Programas Lógicos Disyuntivos y la Demostrabilidad de Átomos en C_ω

Mauricio Osorio[1], José R. Arrazola[2], José L. Carballido[2], and Oscar Estrada[2]

[1] *Universidad de las Américas - Puebla,* osoriomauri@gmail.com
[2] *Benemérita Universidad Autónoma de Puebla,* {arrazola, carballido}@fcfm.buap.mx, oestrada2005@gmail.com

Resumen Tomando como base las ideas presentadas en [1], generalizamos algunos de sus resultados a programas disyuntivos; dichas generalizaciones se obtendrán extendiendo el siguiente resultado presentado en [1]: Dado un programa normal P y a un átomo, se tiene que P $\vdash_C a$ si, y sólo si, P $\vdash_{C_\omega} a$.

1. Sintaxis y Notaciones

Estamos considerando un lenguaje formal (proposicional) conformado por: un conjunto numerable de elementos llamados átomos $\mathcal{L}$, los conectivos binarios estándar $\wedge$, $\vee$, $\rightarrow$, y el conectivo unario $\neg$. Las fórmulas y las teorías son construidas como es usual en lógica clásica.

2. Axiomatizaciones

En este caso partiremos de una lógica básica, la lógica positiva, a partir de la cual al agregar diversos axiomas podremos obtener axiomatizaciones para cada una de las lógicas que incluye nuestro estudio.

La lógica positiva se define como el conjunto de axiomas:

- **Pos1** $\varphi \rightarrow (\psi \rightarrow \varphi)$
- **Pos2** $(\varphi \rightarrow (\psi \rightarrow \sigma)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \sigma))$
- **Pos3** $\varphi \wedge \psi \rightarrow \varphi$
- **Pos4** $\varphi \wedge \psi \rightarrow \psi$
- **Pos5** $\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))$
- **Pos6** $\varphi \rightarrow (\varphi \vee \psi)$
- **Pos7** $\psi \rightarrow (\varphi \vee \psi)$
- **Pos8** $(\varphi \rightarrow \sigma) \rightarrow ((\psi \rightarrow \sigma) \rightarrow (\varphi \vee \psi \rightarrow \sigma))$

Junto con la regla de inferencia Modus Ponens (MP) que nos permite obtener ψ de φ y $\varphi \rightarrow \psi$. Esta regla de inferencia estará presente en todos los sistemas lógicos que se definen a continuación así como en todos aquellos que sean tratados en este trabajo.

El sistema C_ω creado por da Costa, se define como el conjunto de axiomas de la lógica positiva, agregando los siguientes axiomas:

- **Cw1** $\varphi \vee \neg\varphi$
- **Cw2** $\neg\neg\varphi \to \varphi$.

C_ω es la lógica paraconsistente más pequeña, es preciso hacer notar que $\varphi \vee \neg\varphi$ es teorema de C_ω mientras que $(\varphi \wedge \neg\varphi) \to \psi$ no lo es, esta última afirmación es una de las motivaciones en la creación de lógicas paraconsistentes.

La lógica paraconsistente de tres valores Pac se obtiene agregando a la lógica C_ω los siguientes axiomas:

- **Pac1** $((\varphi \to \psi) \to \varphi) \to \varphi$
- **Pac2** $\varphi \to \neg\neg\varphi$.
- **Pac3** $(\neg(\varphi \vee \psi) \to (\neg\varphi \wedge \neg\psi)) \wedge ((\neg\varphi \wedge \neg\psi) \to \neg(\varphi \vee \psi))$
- **Pac4** $(\neg(\varphi \wedge \psi) \to (\neg\varphi \vee \neg\psi)) \wedge ((\neg\varphi \vee \neg\psi) \to \neg(\varphi \wedge \psi))$
- **Pac5** $(\neg(\varphi \to \psi) \to (\varphi \wedge \neg\psi)) \wedge ((\varphi \wedge \neg\psi) \to \neg(\varphi \to \psi))$

En otra dirección, la lógica intuicionista se define como la lógica positiva junto con los siguientes axiomas:

- **Int1** $(\varphi \to \psi) \to ((\varphi \to \neg\psi) \to \neg\varphi)$
- **Int2** $\neg\varphi \to (\varphi \to \psi)$.

