesis de maestría en matemática aplicada,Sau Paulo

[2]Fruwirth T.2003.Essentiasl of constraint programming. Springer-Verlag. Alemania.

[3]Fruwirth T.1996. Temporal annotated constraint logic programming. Journal of Symbolic Computation 11,1-000

[4]Kifer Michael.1992.Theory of generalized annotated logic programming and its applications. Journal of logic programming. 12:335-367

[5]Lu j,Murray N,and Rosental E.2005. Deduction and search strategies for regular multiple valued logics. Journal of Mult-Valued Logic and Soft Computing. Volume 11,pp 375-406/Old City Publishing.

[6]Ortíz G.2002. The annotated logic OPBL, Proceedings of WCP2000,Walter Carnielli, Marcelo Coniglio and Itala D'Ottaviano.

[7]Quesada Luis Omar, Rueda Camilo.2002. Planeamiento Horario Universitario Orientado-Objetos en programación concurrente por restricciones. Epiciclos. Cali-Colombia 1(1):103 116

[8]Raffaeta A.2000. Spatio temporal annotated constraint logic programming, journal of symbolic computation.

[9]Ramírez Ovalle Carlos Ernesto. 2008. Semánticas de punto fijo para programas anotados. Una realización por Mho-Resolución.Matemáticas Enseñanza universitaria. Vol XVI Num 2-2008.

[10]Saraswat V. Models for concurrent constraint programming.XEROX . Palo-Alto. USA