Una de las lógicas intermedias más conocidas, la lógica HT (Here and there), se obtiene a partir de intuicionismo agregando el axioma:

- **G3** $(\neg\psi \to \varphi) \to (((\varphi \to \psi) \to \varphi) \to \varphi)$.

Esta lógica es equivalente a la lógica trivaluada de Gödel G_3.

Podemos obtener la Lógica Clásica de alguna de las dos siguientes formas:

- La lógica intuicionista agregando el axioma:
  - **CL1** $(\neg\varphi \to \varphi) \to \varphi$.
- Un conjunto axiomático más pequeño, que consta de los axiomas **Pos1**, **Pos2** y un tercer axioma:
  - **CL2** $(\neg\psi \to \neg\varphi) \to ((\neg\psi \to \varphi) \to \psi)$

Una vez que hemos presentado estas lógicas es importante encontrar las relaciones existentes entre ellas. Gracias a la axiomatización tipo Hilbert podemos comparar las lógicas, sin embargo más adelante se hará una comparación más detallada.

Notación: Emplearemos la notación $\vdash_{\mathrm{X}} F$ para denotar que la fómula F es demostrable ya sea como tautología o como teorema en una lógica dada X. La notación estándar para hablar de demostrabilidad en teoría de modelos es $\models$ mientras que para el caso de teoría de prueba se emplea $\vdash$. Abusando de la notación en este trabajo emplearemos $\vdash$ para ambos casos.

Una *teoría* T es simplemente un conjunto de fórmulas, su *lenguaje*, denotado como $\mathcal{L}_T$, es el conjunto de átomos que aparecen en la teoría. Dada una teoría

T definimos la negación de la teoría como $\neg T = \{\neg A \mid A \in T\}$. Una *literal* es un átomo a (una literal positiva) o la negación de un átomo $\neg a$ (una literal negativa). El complemento de un conjunto de átomos lo denotamos por $\widetilde{M} = \mathcal{L}_T \setminus M$.

Si T es una teoría empleamos el símbolo $T \vdash_{\mathrm{X}} F$ para denotar que $\vdash_{\mathrm{X}}$ $(F_1 \wedge \cdots \wedge F_n) \to F$ para algunas fórmulas $F_i \in T$. Si T es una teoría y M un conjunto de átomos escribiremos $T \Vdash_{\mathrm{X}} M$ cuando: $T \vdash_{\mathrm{X}} a$ para todo $a \in M$ y M es un modelo estándar bivaluado de T (como en lógica clásica).
Diremos que la lógica X es *más fuerte* que la lógica Y si $Th(Y) \subset Th(X)$, donde Th denota la cerradura deductiva de la teoría, análogamente X es *más débil* que la lógica Y si $Th(X) \subset Th(Y)$. De acuerdo a estas definiciones una lógica Z estará *entre* las lógicas X y Y si es más fuerte que la primera y a la vez más débil que la segunda.

3. Trivaluaciones

3.1. Lógica Paraconsistente de Tres Valores Pac

Posiblemente la forma más fácil de generar una lógica paraconsistente es usando una lógica multivaluada, es decir, una lógica con más de dos valores de verdad. Las fórmulas que son válidas en una interpretación multivaluada son aquellas que tienen valor designado, donde el valor o los valores de verdad designado(s) son simplemente algunos valores de verdad que se fijaron previamente.

Una lógica multivaluada será paraconsistente si permite para alguna fórmula que tanto ella como su negación tomen el valor designado. Esta es la forma en que podemos construir la lógica paraconsistente de tres valores a la cual llamaremos Pac.

La lógica Pac tiene una estructura básica formada por tres valores de verdad V, F y $\perp$; los dos primeros corresponden a los valores de verdad clásicos, "verdadero" y "falso", mientras que el tercero representa que el valor de verdad es tanto verdadero como falso. Nosotros identificaremos los valores de verdad V, $\perp$ y F con los valores 2, 1 y 0 respectivamente. De este conjunto de valores se toman a dos de ellos como designados, 2 y 1. Esta lógica consta de un lenguaje formal (proposicional) que contiene un conjunto numerable de fórmulas atómicas, $\mathcal{L}$, los conectivos binarios $\vee_{\mathrm{Pac}}$, $\wedge_{\mathrm{Pac}}$ y $\to_{\mathrm{Pac}}$ y el conectivo unario $\neg_{\mathrm{Pac}}$. La valuación de los conectivos $\neg_{\mathrm{Pac}}$ y $\to_{\mathrm{Pac}}$ se encuentran en el Cuadro 1, y definimos $a \vee_{\mathrm{Pac}} b = max\{a, b\}$ y $a \wedge_{\mathrm{Pac}} b = min\{a, b\}$.

Cuadro1. Valuación de los conectivos $\neg_{\mathrm{Pac}}$ y $\to_{\mathrm{Pac}}$.

a	$\neg_{\mathrm{Pac}} a$
0	2
1	1
2	0

$\to_{\mathrm{Pac}}$	0	1	2
0	2	2	2
1	0	1	2
2	0	1	2

Esta lógica fue investigada y axiomatizada en [2], donde se muestra que es la lógica paraconsistente máxima en la que las contradicciones no implican cualquier cosa.

Como consecuencia de lo visto en la sección anterior tenemos que C_ω y Pac poseen el Teorema de la Deducción.

3.2. Definición de $\mathbf{G_3}$ y $\mathbf{G'_3}$

Construiremos dos lógicas trivaluadas definiendo primero sus conectivos básicos, negación e implicación, consideraremos la valuación de los conectivos $\neg_{G_3}$, $\neg_{G'_3}$ y $\rightarrow_{G_3}$.

En el Cuadro 2 el lector puede observar la correspondencia entre los conectivos y las valuaciones para $\neg_{G_3}$, $\neg_{G'_3}$ y $\rightarrow_{G_3}$.

Cuadro2. Valuación de los conectivos $\neg_{G_3}$, $\neg_{G'_3}$ y $\rightarrow_{G_3}$.

a	$\neg_{G_3} a$	$\neg_{G'_3} a$
0	2	2
1	0	2
2	0	0

$\rightarrow_{G_3}$	0	1	2
0	2	2	2
1	0	2	2
2	0	1	2

Se tienen dos negaciones distintas, tomando cada una de ellas construimos una lógica diferente: G_3 con los conectivos $\neg_{G_3}$ y $\rightarrow_{G_3}$ y la lógica G'_3 con los conectivos $\neg_{G'_3}$ y $\rightarrow_{G_3}$. La disyunción y conjunción para G_3 y G'_3 son $a \vee b = max\{a, b\}$ y $a \wedge b = min\{a, b\}$.

En Carnielli et al. [3] se presenta la lógica G'_3 únicamente para probar que $\alpha \vee (\alpha \rightarrow \beta)$ no es un teorema de C_ω. Como consecuencia de esto se tiene el siguiente teorema.

Teorema 1 G'_3 *es robusta con respecto a* C_ω. *Es decir cada teorema en* C_ω *es una tautología en* G'_3 .

Si deseamos encontrar la relación que existe entre G'_3 y Pac tenemos que G'_3 no es comparable con Pac dado que $\vdash_{G'_3} \neg a \wedge \neg\neg a \rightarrow b$ mientras que $\nvdash_{\text{Pac}} \neg a \wedge \neg\neg a \rightarrow b$, por otra parte $\vdash_{\text{Pac}} a \rightarrow \neg\neg a$ pero $\nvdash_{G'_3} a \rightarrow \neg\neg a$, donde los conectivos están parametrizados respecto a la lógica indicada. En la Figura 1 podemos encontrar un esquema que nos permite comparar algunas de las lógicas que hemos presentado.

Un resultado simple pero interesante es que en G'_3 es posible expresar G_3, dado que $\neg_{G_3} a = a \rightarrow_{G_3} (\neg_{G'_3} a \wedge \neg_{G'_3} \neg_{G'_3} a)$. Además de que la definición de $\neg_{G'_3}$ está basada en la traducción propuesta por Gelfond[4].

Figura1. Comparación de algunas lógicas

$$\begin{tikzcd}
 & C_\omega \arrow[r] \arrow[dr] & \mathrm{Pac} & \\
\mathrm{Pos} \arrow[ur] \arrow[dr] & & \mathrm{G'}_3 & \\
 & \mathrm{I} \arrow[r] & \mathrm{G}_3 \arrow[r] & \mathrm{C}
\end{tikzcd}$$

4. Programas Lógicos

Definición 1 *En nuestro contexto un* programa lógico *es únicamente una teoría y una* clase de programas lógicos *es un conjunto de programas lógicos.*

De hecho podemos pensar que las palabras teoría y programa son sinónimos, usualmente empleamos la primera cuando estamos en el contexto de lógica y el segundo cuando se trata de programación.

La sintaxis de las fórmulas usualmente se define en términos de algunas fórmulas especiales conocidas como *cláusulas*.

Definición 2 *Una* cláusula *es una fórmula de la forma* $\mathcal{H} \leftarrow \mathcal{B}$ *donde la implicación es el conectivo principal y* $\mathcal{H}, \mathcal{B}$ *representan fórmulas.*

- *Las fórmulas* $\mathcal{H}$ *y* $\mathcal{B}$ *se conocen como* cabeza *y* cuerpo *de la cláusula respectivamente.*
- *La cláusula* $\bot \leftarrow \mathcal{B}$ *se le llama* constraint *y se dice que la cabeza está vacía.*
- *Una fórmula* $\mathcal{H}$*, cuyo conectivo principal no sea una implicación se llamará un* hecho*.*

En la literatura referente a la programación lógica podemos encontrar diferentes clases de programas.

Definición 3 *Una* cláusula disyuntiva *es una cláusula cuya cabeza es una disyunción de átomos mientras que el cuerpo es una conjunción de literales (átomos o átomos negados). Un* programa disyuntivo *es un conjunto de cláusulas disyuntivas.*

5. Demostrabilidad de Átomos en C_ω

Definición 4 *Sea* P *un programa disyuntivo, definimos:*
$\mathbf{To\_Pos}(\mathrm{P}) := \{a_1 \vee a_2 \vee \cdots \vee a_n \vee c_1 \vee c_2 \vee \cdots \vee c_m \longleftarrow b_1 \wedge b_2 \wedge \cdots \wedge b_k$
$\mid a_1 \vee a_2 \vee \cdots \vee a_n \longleftarrow b_1 \wedge b_2 \wedge \cdots \wedge b_k \wedge \neg c_1 \wedge \neg c_2 \wedge \cdots \wedge \neg c_m \in \mathrm{P}\}.$

Lema 2 *Sea* P *un programa disyuntivo y* a *un átomo, entonces:*
$\mathrm{P} \vdash_C \mathrm{a}$ *si, y sólo si,* $\mathbf{To\_Pos}(\mathrm{P}) \vdash_C \mathrm{a}$.

Demostración. Inmediato de la definición de **To_Pos**(P) y los teoremas de C.

Lema 3 (De Jongh and Hendriks [5]) *Sea* $L = \{A \leftarrow B : A, B \in [\wedge, \vee, \bot, \top]\}$ *y sean* T *una teoría contenida en* L *y* F *una fórmula en* L *entonces* $\mathrm{T} \vdash_C \mathrm{F}$ *si, y sólo si,* $\mathrm{T} \vdash_I \mathrm{F}$.

Lema 4 (Carnielli and Marcos [3]) *Sea* L *el lenguaje formado por las fórmulas positivas y sean* T *una teoría contenida en* L *y* F *una fórmula en* L *entonces* $\mathrm{T} \vdash_I \mathrm{F}$ *implica que* $\mathrm{T} \vdash_{C_\omega} \mathrm{F}$.

Observe que, si $A, C \vdash_{C_\omega}$ D y B,$\neg C \vdash_{C_\omega}, D$ entonces $A, B \vdash_{C_\omega} D$, como consecuencia del Teorema de la Deducción y de que **Pos8** pertenece a C_ω.

Lema 5 *Si* $A_1, A_2, ..., A_n, C \vdash_{C_\omega} D$ *y* $B_1, B_2, ..., B_m, \neg C \vdash_{C_\omega} D$*, entonces* $A_1, A_2, ..., A_n, B_1, B_2, ..., B_m \vdash_{C_\omega} D$.

Demostración. Inmediato de la observación anterior.

Lema 6 *Sea* P *un programa disyuntivo, entonces* $\mathrm{P} \vdash_{C_\omega}$ **To_Pos***(*P*)*

Demostración. Observemos que al aplicar el Axioma **Pos7** y Modus Ponens podemos probar que $\mathrm{C} \vdash_{C_\omega} \mathrm{A} \vee \mathrm{C}$ $(*)$, y además, podemos mostrar que: $(B \wedge \neg C) \rightarrow A, B, \neg C \vdash_{C_\omega} A \vee C$ (**). De (*) y (**) se sigue por el Lema 5 que B, $(\mathrm{B} \wedge \neg \mathrm{C}) \rightarrow \mathrm{A} \vdash_{C_\omega} \mathrm{A} \vee \mathrm{C}$ y por Teorema de la Deducción tenemos que $(\mathrm{B} \wedge \neg \mathrm{C}) \rightarrow \mathrm{A} \vdash_{C_\omega} \mathrm{B} \rightarrow (\mathrm{A} \vee \mathrm{C})$.

Haciendo lo mismo para cada clasula de P tenemos que $P \vdash_{C_\omega}$ **To_Pos**(P)

Proposición 7 *Sea* P *un programa disyuntivo y sea* a *un átomo, entonces* $\mathrm{P} \vdash_C \mathrm{a}$ *si, y sólo si,* $\mathrm{P} \vdash_{C_\omega} \mathrm{a}$

Demostración. Supongamos que $P \vdash_{C_\omega} a$, como C_ω es más débil que C tenemos que $P \vdash_C a$.

Supongamos que $P \vdash_C a$, por el Lema 2 sabemos que **To_Pos**$(P) \vdash_C a$, por el Lema 3 tenemos que **To_Pos**$(P) \vdash_I a$, por el Lema 4 **To_Pos**$(P) \vdash_{C_\omega} a$ (*).

Además por el Lema 6 $P \vdash_{C_\omega}$ **To_Pos**(P) (**) así que, por transitividad de (*) y (**) tenemos que $P \vdash_{C_\omega} a$.

Corolario 8 *Sea* P *un programa disyuntivo y* a *un átomo. Si* $X \in \{\mathrm{Pac}, \mathrm{G'}_3\}$ *entonces* $P \vdash_{\mathrm{C}} a$ *si, y sólo si,* $P \vdash_X a$.

Demostración. Inmediato del Teorema 1 y la Proposición 7.

6. Conclusiones

En este artículo fueron abordados diversos sistemas lógicos, algunos de los cuales son nodos inicial o final de lattices de sistemas lógicos paraconsistentes. Además, de este estudio se concluye la relación existente entre estas lattices y la lógica clásica en la deducción de átomos, teniendo como premisas programas lógicos disyuntivos, estos resultados extienden otros desarrollados para programas lógicos normales, como el ya citado en la introducción, el cual es un resultado interesante por sí mismo. Más aún, algunos resultados importantes de [1] que involucran semánticas para programas lógicos normales podrían ser extendidos a programas lógicos disyuntivos; Tal es el caso del Teorema 5.1[1], que establece la invarianza de los modelos p-estables entre las diferentes lógicas paraconsistentes, entre las que se encuentran las estudiadas aqui: $C_\omega, \mathrm{Pac}, \mathrm{G'}_3$.

Existen varias líneas para futuros trabajos, una de estas es investigar cual es la clase de programas en la que nuestros resultados siguen satisfaciéndose y lo que sucede con las semánticas correspondientes.

Referencias

1. Osorio, Arrazola, Navarro y Borja, *Logics where logical weak completions agree*, Cambridge Press University 2006. Reino Unido.
2. Arnon Avron, *Natural 3-valued logics: Characterization and proof theory.* The Journal of Symbolic Logic 56 (1): 276-294, 1991.
3. Walter A. Carnielli and João Marcos, *A Taxonomy of C-Systems*, Marcel Dekker, Inc. New York, 2002.
4. Michael Gelfond and Vladimir Lifschitz, *The Stable Model Semantics for Logic Programming*, in Robert Kowalski and Kenneth Bowen, editors, *Proceedings of the fifth International Conference on Logic Programming*, pages 1070-1080, Cambridge, Massachusetts, 1988. The MIT Press.
5. Dick de Jongh and Lex Hendrix, *Characterization of Strongly Equivalent Logic Programs in Intermediate Logics*, Theory and Practice of Logic Programming, 2002.